chapter 6

探索資料圖形

拿到一些資料，通常第一件要做的事是利用各種簡單的指標及圖形來大致了解給定資料一些可能的特性、特徵、或是規律性。這就是探索資料分析 (exploratory data analysis, EDA) 的範圍。

考慮 iris{datasets} 這個資料集，是對鳶尾花 (或蝴蝶花) 的一些量測數據。我們先將其載入：

```
> data(iris)
```

接下來看一下簡要的內容：

```
> str(iris)

'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 ...
 $ Sepal.Width : num  3.5 3 3.2 ...
 $ Petal.Length: num  1.4 1.4 1.3..
 $ Petal.Width : num  0.2 0.2 0.2..
 $ Species: Factor w/3 levels "setosa","versicolor",..:
    1 1 1...
```

這是一個資料框，共有 5 個變量及 150 筆量測數據。5 個變量分別為花萼長度、花萼寬度、花瓣長度、花瓣寬度、品種 (共有三種)。也可以看看更詳細的一些摘要：

```
> summary(iris)

Sepal.Length    Sepal.Width   ...       Species
Min.   :4.300   Min.   :2.000 ... setosa    :50
1st Qu.:5.100   1st Qu.:2.800 ... versicolor:50
Median :5.800   Median :3.000 ... virginica :50
Mean   :5.843   Mean   :3.057 ...
3rd Qu.:6.400   3rd Qu.:3.300 ...
Max.   :7.900   Max.   :4.400
```

由上可知，在此共有三個品種且各有 50 筆數據。

要看看最前面的 5 筆資料可以使用下面的指令：

```
> head(iris, n = 5)

  Sepal.Length Sepal.Width ... Species
1          5.1         3.5 ...  setosa
2          4.9         3.0 ...  setosa
3          4.7         3.2 ...  setosa
4          4.6         3.1 ...  setosa
5          5.0         3.6 ...  setosa
```

要看看最後面的 5 筆資料可以使用下面的指令：

```
> tail(iris, n = 5)

    Sepal.Length Sepal.Width ...   Species
146          6.7         3.0 ... virginica
147          6.3         2.5 ... virginica
148          6.5         3.0 ... virginica
149          6.2         3.4 ... virginica
150          5.9         3.0 ... virginica
```

若想知道品種為 "setosa" 的花萼長度數據可鍵入：

```
> iris[iris$Species == "setosa", 1]
```

或

```
> subset(iris, Species == "setosa", select = Sepal.Length)
```

若想知道品種為 "setosa" 的花萼長度與花萼寬度數據可鍵入：

```
> iris[iris$Species == "setosa", 1:2]
```

或

```
> subset(iris, Species == "setosa",
+   select = c(Sepal.Length, Sepal.Width))
```

6.1 一維資料

我們先以 iris{datasets} 這個資料框的第一行花萼長度之資料來做一維資料分析之範例：

```
> x <- iris[, 1]
```

首先來看看資料分佈之範圍：

```
> c(min(x), max(x))
[1] 4.3 7.9

> range(x)
[1] 4.3 7.9
```

需要更好的摘要可使用下列的指令：

```
> summary(x)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  4.300   5.100   5.800   5.843   6.400   7.900
```

在上面的列印中，Min. 代表最小值，1st Qu. 代表第一四分位數 (first quartile) 或是 25% 分位數 (quantile) 或百分位數 (percentile)，Median 代表中位數 (即第二四分位數或是 50% 分位數)，Mean 代表樣本平均數 (sample mean)，3rd Qu. 代表第三四分位數或是 75% 分位數，Max. 代表最大值。我們也可以使用下列的指令：

```
> fivenum(x)  # Tukey's five number summary
[1] 4.3 5.1 5.8 6.4 7.9
```

這五個數字分別為最小值、第一四分位數、第二四分位數、第三四分位數、及最大值。

要了解一群數字之中心在那兒可以計算樣本平均數或是中位數：

```
> mean(x)
[1] 5.843333

> median(x)
[1] 5.8
```

我們也可以使用四分位 (數間) 距 (interquartile range)，即第一四分位數與第三四分位數之間距，來了解一群數字之中心部份：

```
> IQR(x)  # interquartile range
[1] 1.3
```

在某些情況下我們需要計算去掉某些較小值及較大值之後的平均數。比方說先去掉 10% 之較小值及較大值之後，再來求平均數。我們可以使用下列的指令：

```
> mean(x, trim = 0.1)
[1] 5.808333
```

要了解一群數字離中心點之集中程度可以計算樣本標準差：

```
> sd(x)
[1] 0.8280661
```

樣本標準差與樣本平均數之比值稱為變異係數 (coefficient of variation, CV)：

```
> cv <- sd(x) / mean(x)
> cv
[1] 0.1417113
```

請注意 CV 是一個沒有單位的數字。CV 值大代表資料的變化量是相對地大。

藉由一些有意義的圖形來了解資料之分佈是一個更直覺的作法。底下我們介紹一些常用來了解資料分佈的圖形。請參閱表 6.1.1。

表 6.1.1：一些一維資料的繪圖函數

method	in {graphics}	in {package}
scatterplot	plot	plot {lattice}
add regression line to plot	abline	
add reference line to plot	abline	
reference curve	curve	
box plot (box-and-whisker plot)	boxplot	bwplot {lattice}
histogram	hist	truehist {MASS} histogram {lattice}

method	in {graphics}	in {package}
stem-and-leaf plot	stem	stem.leaf {aplpack} slider.stem.leaf {aplpack}
strip chart	stripchart	stripplot {lattice}
kernel density estimator	density	densityplot {lattice} bkde {KernSmooth} locpoly {KernSmooth}
empirical cumulative distribution function	ecdf	
plot empirical CDF	plot.ecdf	
normal QQ plot	qqnorm	qqmath {lattice}
QQ normal reference line	qqline	
bar plot	barplot	barchart {lattice} barplot2 {gplots}
Cleveland dot plot	dotchart	dotplot {lattice}
pie chart	pie	
QQ plot	qqplot	qqmath {lattice} qq {lattice}

首先來看看 x 的盒鬚圖 (box plot) (如圖 6.1.1)：

```
> win.graph(width = 4.5, height = 2.8, pointsize = 8)
> old.par <- par(mfrow = c(1, 2), mex = 0.5,
+   mar = c(4, 4, 3, 2) + 0.1)

> boxplot(x)
> rug(x, side = 4)  # Add a rug to the plot.
> boxplot(x, horizontal = TRUE)
> rug(x, side = 1)

> par(old.par)
```

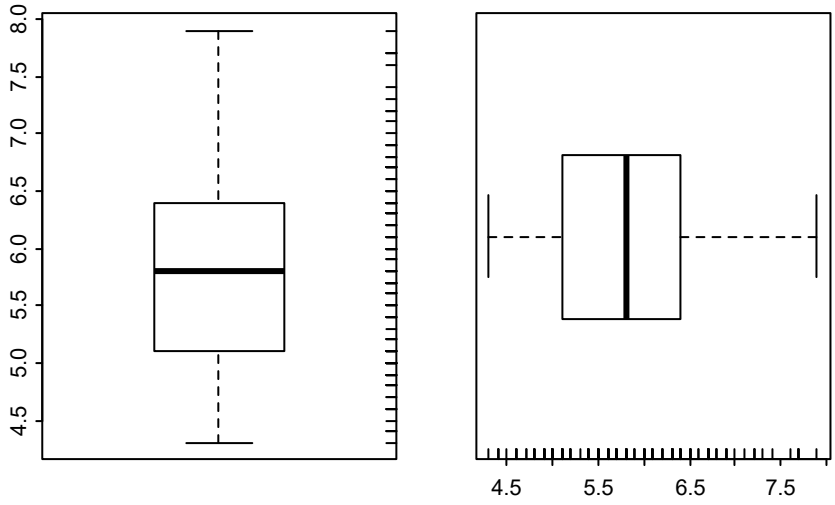

圖 6.1.1：盒鬚圖

在上面左圖長方盒部份，由下而上的三條線分別代表第一四分位數、第二四分位數或稱中位數 (粗線)、第三四分位數。若這個長方盒較小，則大部份資料較集中於中心點附近。在鬍鬚部份，最上面的那條線大約是最大值或是第三四分位數往上 1.5 倍四分位 (數間) 距之值；最下面的那條線亦同。若有某個資料點落在最下面或是最上面的線外面，則此資料點將可能視為離群值 (outlier)。在目前所考慮的資料並無離群值。由於上面 (或右邊) 鬍鬚比下面 (或左邊) 來得長許多，因此此資料之分佈是向右偏斜 (skewed to the right)，即資料分佈之長尾巴是在右邊。

接下來看看 x 的直方圖 (histogram) (如圖 6.1.2)：

```
> win.graph(width = 4.5, height = 2.8, pointsize = 8)
> old.par <- par(mfrow = c(1, 2), mex = 0.8,
+   mar = c(5, 5, 3, 1) + 0.1)

> hist(x, freq = TRUE, breaks = "Sturges")
> rug(x, side = 1)
> hist(x, prob = TRUE, breaks = "Sturges",
+   col = "lightblue", border = "magenta")
> rug(x, side = 1)
```

```
> par(old.par)
```

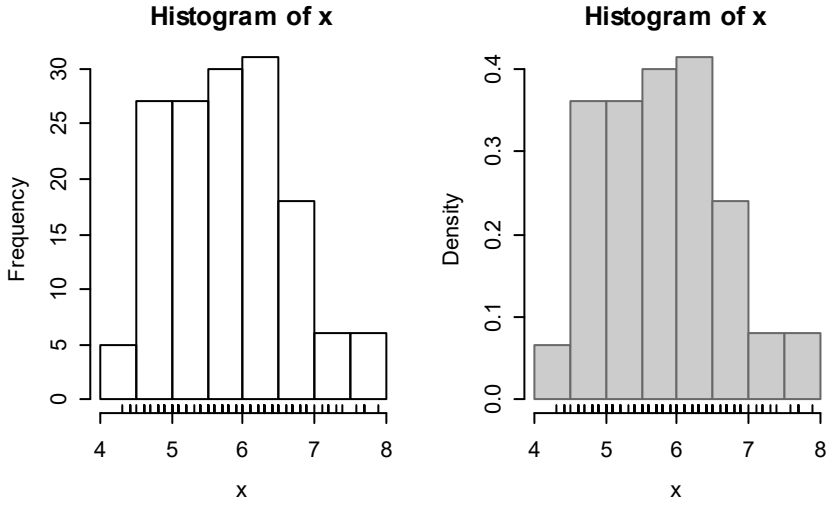

圖 6.1.2：直方圖

左邊圖形的縱座標是以數據落在某個區間 (bin) 之次數為數值，而右邊圖形的縱座標是以比率為數值。右邊圖形可以做為此資料機率密度函數 (probability density function, PDF) 的一個估測。由直方圖與盒鬚圖所得到在資料分佈的結論差不多。

請注意下面的兩個指令是一樣的：

```
hist(x, prob = TRUE, ...)
hist(x, freq = FALSE, ...)
```

在決定一個直方圖時一件很重要的事是區間大小 (binwidth) 或是那些斷點 (break points) 之設定。在上面的的函數 hist() 有三種方法可以選擇：

```
hist(x, prob = TRUE, breaks = "Sturges")
hist(x, prob = TRUE, breaks = "Scott")
hist(x, prob = TRUE, breaks = "Freedman-Diaconis")
```

內定的方法是 "Sturges"。詳情請參閱參考文獻 [Rizzo, 2008]。我們也可以直接設定要 10 個區間：

```
hist(x, prob = TRUE, breaks = 10)
```

另外我們也可以直接設定斷點之位置：

```
hist(x, prob = TRUE, breaks = seq(from = 4, to = 8, by = 0.25))
```

我們也可以使用 truehist{MASS} 來繪出直方圖 (如圖 6.1.3)：

```
> library(MASS)

> win.graph(width = 4.5, height = 2.8, pointsize = 8)
> old.par <- par(mfrow = c(1, 2), mex = 0.8,
+   mar = c(5, 5, 3, 1) + 0.1)

> truehist(x, prob = FALSE, ylab = "Frequency",
+   main = "Histogram")
> truehist(x, prob = TRUE, ylab = "Density",
+   main = "Histogram")

> par(old.par)
```

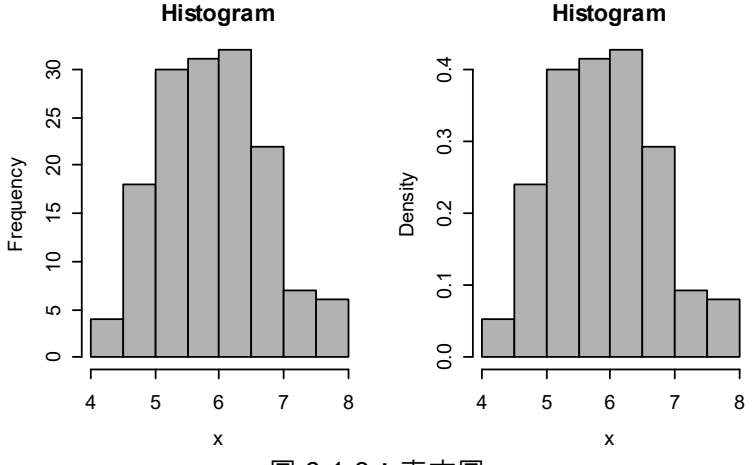

圖 6.1.3：直方圖

在 truehist() 中區間大小之決定方法 (nbins) 有 "Scott" 和 "Freedman-Diaconis"，並無 "Sturges"。

再來看看 x 的莖葉圖 (stem-and-leaf plot)：

```
> stem(x, scale = 0.5)

  The decimal point is at the |

  4 | 3444
  4 | 566667788888999999
  5 | 00000000001111111122223444444
  5 | 55555556666667777777888888999
  6 | 00000011111122223333333334444444
  6 | 5555566777777778889999
  7 | 0122234
  7 | 677779
```

在上面的指令中，我們設定區間之大小為 0.5，數據出現幾次就寫幾次。比方說，4.4 這個數字出現了 3 次：

```
> sum(x == 4.4)
[1] 3
```

我們也可觀察到 x 之資料大部份集中於 5.0 ~ 6.5 之間。

接下來看看 x 的長條記錄圖 (strip chart) (我們使用了三種不同展示資料的方法) (如圖 6.1.4)：

```
> win.graph(width = 4.5, height = 3.3, pointsize = 8)
> old.par <- par(mex = 0.8, mar = c(5, 4, 3, 1) + 0.1)

> stripchart(x, method = "overplot", at = 0.7)
> text(6, 0.65, "overplot")
> stripchart(x, method = "stack", add = TRUE, at = 0.85)
> text(6, 0.8, "stack")
> stripchart(x, method = "jitter", add = TRUE, at = 1.2)
```

```
> text(6, 1.05, "jitter")
> title(main = "strip chart")

> par(old.par)
```

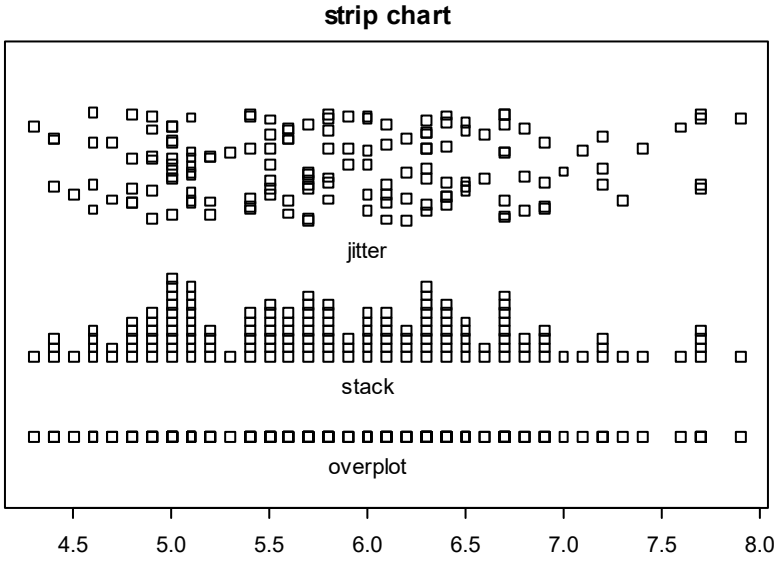

圖 6.1.4：長條記錄圖

由此圖可以很清楚地看出資料分佈之狀況。

假設我們將數據之範圍切為六等份，再來看看資料分佈在這六等份之分佈情況。我們可以使用圓餅圖 (pie chart) (如圖 6.1.5)：

```
> y <- cut(x, breaks = 6)
> z <- table(y)
> z
y
(4.3,4.9] (4.9,5.5] (5.5,6.1] (6.1,6.7] (6.7,7.3] (7.3,7.9]
       22        37        36        35        13         7

> windows(width = 4.5, height = 3.3, pointsize = 8)
> old.par <- par(mfrow = c(2, 2), mex = 0.2,
```

```
+   mar = c(3, 3, 3, 2) + 0.1)

> pie(z)  # default color
> pie(z, clockwise = TRUE)
> pie(z, col = terrain.colors(6))
> pie(z, col = gray(seq(from = 0.4, to = 1.0, length = 6)))

> par(old.par)
```

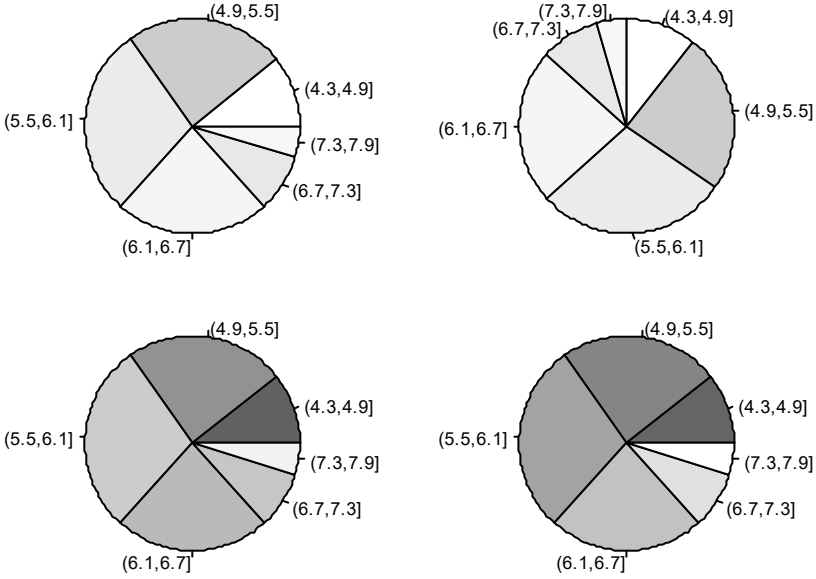

圖 6.1.5：圓餅圖

在上圖中，色塊愈大表示落於該等份之資料點愈多。

在前面曾提到過直方圖可以做為資料機率分佈的一個估測。事實上我們有更好的方式來估測資料機率分佈，這是屬於機率分佈估測 (density estimation) 之領域。我們可以用 density() 這個函數來求出一個核密度估計 (kernel density estimator, KDE) (如圖 6.1.6)：

```
> win.graph(width = 4.5, height = 2.8, pointsize = 8)
> old.par <- par(mfrow = c(1, 2), mex = 0.8,
+   mar = c(5, 4, 3, 1) + 0.1)

> plot(density(x), col = "red",
+   main = "Kernel density estimate")
> rug(x, side = 1)
> hist(x, prob = TRUE, breaks = "Sturges",
+   main = "Histogram and KDE")
> lines(density(x), col = "red")
> rug(x, side = 1)

> par(old.par)
```

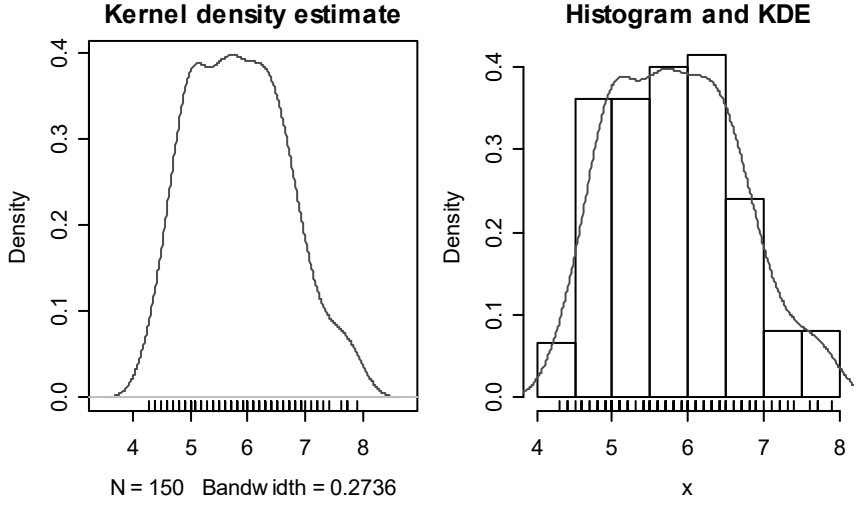

圖 6.1.6：核密度估計

在求 KDE 時，我們得先選定使用何種核函數 (kernel function) $K(\cdot)$ 及其帶寬 (bandwidth) h。假設我們有一組數據 $x_1, ..., x_n$，則 KDE 在 x 點之值為

$$\hat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right).$$

常見的核函數有：

uniform (box, rectangular): $K(u)=\frac{1}{2}I(|u|\le 1)$

triangle: $K(u)=(1-|u|)I(|u|\le 1)$

Epanechnikov: $K(u)=\frac{3}{4}(1-u^2)I(|u|\le 1)$

quartic (biweight): $K(u)=\frac{15}{16}(1-u^2)^2 I(|u|\le 1)$

triweight: $K(u)=\frac{35}{32}(1-u^2)^3 I(|u|\le 1)$

cosine: $K(u)=\frac{\pi}{4}\cos\left(\frac{\pi}{2}u\right)I(|u|\le 1)$

Tukey-Hanning: $K(u)=\frac{1}{2}[1+\cos(\pi u)]I(|u|\le 1)$

normal (Gaussian): $K(u)=\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}u^2\right)$

$$I(|u|\le 1):=\begin{cases}1, & |u|\le 1,\\ 0, & |u|>1.\end{cases}$$

在目前 R 中可以選擇的核函數為

```
kernel = c("gaussian", "epanechnikov", "rectangular",
"triangular", "biweight", "cosine", "optcosine")
```

內定的核函數為 "gaussian"。帶寬的設定參數是 bw。

我們可以將 density() 之結果存起來，試試下面的指令：

```
> f <- density(x)
> class(f)
> names(f)
```

```
> print(f)

> plot(f, type = "n")
> polygon(f, col = "wheat")
```

有關 KDE 更多的詳情可參閱參考文獻 [Härdle, Müller, Sperlich, and Werwatz, 2004][Rizzo, 2008]。

經驗累積分佈函數 (empirical cumulative distribution function, ECDF) 是累積分佈函數 (cumulative distribution function, CDF) 的一個估測 (詳見第七章)。以 x 為例 (如圖 6.1.7)：

```
> win.graph(width = 4.5, height = 3.3, pointsize = 8)
> old.par <- par(mex = 0.8, mar = c(5, 4, 3, 1) + 0.1)
> plot.ecdf(x)
> par(old.par)
```

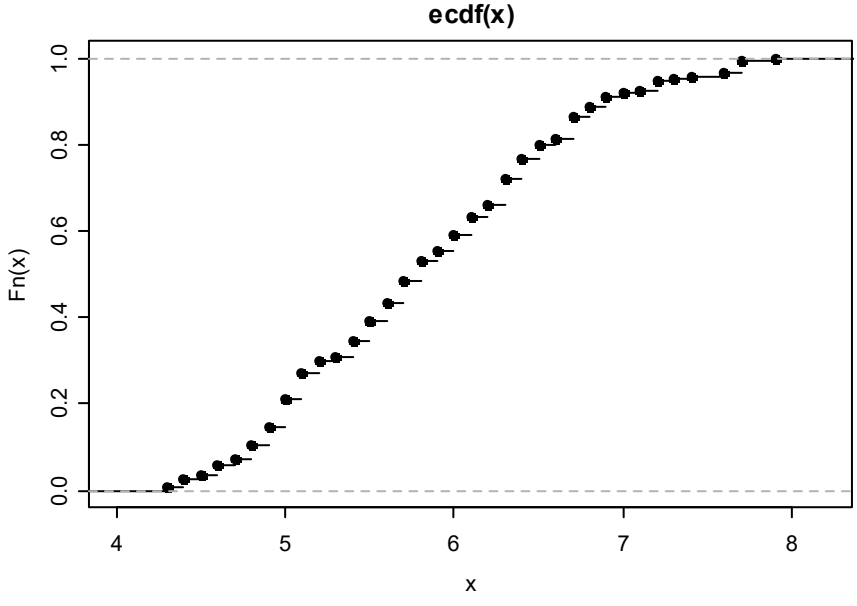

圖 6.1.7：經驗累積分佈函數

事實上 ECDF 本身也是一個 CDF。在 ECDF 的定義中, 每個資料點的機率質量皆相同。因此若某個資料點有重覆，則該點的機率質量就會累加。至於為何每條線段的最左邊皆有黑點是因為 ECDF (如同一般之 CDF) 是一個右連續函數 (right continuous function)。

我們可以將 ecdf() 之結果存起來，試試下面的指令：

```
> F <- ecdf(x)
> class(F)
> names(F)

> print(F)
> summary(F)

> plot(F)
```

要判斷我們是否可以合理地假設一組數據是來自常態分佈可以使用常態機率圖 (normal QQ plot, normal probability plot)。以 x 為例 (如圖 6.1.8)：

```
> win.graph(width = 4.5, height = 3.3, pointsize = 8)
> old.par <- par(mex = 0.8, mar = c(5, 4, 3, 1) + 0.1)
> qqnorm(x)
> qqline(x, col = "red", lwd = 2)
> par(old.par)
```

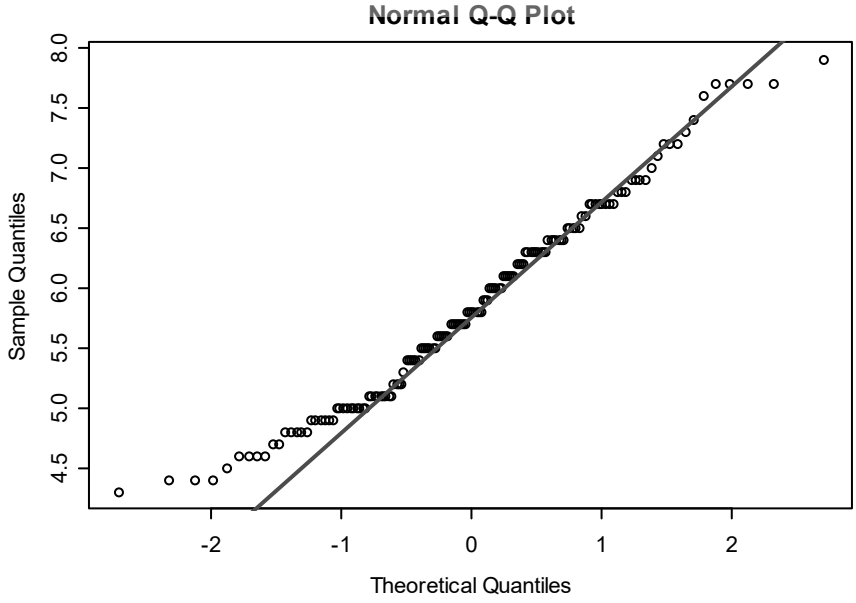

圖 6.1.8：常態機率圖

上圖之縱軸是給定數據之分位數，橫軸是常態分佈的分位數，而直線通過第一及第三四分位數。若絕大部份的點皆集中於這條線的附近，則我們可以合理地假設這一組數據是來自常態分佈。顯然上圖中之 x 並不是來自常態分佈。

考慮 VADeaths{datasets} 這個資料集。這組資料是在 1940 年美國維吉尼亞州死亡人數比率 (以千人為單位) 的統計資料。此資料以年齡層及居住於鄉村或城市之男女為分類的標準。我們先將其載入：

```
> data(VADeaths)
```

接下來看一下資料的內容：

```
> str(VADeaths)

num [1:5, 1:4] 11.7 18.1 26.9 41 66 8.7 11.7 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:5] "50-54" "55-59" "60-64" "65-69" ...
  ..$ : chr [1:4] "Rural Male" "Rural Female"
```

```
                  "Urban Male" "Urban Female"

> class(VADeaths)
[1] "matrix"

> VADeaths

      Rural Male Rural Female Urban Male Urban Female
50-54       11.7          8.7       15.4          8.4
55-59       18.1         11.7       24.3         13.6
60-64       26.9         20.3       37.0         19.3
65-69       41.0         30.9       54.6         35.1
70-74       66.0         54.3       71.1         50.0
```

我們可以使用長條圖 (bar plot) 來展示這組資料 (如圖 6.1.9)：

```
> names <- c("RM", "RF", "UM", "UF")
> colors <- c("lightblue", "mistyrose", "lightcyan",
+   "lavender", "cornsilk")

> windows(width = 4.5, height = 3.3, pointsize = 8)
> old.par <- par(mfrow = c(2, 2), mex = 0.8,
+   mar = c(3, 3, 3, 2) + 0.1)

> barplot(VADeaths, names.arg = names)
> barplot(VADeaths, names.arg = names, horiz = TRUE)
> barplot(VADeaths, names.arg = names, col = colors,
+   border = "blue")
> barplot(VADeaths, names.arg = names, col = colors,
+   border = "blue", space = 1.5)

> par(old.par)
```

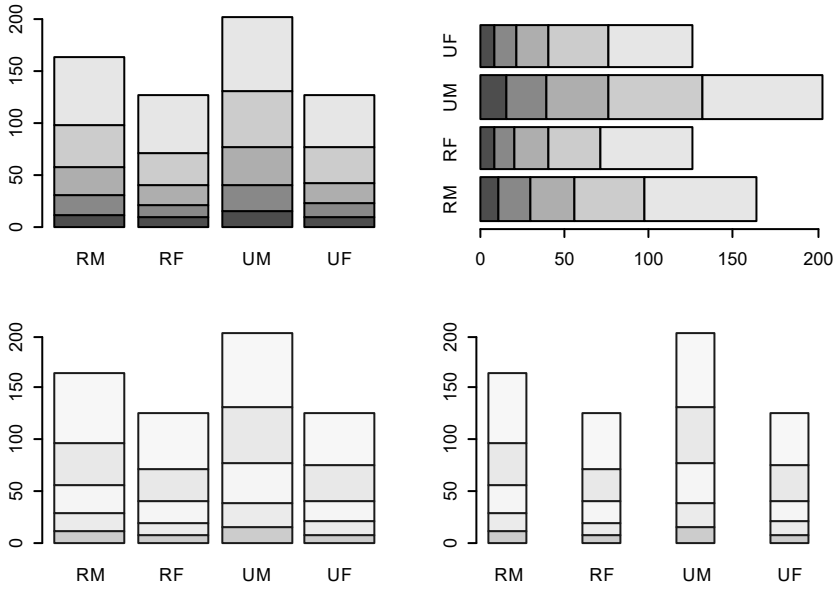

圖 6.1.9：長條圖

我們可以畫更棒一些的圖 (如圖 6.1.10)：

```
> win.graph(width = 4.5, height = 3.3, pointsize = 8)
> old.par <- par(mex = 0.8, mar = c(5, 4, 3, 1) + 0.1)

> barplot(VADeaths, beside = TRUE,
+   col = c("lightblue", "mistyrose", "lightcyan",
+   "lavender", "cornsilk"),
+   legend.text = rownames(VADeaths), ylim = c(0, 100))
> title(main = "Death Rates in Virginia", font.main = 4)

> par(old.par)
```

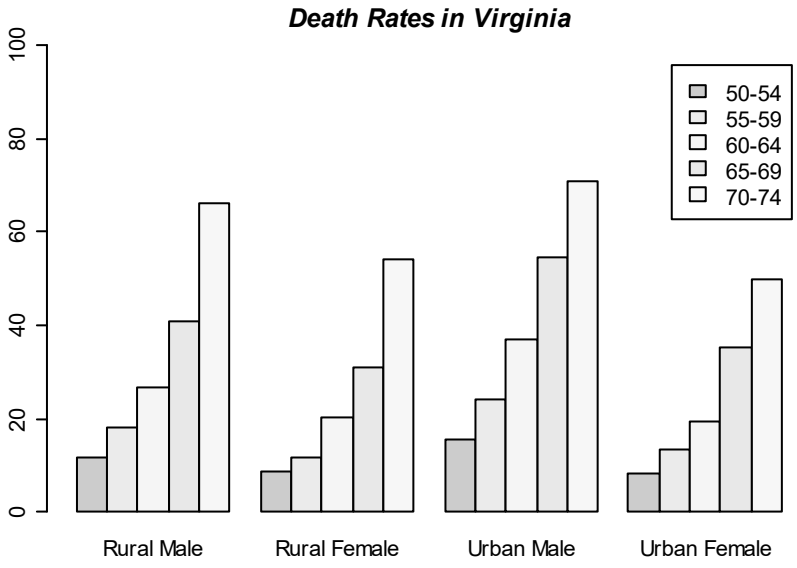

圖 6.1.10：更詳細的長條圖(一)

或是 (如圖 6.1.11)：

```
> win.graph(width = 4.5, height = 3.3, pointsize = 8)
> old.par <- par(mex = 0.8, mar = c(5, 4, 3, 1) + 0.1)

> barplot(t(VADeaths), beside = TRUE,
+   col = c("lightblue", "mistyrose", "lightcyan",
+   "lavender"), legend.text = rownames(t(VADeaths)),
+   ylim = c(0, 80), args.legend = list(x = "topleft"))
> title(main = "Death Rates in Virginia", font.main = 4)

> par(old.par)
```

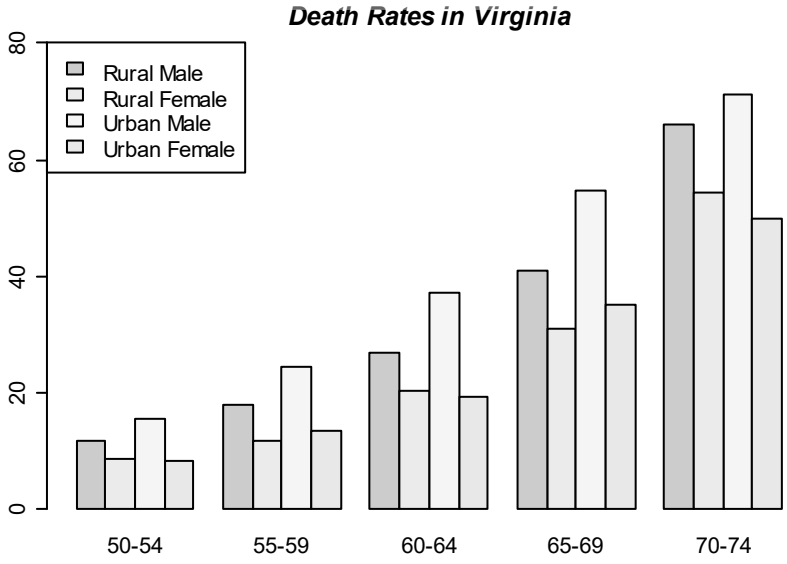

圖 6.1.11：更詳細的長條圖(二)

由圖中可看出不管是居住於鄉村或城市之男女，年齡層愈高死亡人數愈多；這沒什麼好驚訝的。不管是居住於鄉村或城市的男性似乎都比女性的死亡人數多，但在低年齡層這個情況較不明顯。

同樣的結論我們也可以使用點圖 (Cleveland dot plot) 得到 (如圖 6.1.12)：

```
> colnames(VADeaths) <- c("RM", "RF", "UM", "UF")

> win.graph(width = 4.5, height = 3.3, pointsize = 8)
> old.par <- par(mfrow = c(1, 2), mex = 0.8,
+   mar = c(5, 4, 3, 1) + 0.1)

> dotchart(VADeaths, xlim = c(0, 100), xlab =
+   "Deaths per 1000", main = "Death rates")
> dotchart(t(VADeaths), xlim = c(0,100), xlab =
+   "Deaths per 1000", main = "Death Rates")

> par(old.par)
```

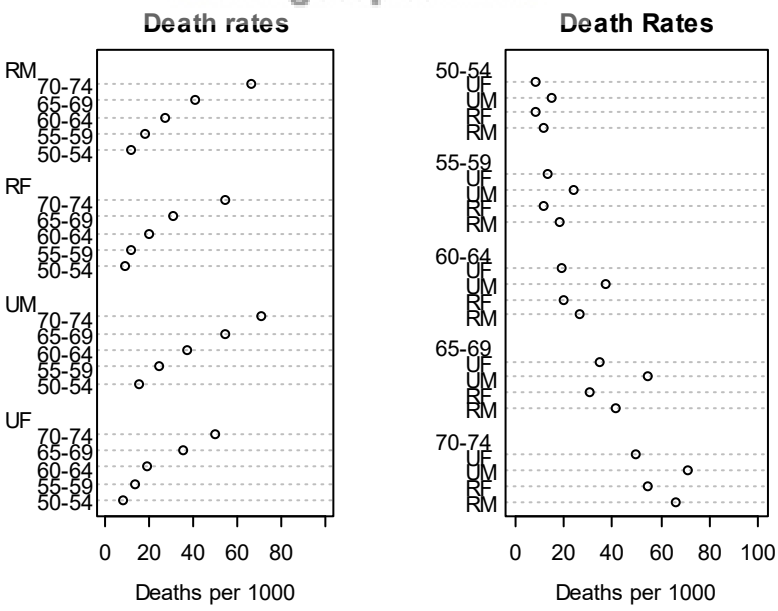

圖 6.1.12：點圖

考慮 warpbreaks{datasets} 這個資料集，是毛線在織布機編織過程中斷線的次數。我們先將其載入並看一下簡要的內容：

```
> data(warpbreaks)

> str(warpbreaks)

'data.frame':   54 obs. of  3 variables:
 $ breaks : num  26 30 54 25 70 52 51 26 67 18 ...
 $ wool   : Factor w/ 2 levels "A","B": 1 1 1 1 1 ...
 $ tension: Factor w/ 3 levels "L","M","H": 1 1 1 1 ...
```

這一個資料集共有 3 個變量及 54 筆量測數據。3 個變量分別為毛線斷線次數 (breaks)、毛線種類 (wool) (共有 AB 二種)、及張力 (tension) (共有 LMH 三種)。

假設我們想知道不同毛線種類之斷線次數可使用下列的指令：

```
> with(warpbreaks, tapply(breaks, INDEX = wool,
+   FUN = sum))
  A   B
838 682
```

假設我們想知道不同張力下之斷線次數可使用下列的指令：

```
> with(warpbreaks, tapply(breaks, INDEX = tension,
+   FUN = sum))
  L   M   H
655 475 390
```

假設我們想知道不同毛線種類及不同張力下之斷線次數可使用下列的指令：

```
> with(warpbreaks, tapply(breaks, INDEX =
+   list(wool, tension), FUN = sum))
    L   M   H
A 401 216 221
B 254 259 169
```

上面的結果也可以由下列的指令得到：

```
> xtabs(breaks ~ wool, data = warpbreaks)
> xtabs(breaks ~ tension, data = warpbreaks)
> xtabs(breaks ~ wool + tension, data = warpbreaks)
> ftable(xtabs(breaks ~ wool + tension,
+   data = warpbreaks))
```

我們可以繪出一些有關毛線斷線次數及其平均值之長條圖 (如圖 6.1.13)：

```
> t1 <- with(warpbreaks, tapply(breaks, INDEX =
+   list(wool, tension), FUN = sum))
> t2 <- with(warpbreaks, tapply(breaks, INDEX =
+   list(wool, tension), FUN = mean))
```

```
> win.graph(width = 4.5, height = 2.8, pointsize = 8)
> old.par <- par(mfrow = c(1, 2), mex = 0.8,
+   mar = c(5, 4, 3, 1) + 0.1)

> barplot(t1, beside = TRUE,
+   col = c("lightblue", "mistyrose"), main = "counts",
+   legend.text = rownames(t1), ylim = c(0, max(t1)))

> barplot(t2, beside = TRUE,
+   col = c("lightblue", "mistyrose"), main = "means",
+   legend.text = rownames(t2), ylim = c(0, max(t2)))

> par(old.par)
```

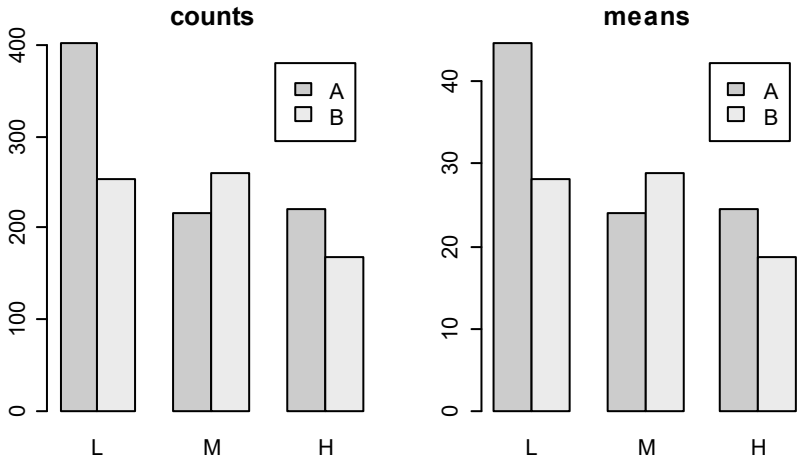

圖 6.1.13：長條圖

由上圖可知，在張力為中等 (M) 時，毛線 A 在斷線次數及其平均值皆比毛線 B 來得少。

現在來看看在不同毛線種類及不同張力下之斷線次數之圓餅圖 (如圖 6.1.14)：

```
> brks <- as.integer(xtabs(breaks ~ wool + tension,
+   data = warpbreaks))
> label <- c("AL", "BL", "AM", "BM", "AH", "BH")

> windows(width = 4.5, height = 2.8, pointsize = 8)
> old.par <- par(mfrow = c(1, 2), mex = 0.2,
+   mar = c(3, 3, 3, 2) + 0.1)

> pie(brks, label = label)
> pie(brks, label = label, col = gray(seq(from = 0.4,
+   to = 1.0, length = 8)))

> par(old.par)
```

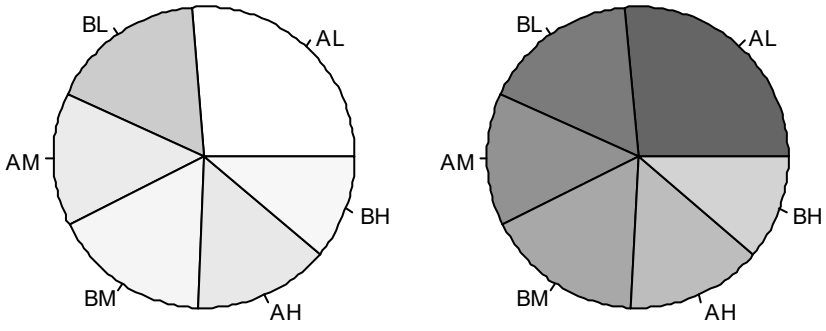

圖 6.1.14：圓餅圖

一般而言，長條圖比圓餅圖更容易展示出資料的特性。

6.2 多維資料

假設我們想觀察好幾個變量之間的關係。在 R 中提供了一些多維資料的繪圖函數。請參閱表 6.2.1。

表 6.2.1：一些多維資料的繪圖函數

method	in {graphics}	in {package}
matrix of scatterplots (scatterplot matrix)	pairs	splom {lattice} clPairs {mclust} pairs2 {TeachingDemos}
3D scatterplot		cloud {lattice} scatterplot3d {scatterplot3d} plot3d {rgl}
perspective plot, surface plot	persp	wireframe {lattice} persp3d {rgl}
contour plot	contour image filled.contour contourLines {grDevices}	contourplot {lattice} levelplot {lattice}
2D kernel density estimator		bkde2D {KernSmooth}
2D histogram		hexbin {hexbin} hist2d {gplots}
mosaic plot	mosaicplot	
parallel coordinate plot		parallel {lattice} parcoord {MASS}
star plot	stars	
segment plot	stars	
interactive 3D graphics		{rggobi} {rgl}

若我們想看看 iris{datasets} 資料集中花萼長度、花萼寬度、花瓣長度、花瓣寬度彼此之間的關係可以使用二維的散點圖 (散佈圖) 矩陣 (scatterplot matrix)。吾人可以使用下面的指令 (如圖 6.2.1)：

```
> win.graph(width = 4.5, height = 3.3, pointsize = 8)
> old.par <- par(mex = 0.8, mar = c(5, 4, 3, 1) + 0.1)
> pairs(iris[, 1:4], panel = panel.smooth)
> par(old.par)
```

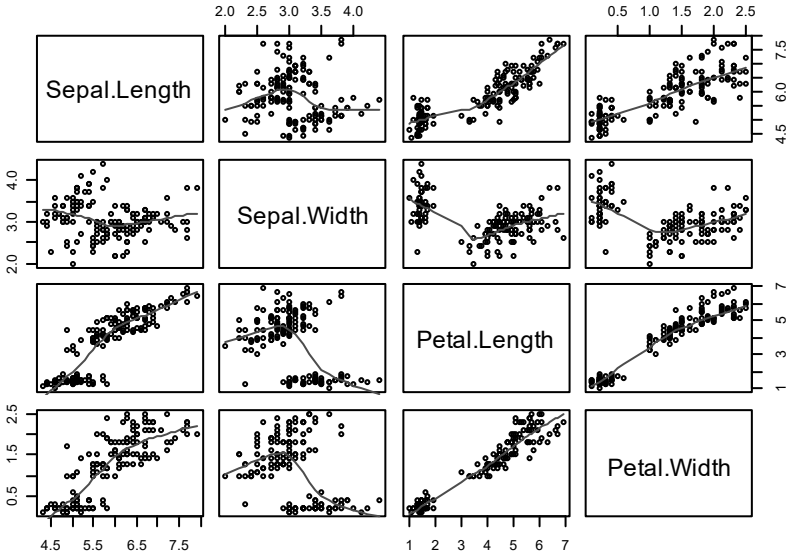

圖 6.2.1：散點圖矩陣

左下圖之橫座標為 Sepal.Length (花萼長度)，縱座標為 Petal.Width (花瓣寬度)。由上圖可發現 Petal.Length (花瓣長度) 與 Petal.Width (花瓣寬度) 似乎具有線性關係，但 Sepal.Length (花萼長度) 與 Sepal.Width (花萼寬度) 之間的關係顯得較為複雜。

上圖的對角線只放了變數名稱而已。在這些地方我們可以畫上該變量之直方圖及核密度估計等相關之圖形。但我們必須要定義一個繪圖函數才能繪圖 (如圖 6.2.2)：

```
> panel.hist <- function(x, ...) {
+   usr <- par("usr"); on.exit(par(usr))
+   par(usr = c(usr[1:2], 0, 1.5))
+   h <- hist(x, plot = FALSE)
```

```
+   breaks <- h$breaks; nB <- length(breaks)
+   y <- h$counts; y <- y / max(y)
+   rect(breaks[-nB], 0, breaks[-1], y, col = "cyan", ...)
+   lines(density(x, na.rm = TRUE), col = "red")
+}

> win.graph(width = 4.5, height = 3.3, pointsize = 8)
> old.par <- par(mex = 0.8, mar = c(5, 4, 3, 1) + 0.1)

> pairs(iris[, 1:4], panel = panel.smooth, pch = 1,
+   bg = "lightcyan", diag.panel = panel.hist,
+   font.labels = 2, cex.labels = 1.2)

> par(old.par)
```

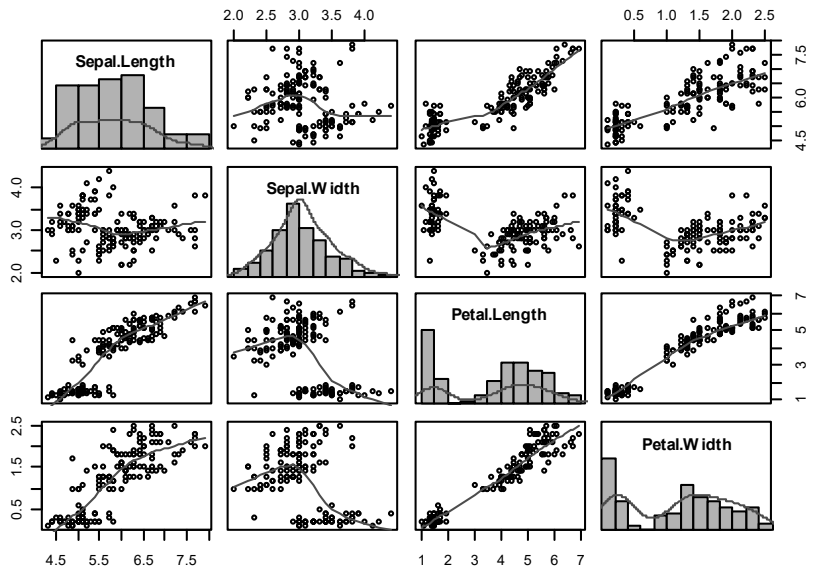

圖 6.2.2：散點圖矩陣(對角線有直方圖及核密度估計)