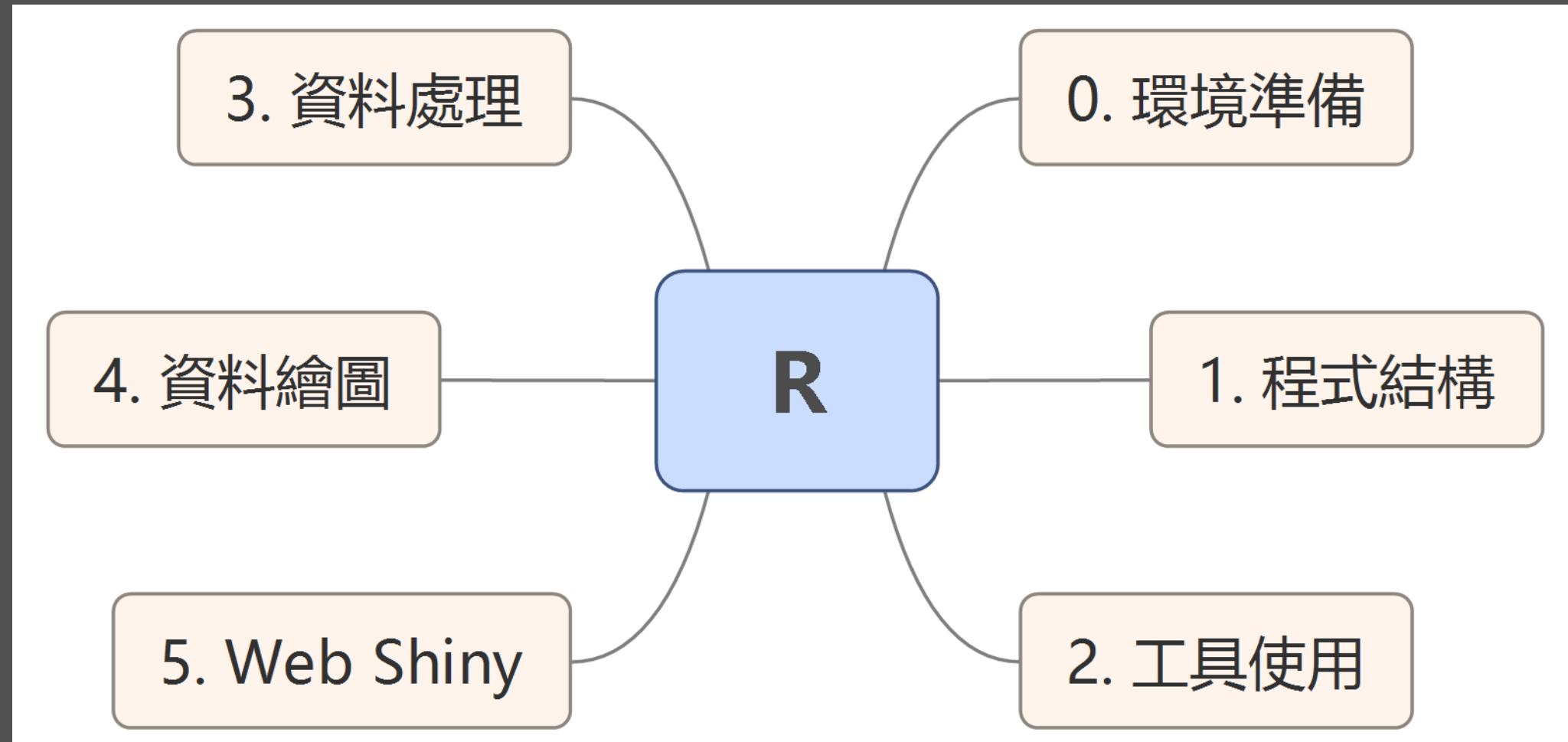


R 資料繪圖

目的

在此說明 R 的基礎視覺化方案



R 內建繪圖功能

使用 ggplot2 中的 diamonds 的資料集

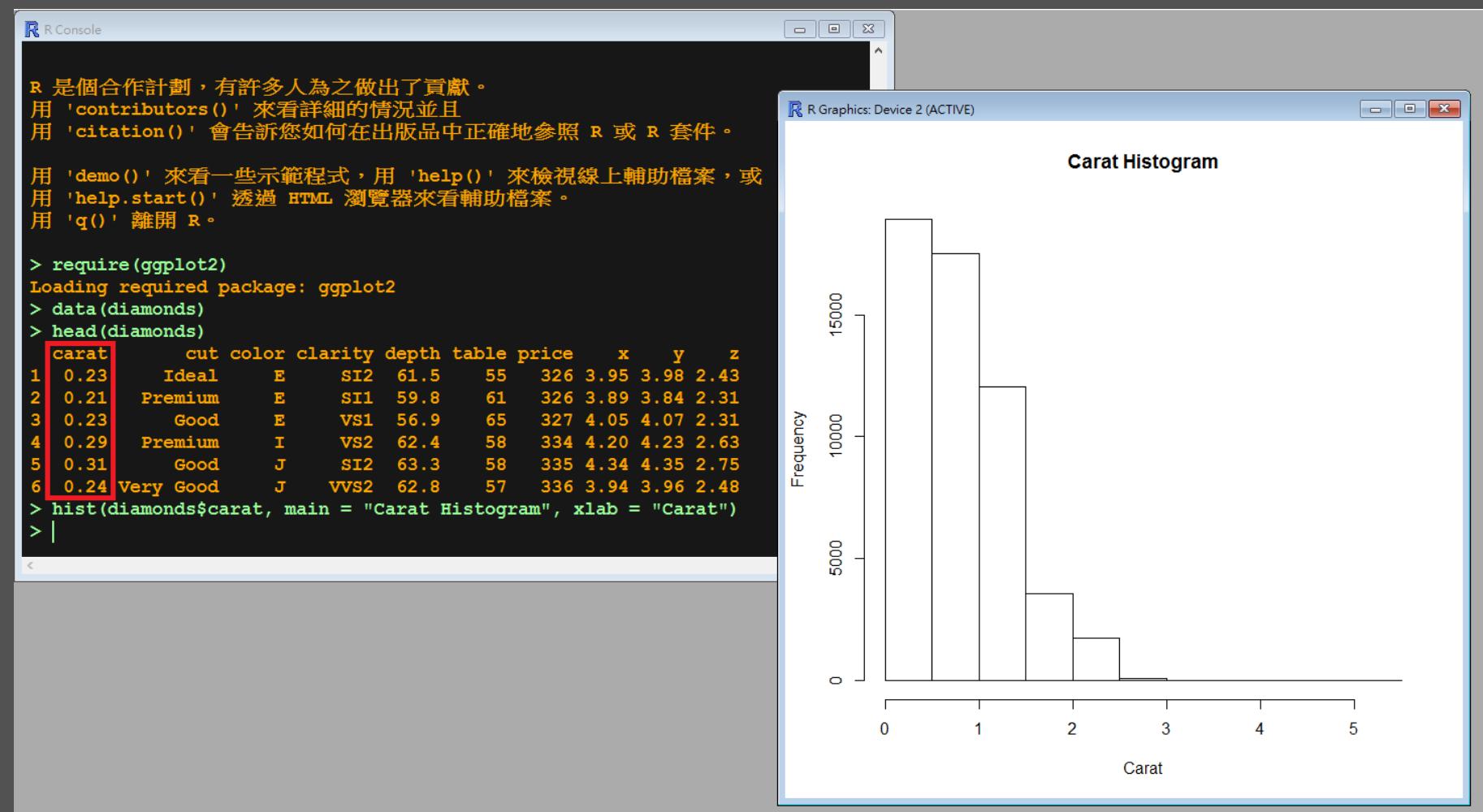
```
require(ggplot2)  
data(diamonds)  
head(diamonds)
```

```
> require(ggplot2)  
Loading required package: ggplot2  
> data(diamonds)  
> head(diamonds)  
   carat       cut color clarity depth table price     x     y     z  
1  0.23     Ideal    E     SI2   61.5     55   326 3.95 3.98 2.43  
2  0.21 Premium    E     SI1   59.8     61   326 3.89 3.84 2.31  
3  0.23      Good    E     VS1   56.9     65   327 4.05 4.07 2.31  
4  0.29 Premium    I     VS2   62.4     58   334 4.20 4.23 2.63  
5  0.31      Good    J     SI2   63.3     58   335 4.34 4.35 2.75  
6  0.24 Very Good   J     VVS2   62.8     57   336 3.94 3.96 2.48  
> |
```

R 直方圖

```
hist(diamonds$carat, main = "Carat Histogram", xlab = "Carat")
```

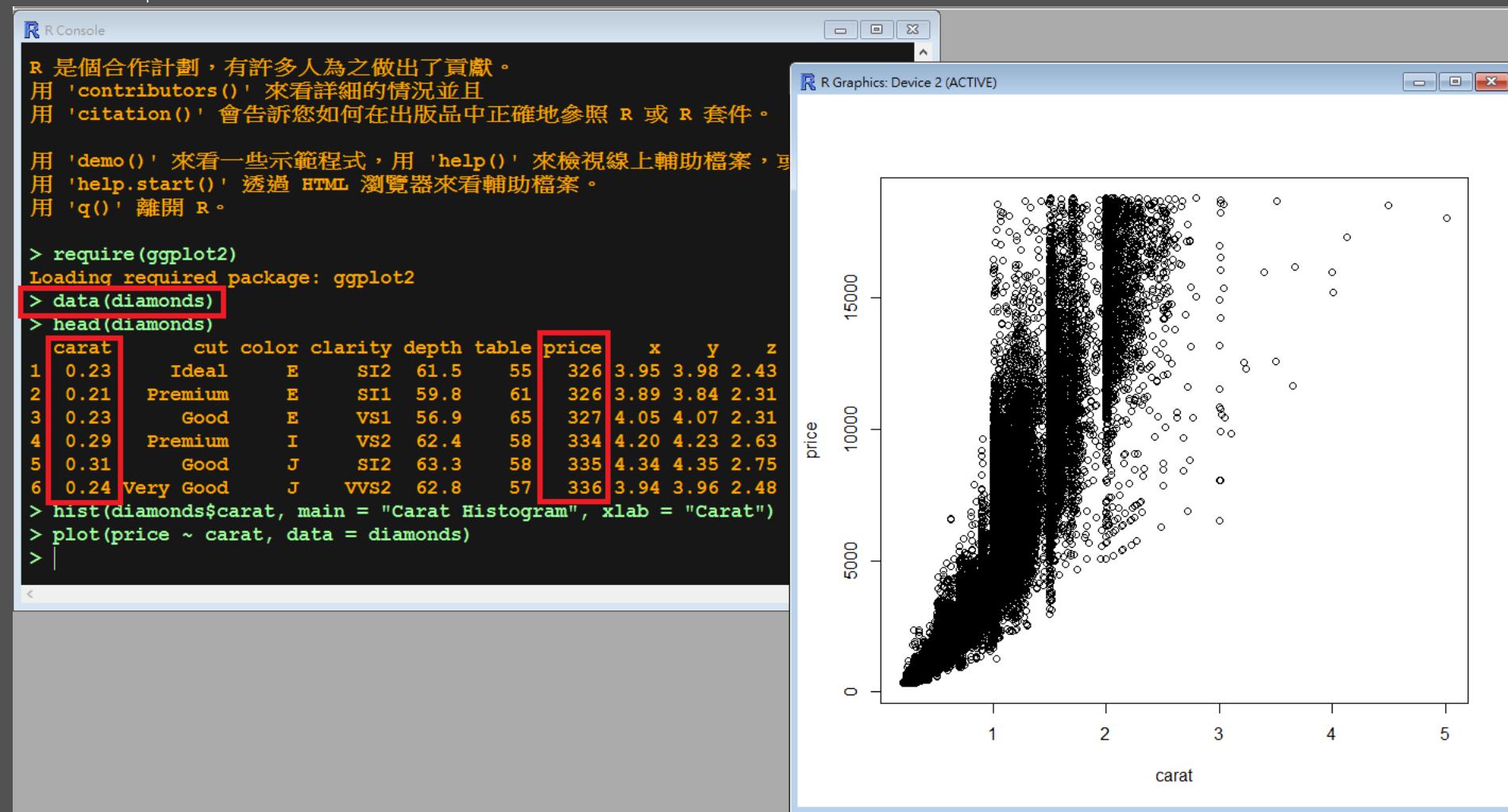
單變量最常用的圖，其函數為 hist()，diamonds\$carat 為匯入的 diamonds - 鑽石資料中選擇 carat 克拉的欄位，main、xlab 兩者皆為"引數"，main 為整圖的主旨，xlab 為 x 軸的名稱。



R 散佈圖

```
plot(price ~ carat, data = diamonds)
```

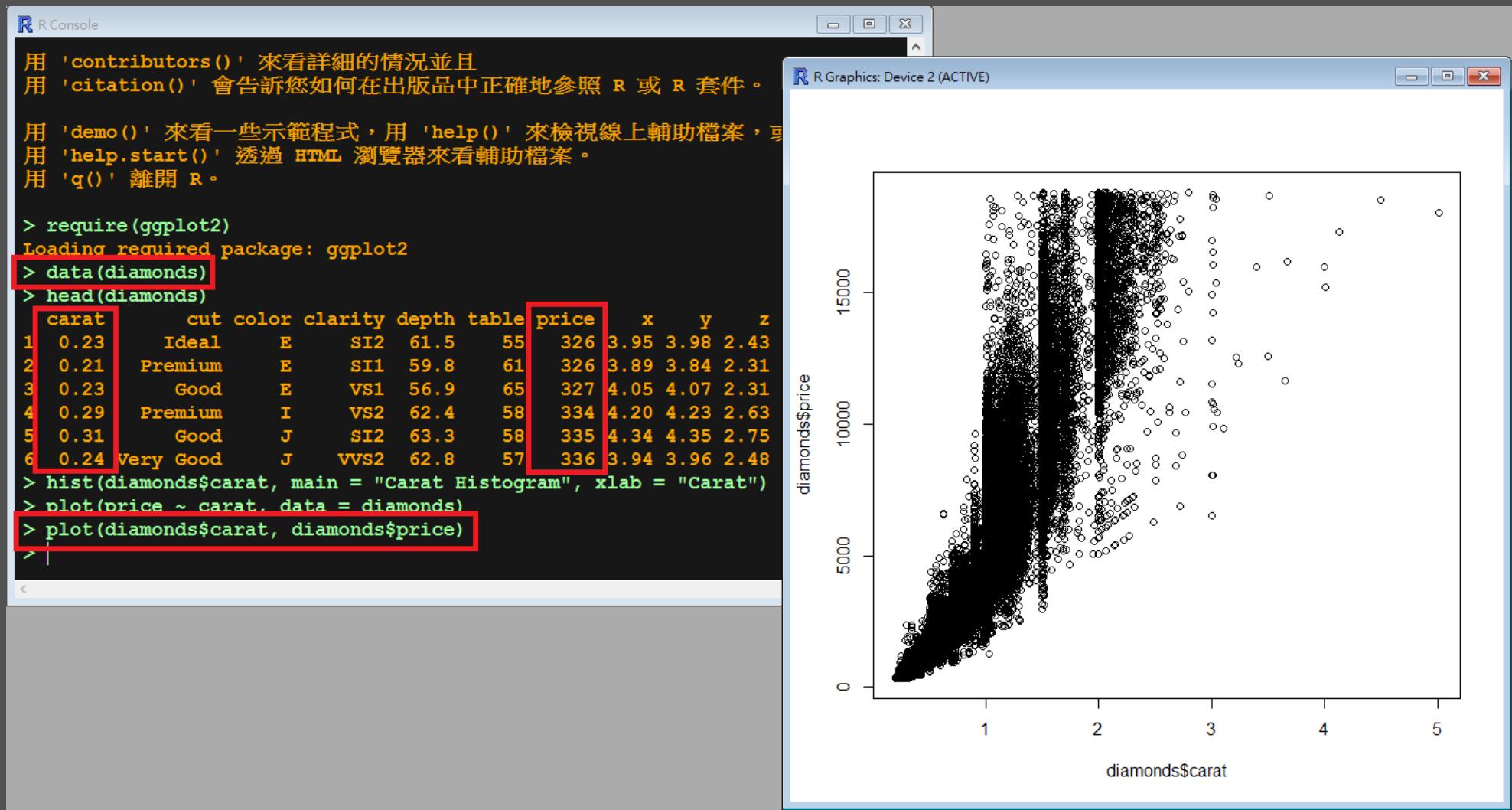
其特性為兩個變數的比較 X 軸、Y 軸，各自代表一個變數，其使用的函數為 plot()。"~" 為 formula，做出價格 - price、重量 - carat 間的散佈關係圖!!!



R 散佈圖

plot(diamonds\$carat, diamonds\$price)

其當然也可以不用 data 的參數。

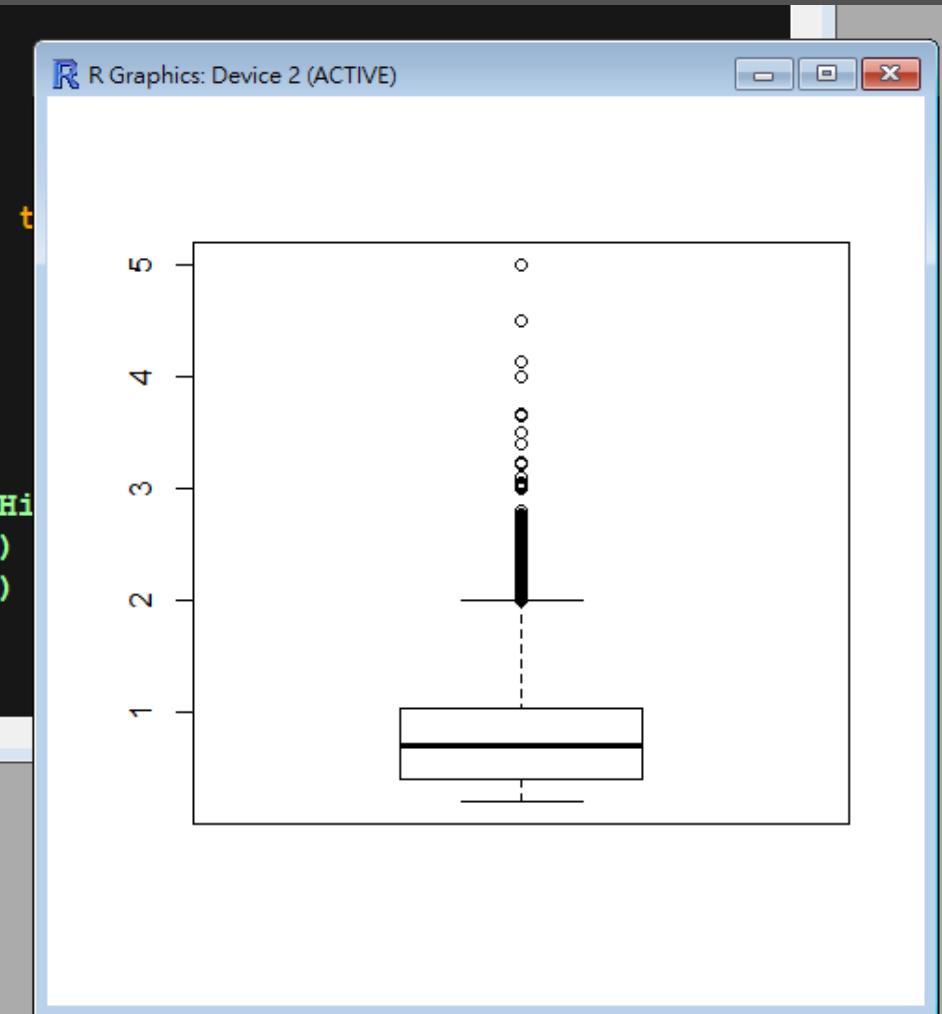


R 箱型圖

稱為 "盒鬚圖", 其函數 boxplot()。

boxplot(diamonds\$carat)

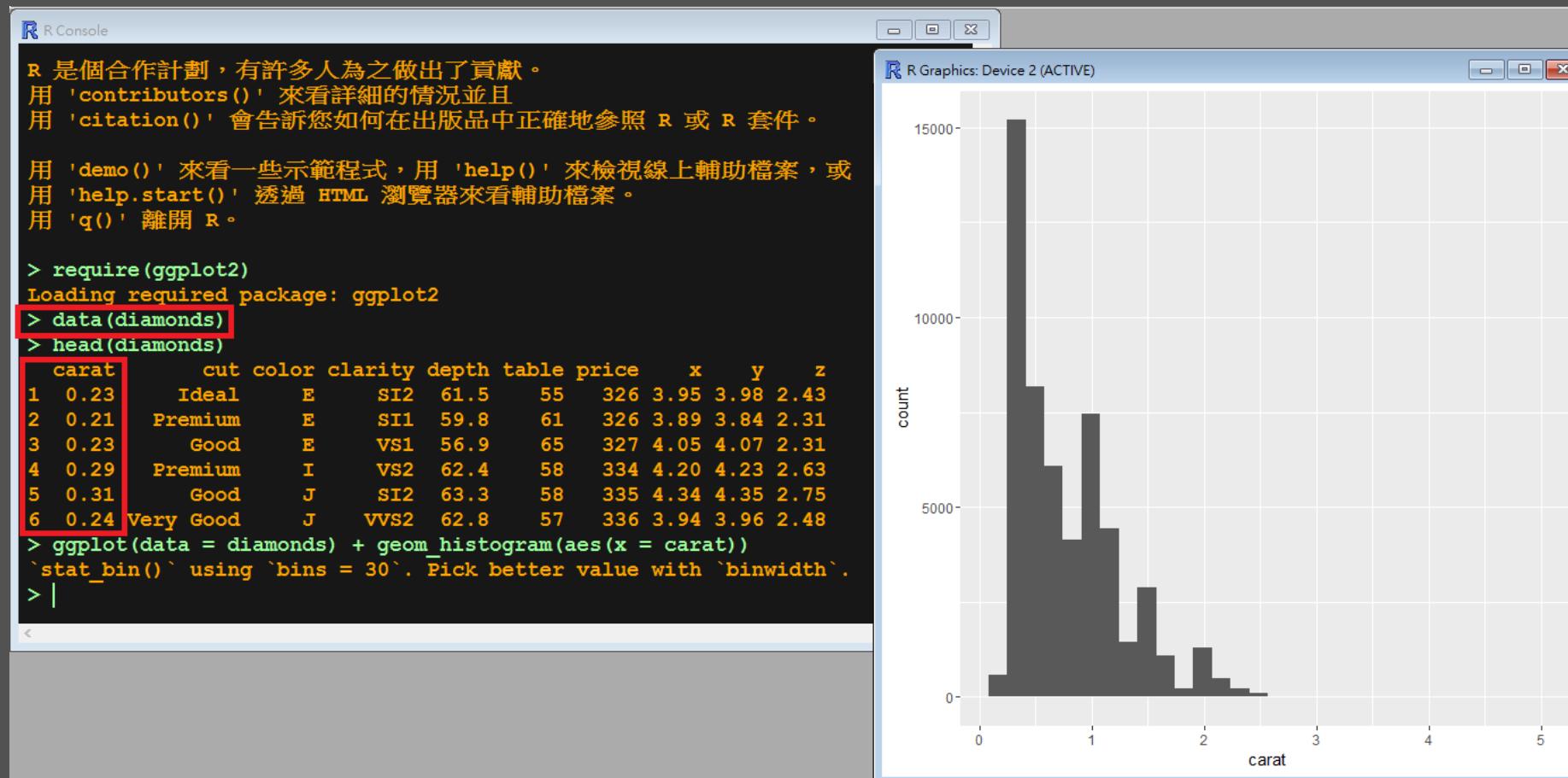
```
> require(ggplot2)
Loading required package: ggplot2
> data(diamonds)
> head(diamonds)
  carat       cut color clarity depth t
1 0.23      Ideal   E    SI2   61.5
2 0.21 Premium   E    SI1   59.8
3 0.23      Good   E    VS1   56.9
4 0.29 Premium   I    VS2   62.4
5 0.31      Good   J    SI2   63.3
6 0.24 Very Good   J    VVS2   62.8
> hist(diamonds$carat, main = "Carat Hi
> plot(price ~ carat, data = diamonds)
> plot(diamonds$carat, diamonds$price)
> boxplot(diamonds$carat)
>
```



R ggplot2 直方圖

ggplot() 為設定資料來源變數名稱， ggplot(data = [資料來源變數名稱]), geom_histogram() 為設定其為離散直方圖 並包含 aes() 函數， aes() 將其 x 軸 設為 carat 克拉。

ggplot(data = diamonds) + geom_histogram(aes(x = carat))

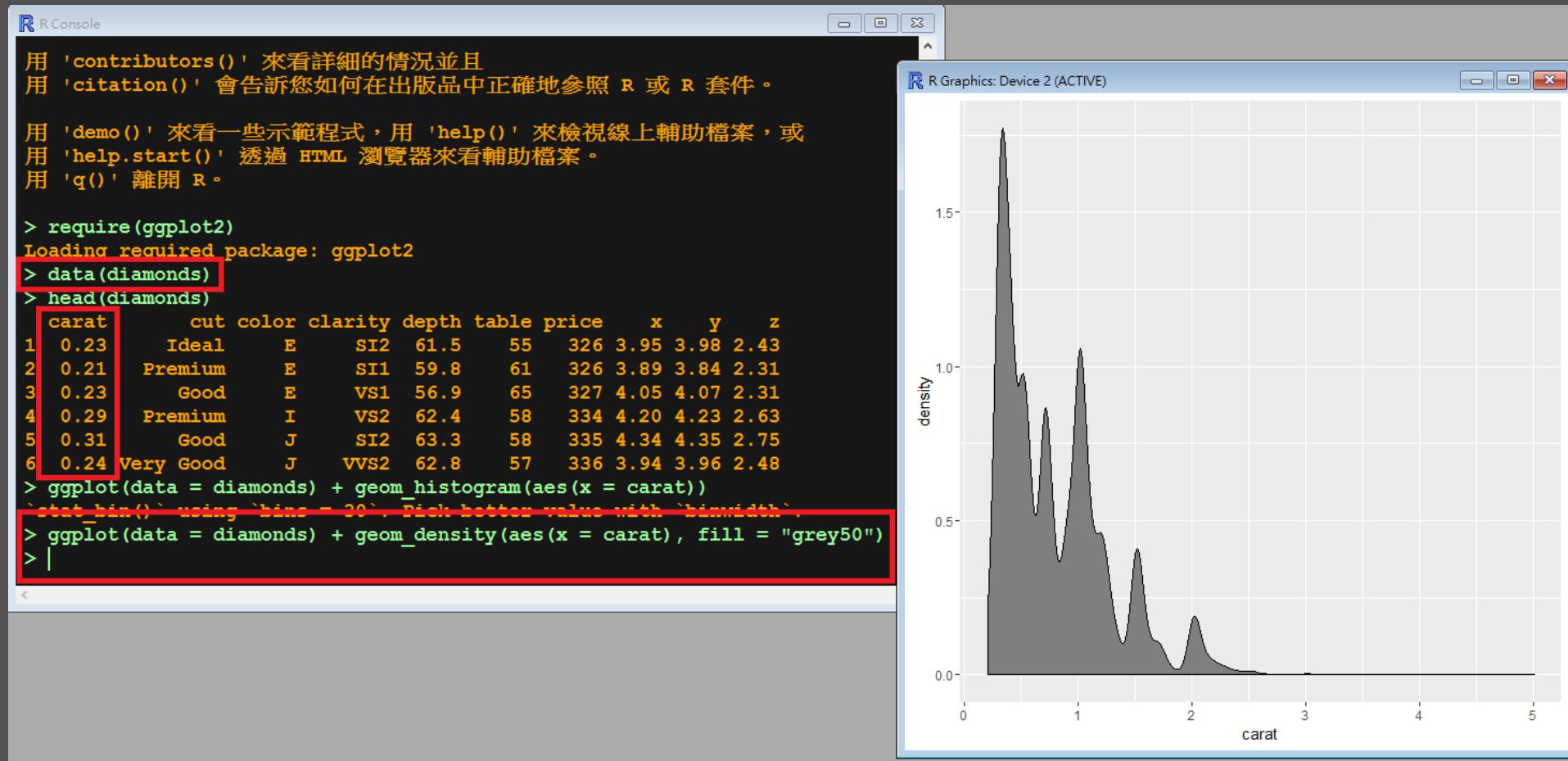


R ggplot2 機率密度圖

ggplot() -> 設定資料來源變數名稱 -> ggplot(data = [資料來源變數名稱])

geom_density() -> 設定為連續機率密度圖 並包含 aes() 函數、fill 引數。aes() 將其 x 軸 設為 carat 克拉， fill = " " 引數 為設定其色彩。

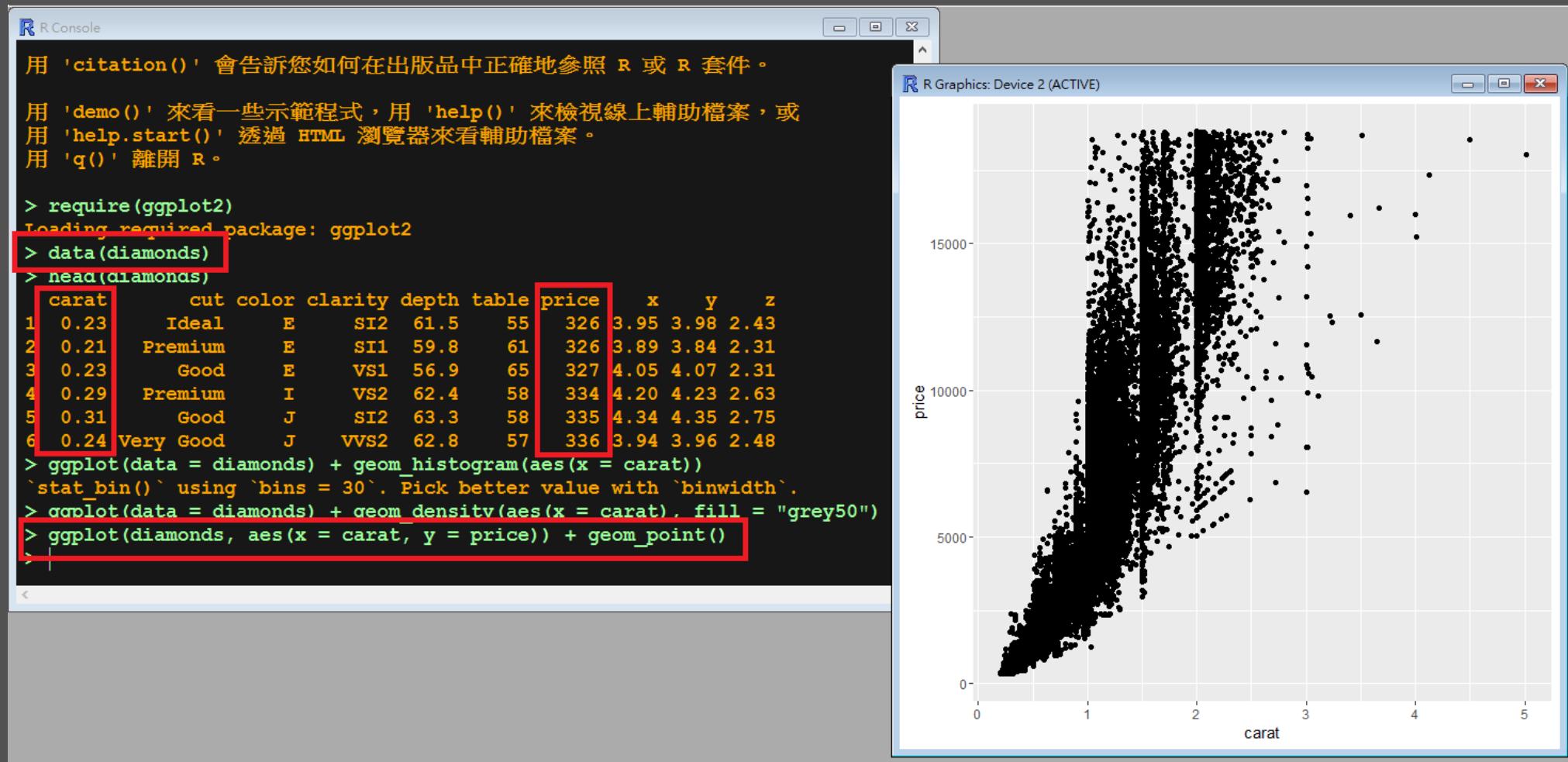
```
ggplot(data = diamonds) + geom_density(aes(x = carat), fill = "grey50")
```



R ggplot2 基本散佈圖

ggplot() -> 設定資料來源變數名稱 -> ggplot(data = [資料來源變數名稱])
將其 aes() 函數 兩軸 X 軸、Y 軸 各自設定 兩個資料行，分別為 克拉 與 價格，geom_point() 設其為散佈圖。

```
ggplot(diamonds, aes(x = carat, y = price)) + geom_point()
```



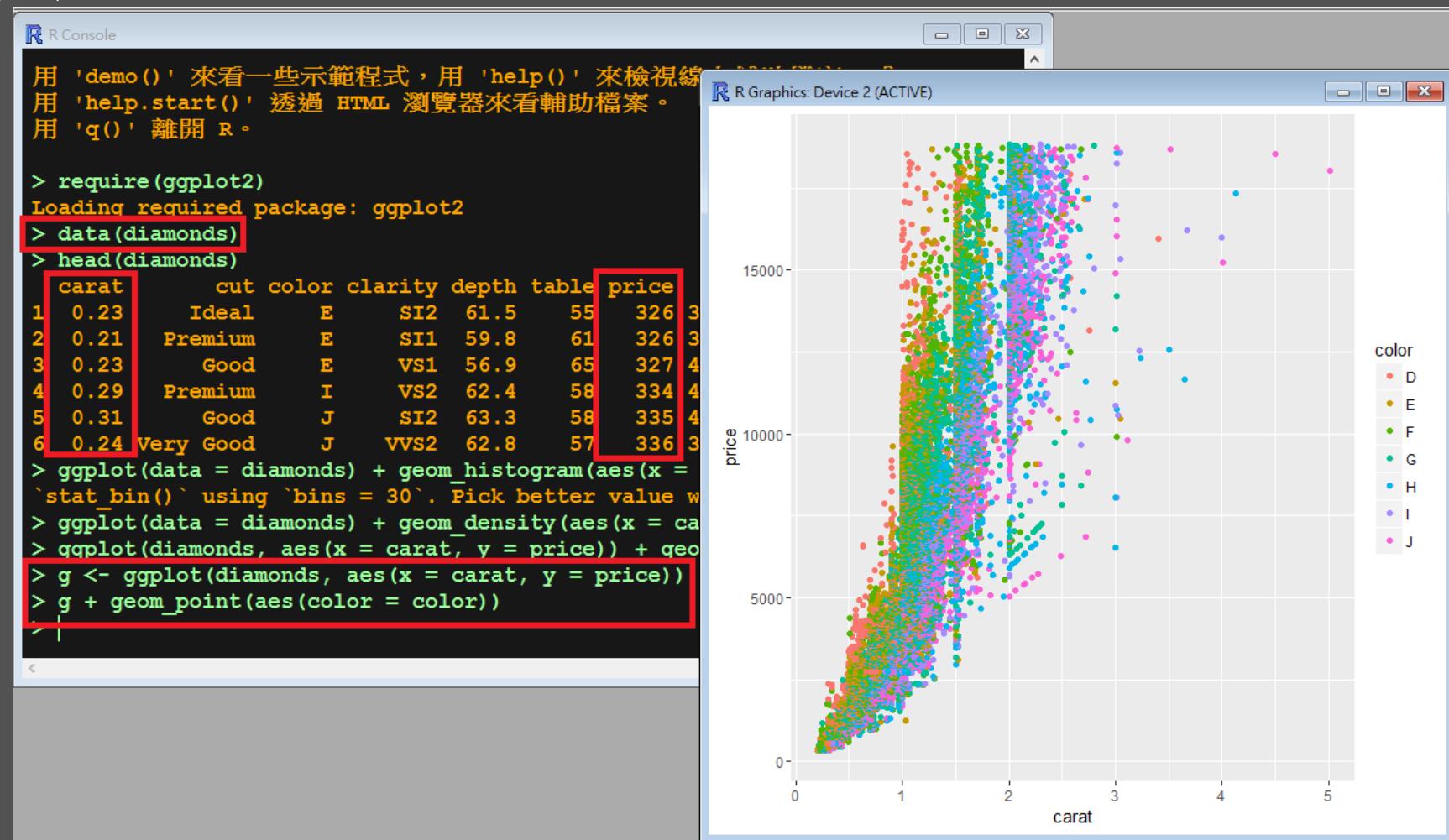
R ggplot2 基本散佈圖

因為後續要使用重複的程式，在此 ggplot 初始物件 令為 變數 g，以利重複使用 !!!

```
g <- ggplot(diamonds, aes(x = carat, y = price))
```

改變色系

```
g + geom_point(aes(color = color))
```



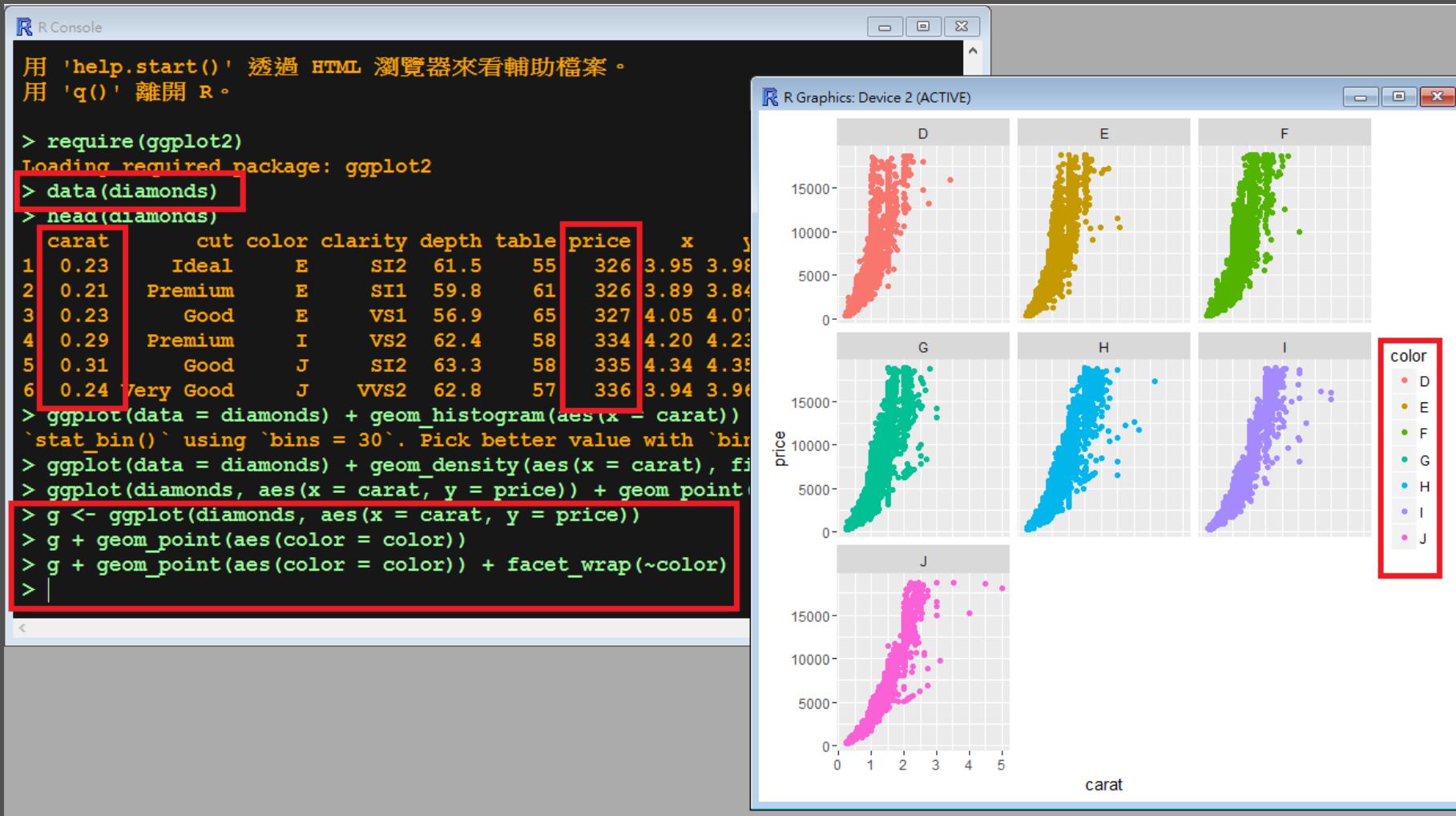
R ggplot2 基本散佈圖

要用多散佈圖 - 顏色分層，在此會應用到兩個函數。

1. facet_grid()
2. facet_wrap()

facet_wrap() 而基本用顏色分層，是將一群資料做好分群，並各自繪圖而後排列。

`g + geom_point(aes(color = color)) + facet_wrap(~color)`



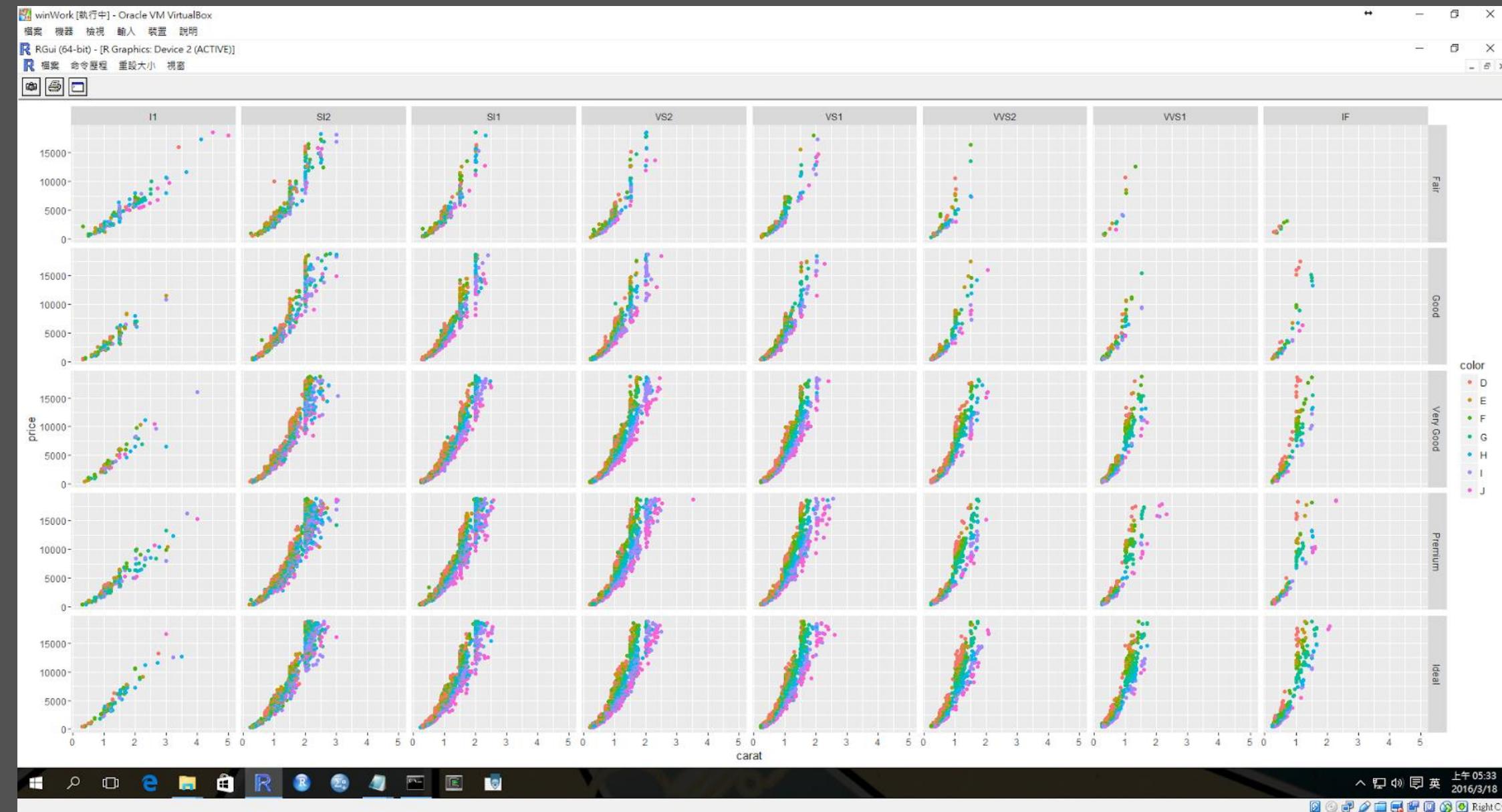
R ggplot2 基本散佈圖

多散佈圖 - 四軸顏色分層，使用的就是剛剛前面沒有說明的函數 `facet_grid()`，當中又插入兩個資料行分別為鑽石切割 (cut) 和 鑽石淨度 (clarity)

```
g + geom_point(aes(color = color)) + facet_grid(cut ~ clarity)
```

```
> data(diamonds)
> head(diamonds)
#> #> #> #> #> #>
#> #> #> #> #> #>
#> carat      cut   color clarity depth table price    x     y     z
#> 1 0.23     Ideal  D     SI2   61.5  55   326 3.95 3.98 2.43
#> 2 0.21     Premium  D   SII1   59.8  61   326 3.89 3.84 2.31
#> 3 0.23     Good   D   VS1    56.9  65   327 4.05 4.07 2.31
#> 4 0.29     Premium  D   VS2    62.4  58   334 4.20 4.23 2.63
#> 5 0.31     Good   J     SI2   63.3  58   335 4.34 4.35 2.75
#> 6 0.24 Very Good  J   VVS2   62.8  57   336 3.94 3.96 2.48
> ggplot(data = diamonds) + geom_histogram(aes(x = carat))
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
> ggplot(data = diamonds) + geom_density(aes(x = carat), fill = "grey50")
> ggplot(diamonds, aes(x = carat, v = price)) + geom_point()
> g <- ggplot(diamonds, aes(x = carat, y = price))
> g + geom_point(aes(color = color))
> g + geom_point(aes(color = color)) + facet_wrap(~color)
> g + geom_point(aes(color = color)) + facet_grid(cut ~ clarity)
```

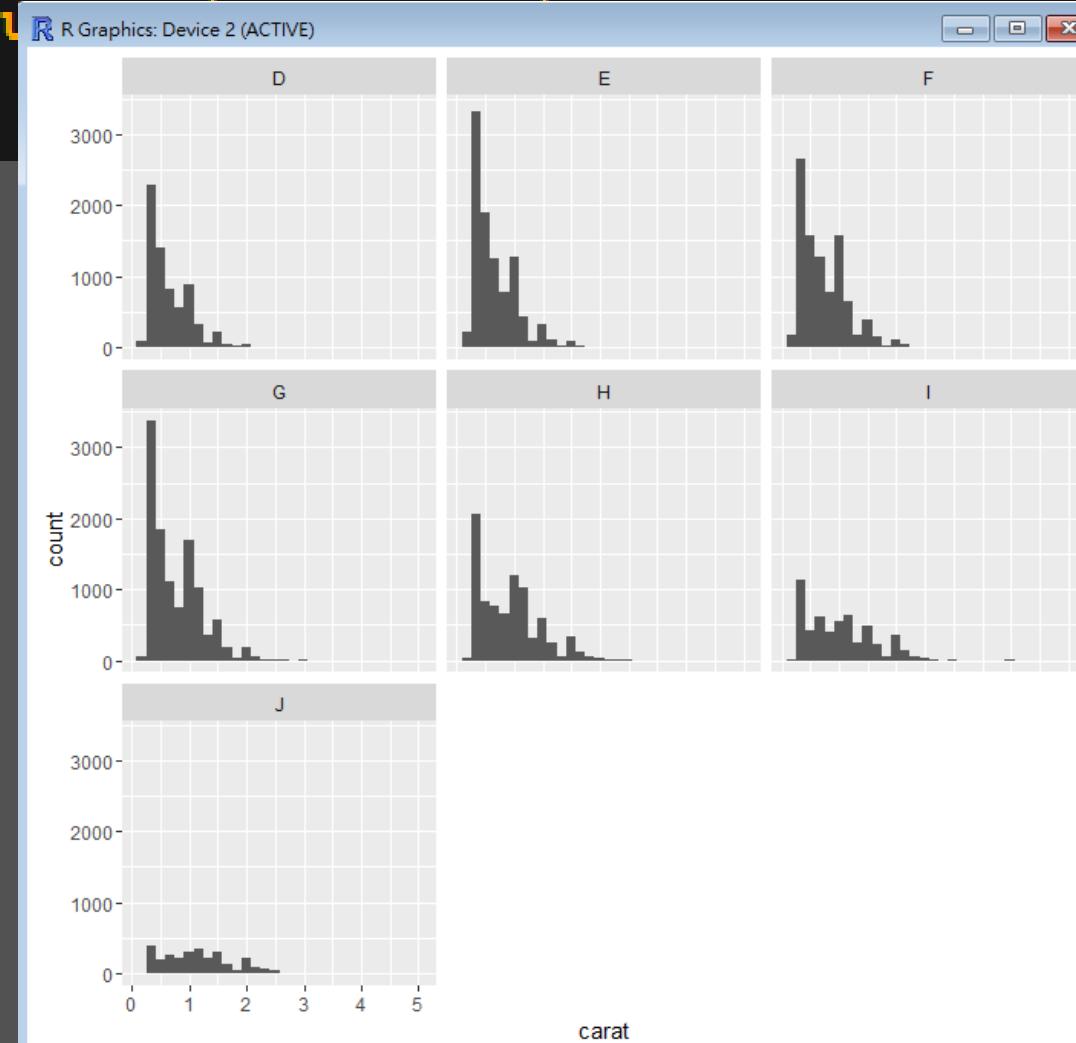
R ggplot2 基本散佈圖



R ggplot2 基本散佈圖

```
ggplot(diamonds, aes(x = carat)) + geom_histogram() +  
  facet_wrap(~color)
```

```
> ggplot(diamonds, aes(x = carat)) + geom_histogram() + facet_wrap(~color)  
`stat_bin()` using `bins = 30`. Consider specifying the number of bins  
with `binwidth`.
```



R ggplot2 箱型圖 與 小提琴圖

兩個 geom 的函數

1. geom_boxplot() -> 箱型圖

2. geom_violin() -> 小提琴圖

根據前面的玩法類推，我們可以選擇設定"單變數單維度" 亦或者是 "兩變數兩維度"

箱型圖

1. 單變數單維度

```
ggplot(diamonds, aes(y = carat, x = 1)) + geom_boxplot()
```

2. 兩變數兩維度

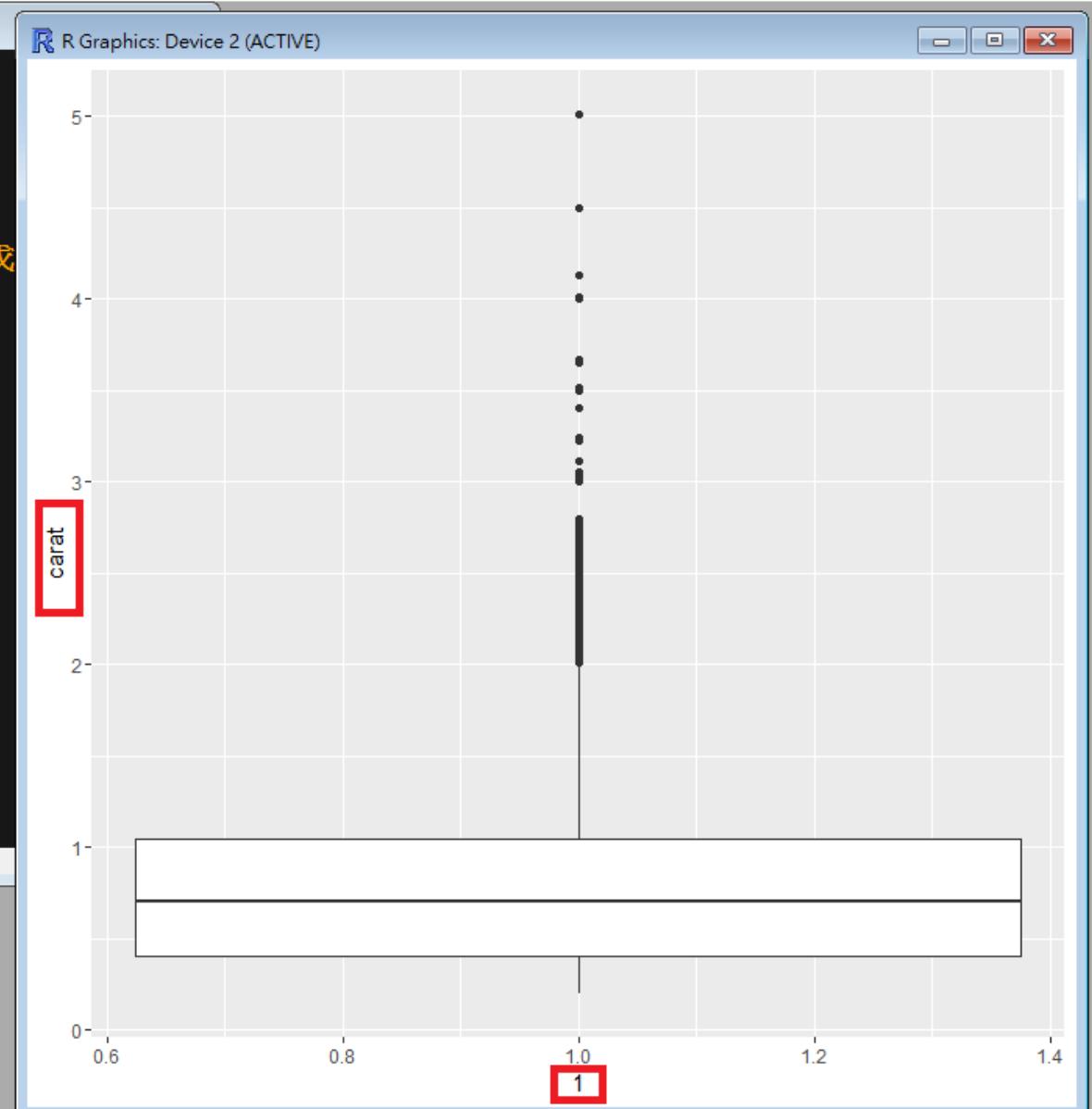
```
ggplot(diamonds, aes(y = carat, x = cut)) + geom_boxplot()
```

R ggplot2 箱型圖 與 小提琴圖

R 是個合作計劃，有許多人為之做出了貢獻。
 用 'contributors()' 來看詳細的情況並且
 用 'citation()' 會告訴您如何在出版品中正確地參照 R 或 R 套件。

用 'demo()' 來看一些示範程式，用 'help()' 來檢視線上輔助檔案，或
 用 'help.start()' 透過 HTML 瀏覽器來看輔助檔案。
 用 'q()' 離開 R。

```
> require(ggplot2)
Loading required package: ggplot2
> data(diamonds)
> head(diamonds)
   carat      cut color clarity depth table price     x     y     z
1 0.23    Ideal    E    SI2   61.5     55    326 3.95 3.98 2.43
2 0.21 Premium    E    SI1   59.8     61    326 3.89 3.84 2.31
3 0.23    Good    E    VS1   56.9     65    327 4.05 4.07 2.31
4 0.29 Premium    I    VS2   62.4     58    334 4.20 4.23 2.63
5 0.31    Good    J    SI2   63.3     58    335 4.34 4.35 2.75
6 0.24 Very Good   J   VVS2   62.8     57    336 3.94 3.96 2.48
> ggplot(diamonds, aes(y = carat, x = 1)) + geom_boxplot()
> |
```



R ggplot2 箱型圖 與 小提琴圖

R 是個合作計劃，有許多人為之做出了貢獻。
 用 'contributors()' 來看詳細的情況並且
 用 'citation()' 會告訴您如何在出版品中正確地參照 R 或 R 套件。

用 'demo()' 來看一些示範程式，用 'help()' 來檢視線上輔助檔案，或
 用 'help.start()' 透過 HTML 瀏覽器來看輔助檔案。
 用 'q()' 離開 R。

```
> require(ggplot2)
Loading required package: ggplot2
> data(diamonds)
> head(diamonds)
#> carat      cut color clarity depth table price     x     y     z
#> 1 0.23     Ideal E     SI2   61.5    55   326 3.95 3.98 2.43
#> 2 0.21     Premium E    SI1   59.8    61   326 3.89 3.84 2.31
#> 3 0.23      Good E    VS1   56.9    65   327 4.05 4.07 2.31
#> 4 0.29     Premium I    VS2   62.4    58   334 4.20 4.23 2.63
#> 5 0.31      Good J    SI2   63.3    58   335 4.34 4.35 2.75
#> 6 0.24 Very Good J   VVS2   62.8    57   336 3.94 3.96 2.48
> ggplot(diamonds, aes(y = carat, x = 1)) + geom_boxplot()
> ggplot(diamonds, aes(y = carat, x = cut)) + geom_boxplot()
>
```

The R Graphics window displays a boxplot with 'carat' on the y-axis and 'cut' on the x-axis. The 'cut' categories are Fair, Good, Very Good, Premium, and Ideal. The boxplot shows the distribution of diamond carat weight for each cut grade. The 'carat' label is highlighted with a red box in the R Graphics window.

R ggplot2 箱型圖 與 小提琴圖

小提琴圖

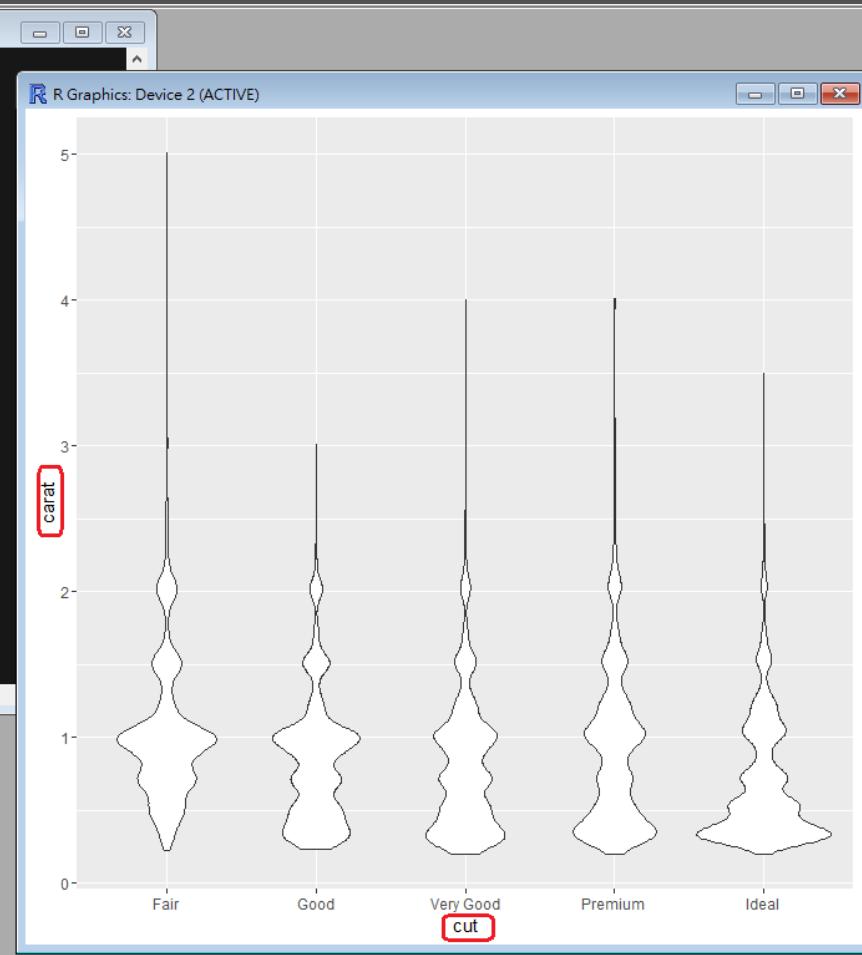
接下來說明 `geom_violin()` -> 小提琴圖函數，他跟箱型圖類似，把前面箱型圖的函數換成小提琴圖就可以。

```
ggplot(diamonds, aes(y = carat, x = cut)) +
  geom_violin()
```

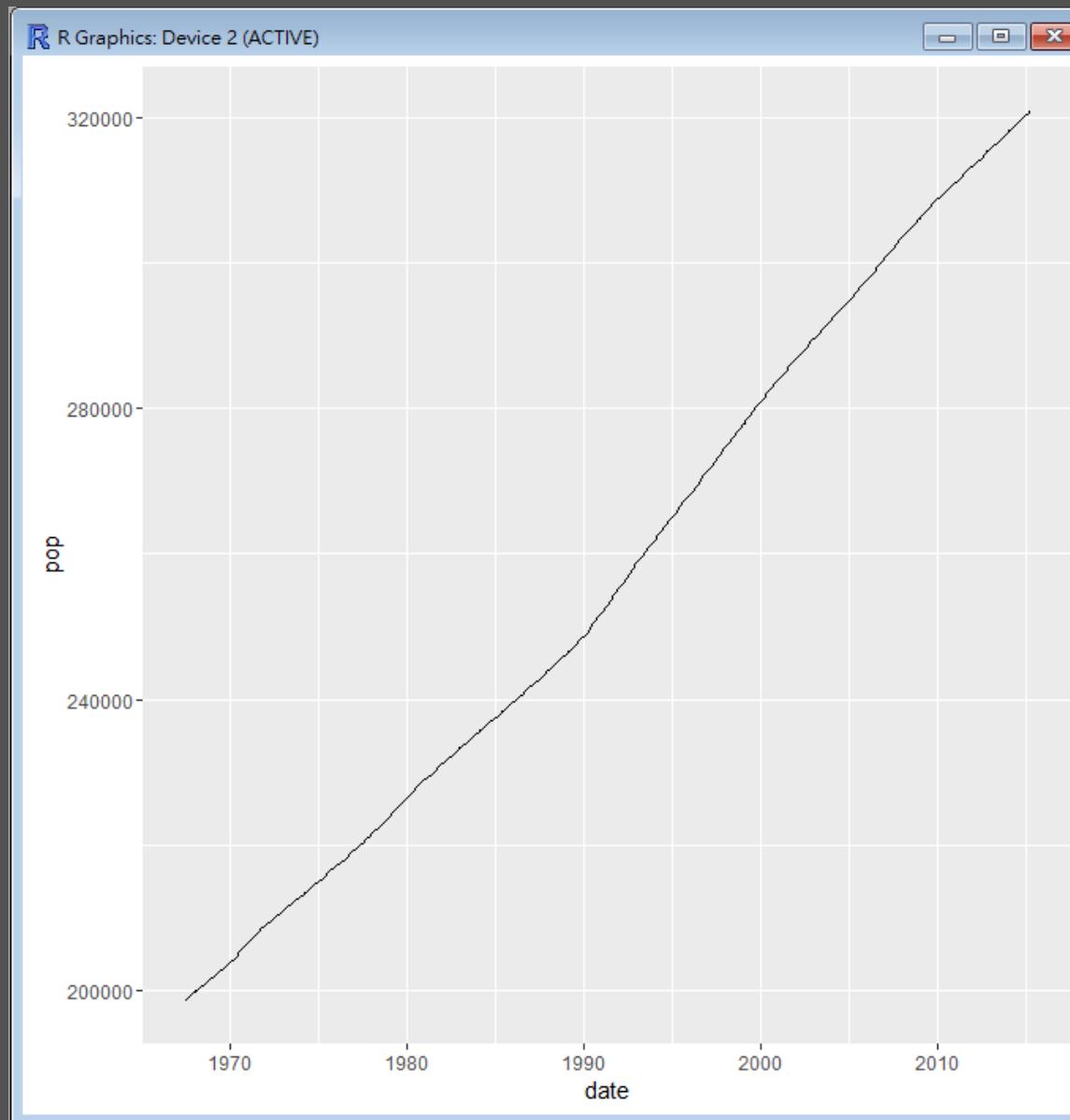
R Console

```
用 'contributors()' 來看詳細的情況並且
用 'citation()' 會告訴您如何在出版品中正確地參照 R 或 R 套件。
用 'demo()' 來看一些示範程式，用 'help()' 來檢視線上輔助檔案，或
用 'help.start()' 透過 HTML 瀏覽器來看輔助檔案。
用 'q()' 離開 R。

>
> require(ggplot2)
Loading required package: ggplot2
> data(diamonds)
> head(diamonds)
  carat       cut color clarity depth table price     x     y     z
1 0.23      Ideal   E    SI2   61.5     55    326 3.95 3.98 2.43
2 0.21    Premium   E    SI1   59.8     61    326 3.89 3.84 2.31
3 0.23      Good    E    VS1   56.9     65    327 4.05 4.07 2.31
4 0.29    Premium   I    VS2   62.4     58    334 4.20 4.23 2.63
5 0.31      Good    J    SI2   63.3     58    335 4.34 4.35 2.75
6 0.24  Very Good   J   VVS2   62.8     57    336 3.94 3.96 2.48
>
> ggplot(diamonds, aes(y = carat, x = cut)) + geom_violin()
> |
```



R ggplot2 線性圖



線性圖大多用於連續的變數，當然也可以用於離散的資料，其函數為 `geom_line()`，在此使用 ggplot2套件中的 economics - 經濟資料。

```
ggplot(economics, aes(x = date, y = pop)) +  
  geom_line()
```

R ggplot2 線性圖

接下來為了簡化，我們利用多個變數進行動作，使得日期從2000年開始，利用 lubridate 套件(R本身內涵的套件)來處理時間資料。

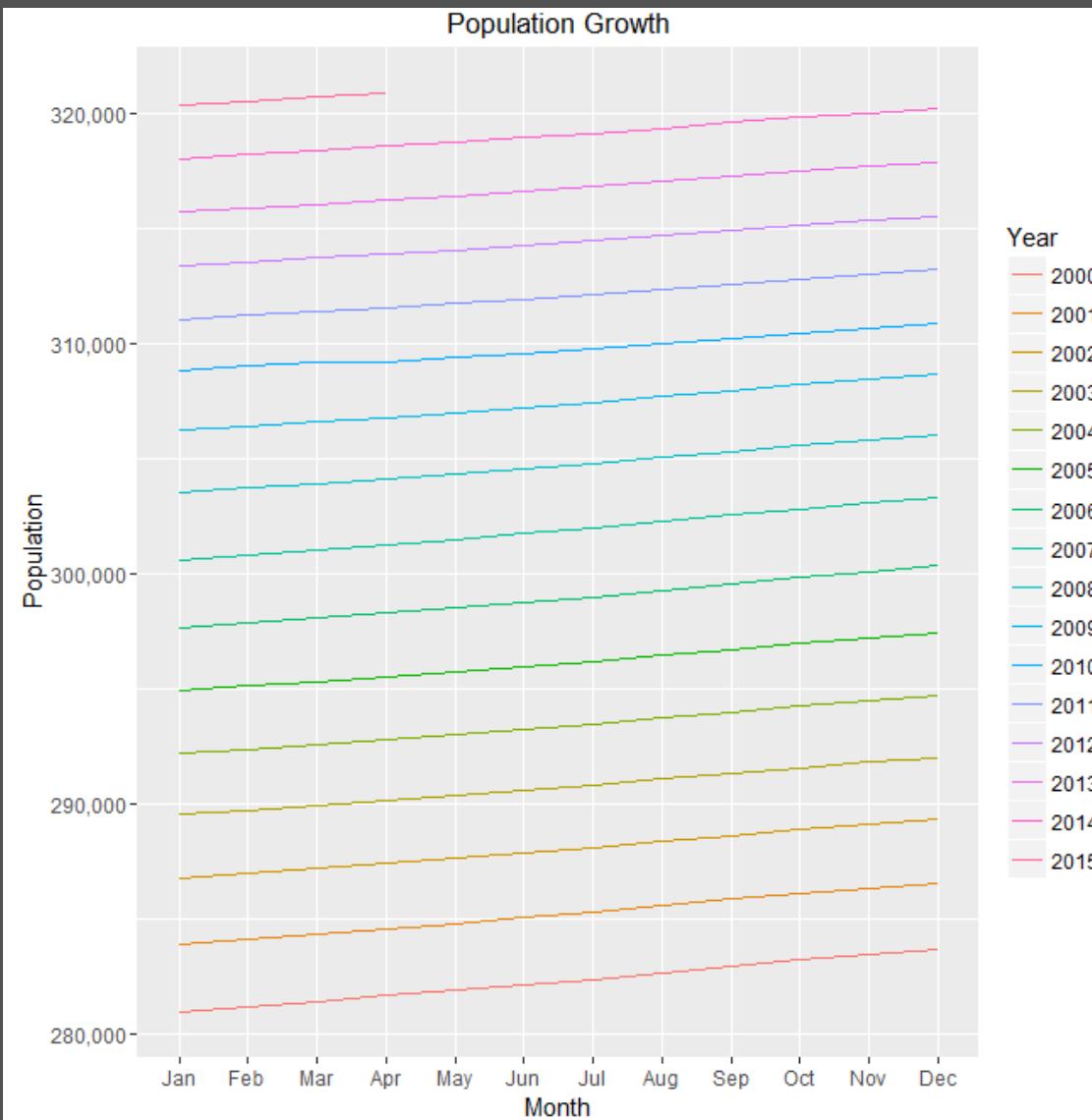
1. 日期 -> date(時間)
2. 年份 -> year(年份)
3. 月份 -> month(月)

用 lubridate 套件的原因是因為其時間直行資料欄之有一個名為 data ，當中將年、月、日，都塞在一起。匯入 R lubridate 套件，各自設定好年跟月的欄位!!! 最後一行則是一個判斷，將 2000 年以外的都幹掉，並存入一個變數。

```
require(lubridate)
economics$year <- year(economics$date)
economics$month <- month(economics$date, label=TRUE)
econ2000 <- economics[which(economics$year >= 2000), ]
```

	date	pce	pop	psavert	uempmed	unemploy
1	1967-07-01	507.4	198712	12.5	4.5	2944
2	1967-08-01	510.5	198911	12.5	4.7	2945
3	1967-09-01	516.3	199113	11.7	4.6	2958
4	1967-10-01	512.9	199311	12.5	4.9	3143
5	1967-11-01	518.1	199498	12.5	4.7	3066
6	1967-12-01	525.8	199657	12.1	4.8	3018

R ggplot2 線性圖



使用了 R 的名為 scales 的 內部套件，目的是為了處理圖中的 "軸"，接下來才是使用 ggplot2 的線性圖函數 -> geom_line(), 最後整理出來就能看到經濟上的月份與人口的成長。

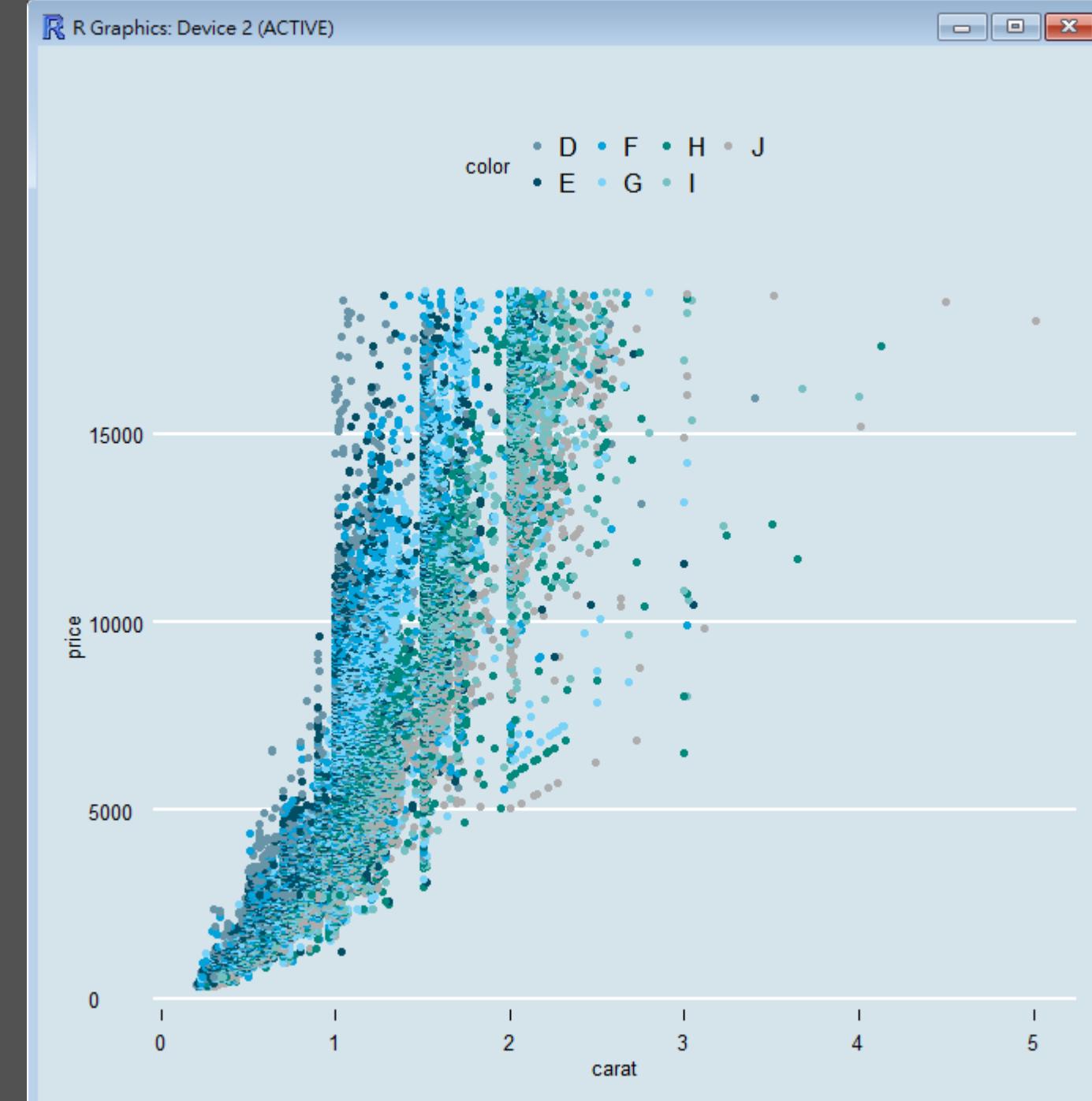
```
require(scales)
g <- ggplot(econ2000, aes(x=month, y=pop))
g <- g + geom_line(aes(color=factor(year), group=year))
g <- g + scale_color_discrete(name="Year")
g <- g + scale_y_continuous(labels=comma)
g <- g + labs(title="Population Growth", x="Month", y="Population")
g
```

R ggplot2 主題樣式

```
require(ggthemes)
g2 <- ggplot(diamonds, aes(x=carat, y=price)) + geom_point(aes(color=color))
g2 + theme_economist() + scale_colour_economist()
g2 + theme_excel() + scale_colour_excel()
g2 + theme_tufte()
g2 + theme_wsj()
```

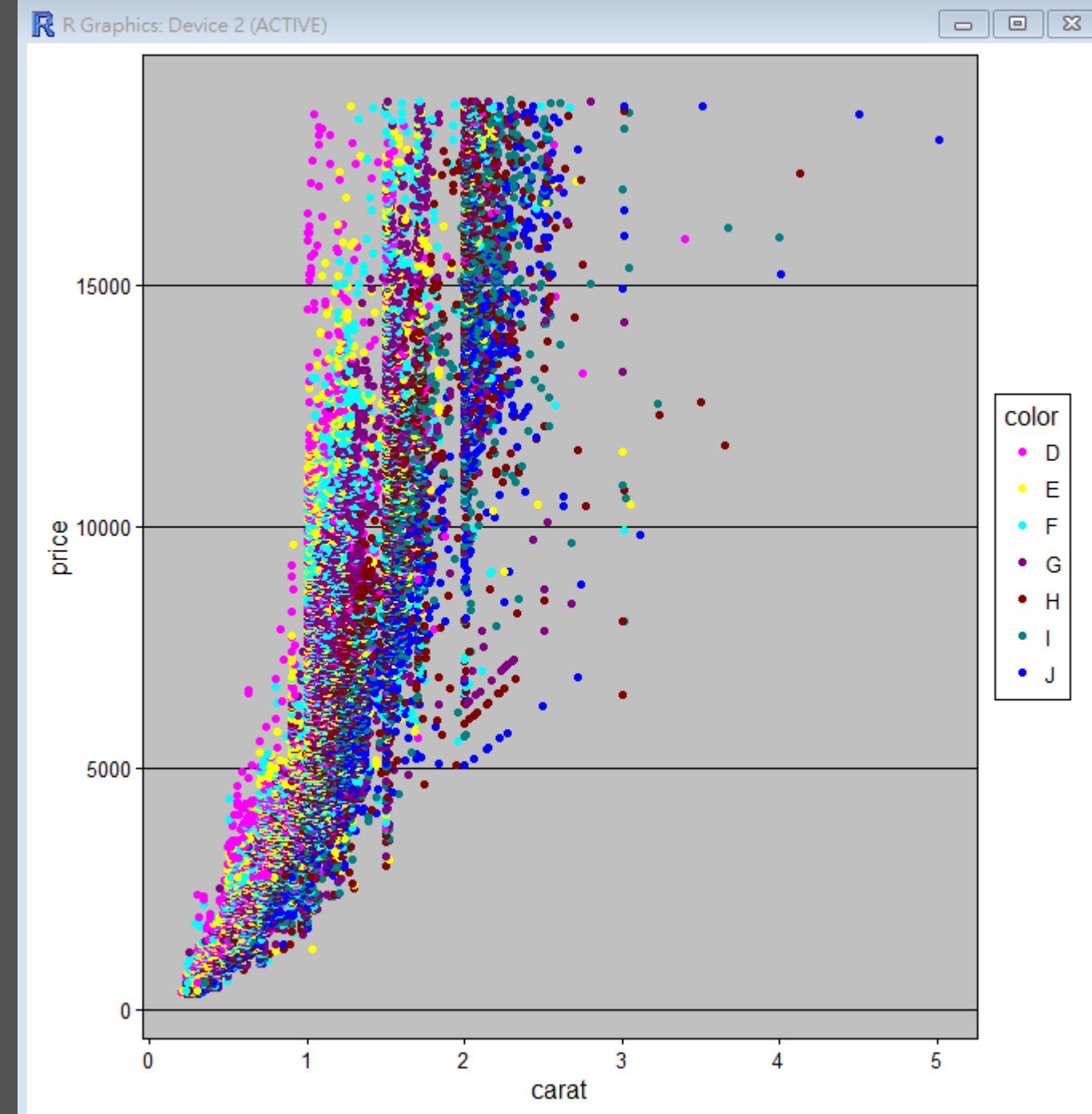
R ggplot2 主題樣式

```
g2 + theme_economist() + scale_colour_economist()
```



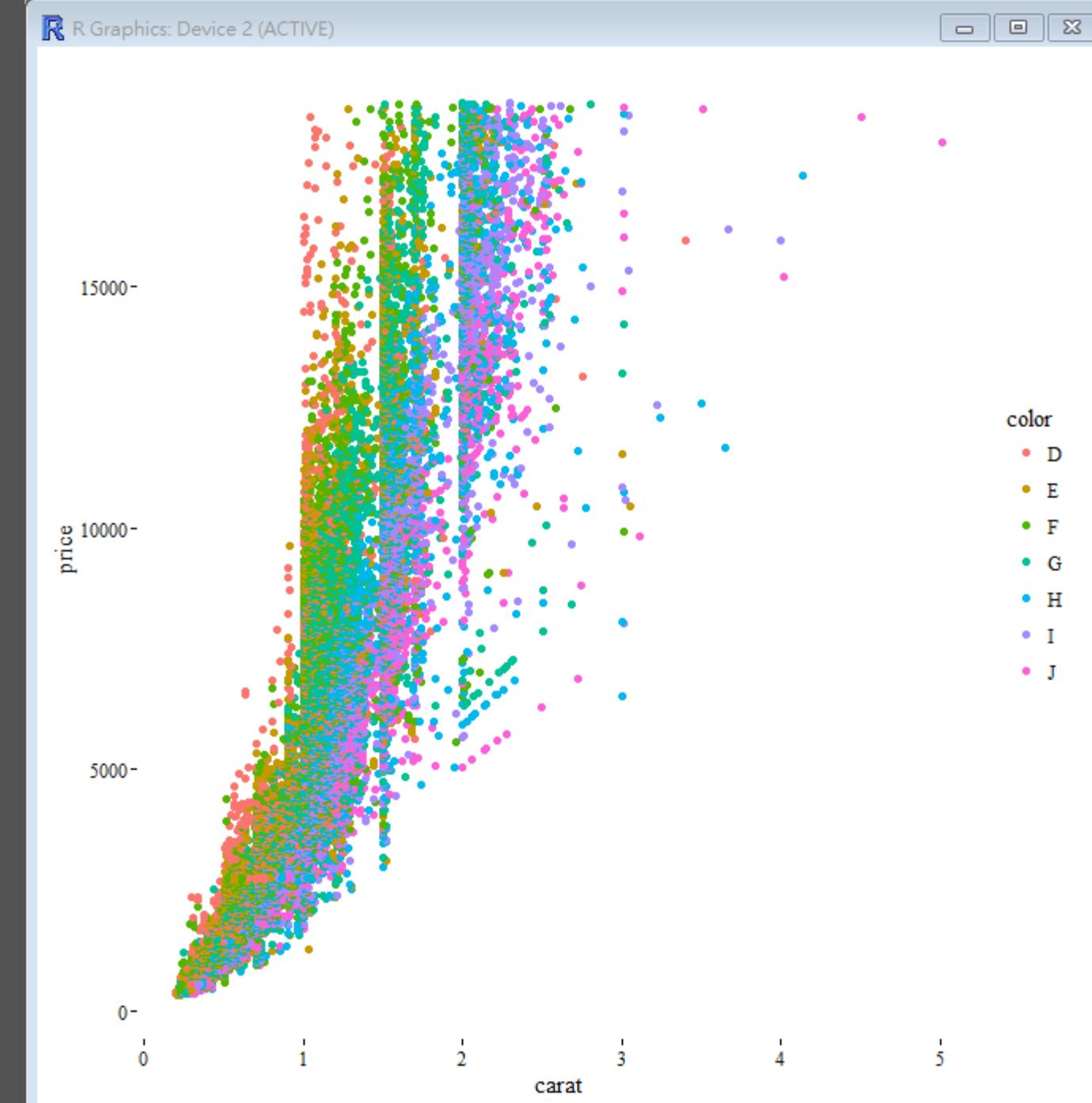
R ggplot2 主題樣式

g2 + theme_excel() + scale_colour_excel()



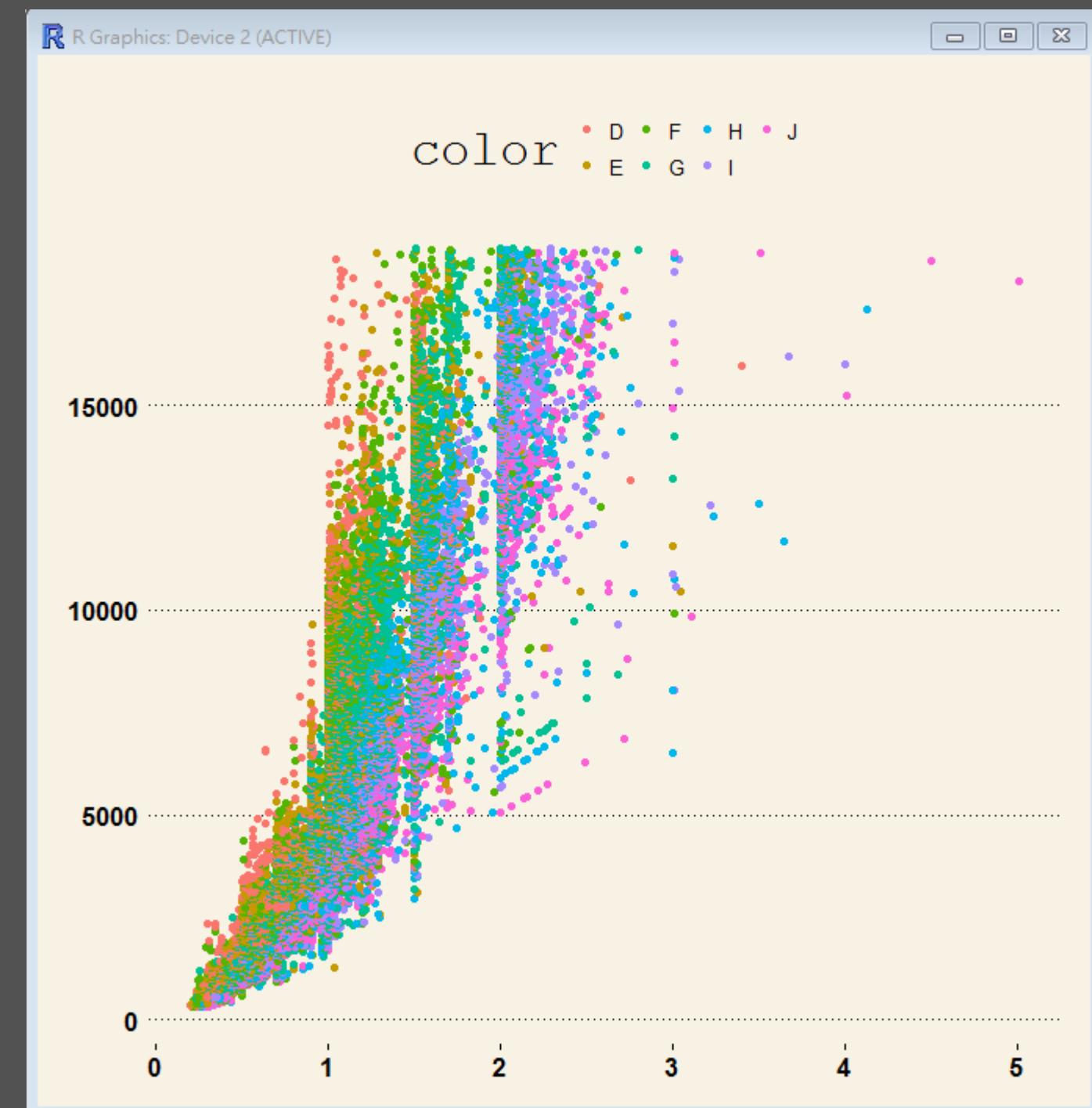
R ggplot2 主題樣式

g2 + theme_tufte()

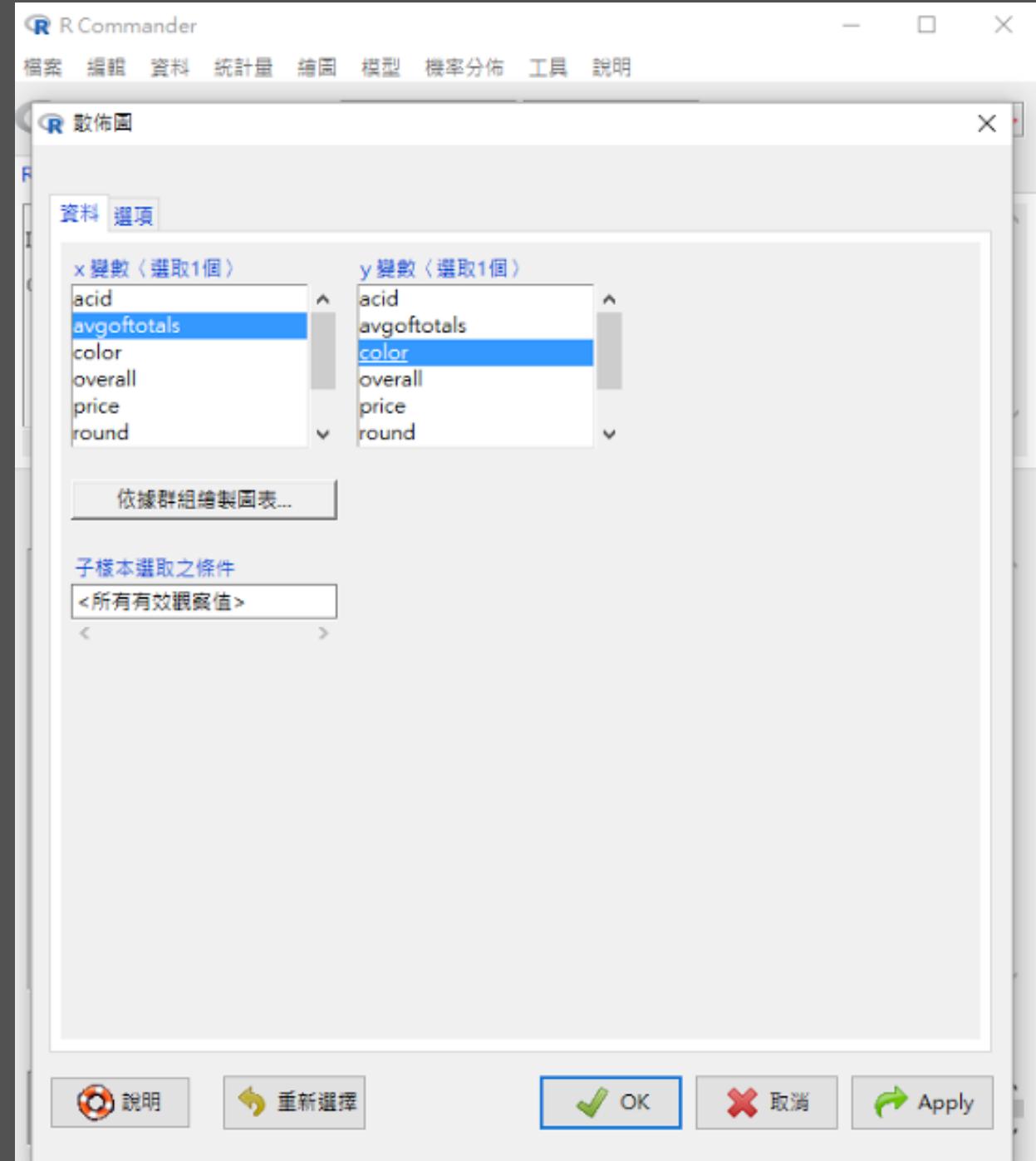
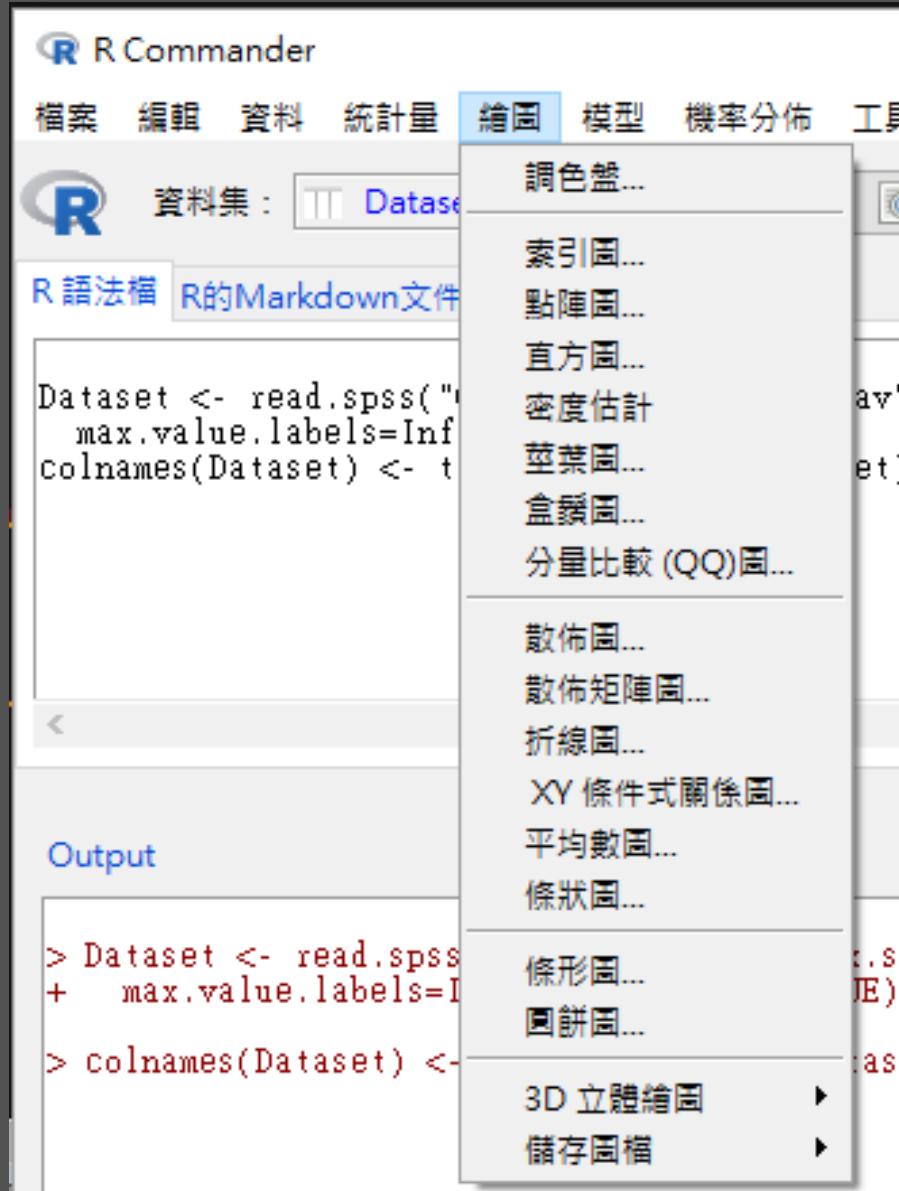


R ggplot2 主題樣式

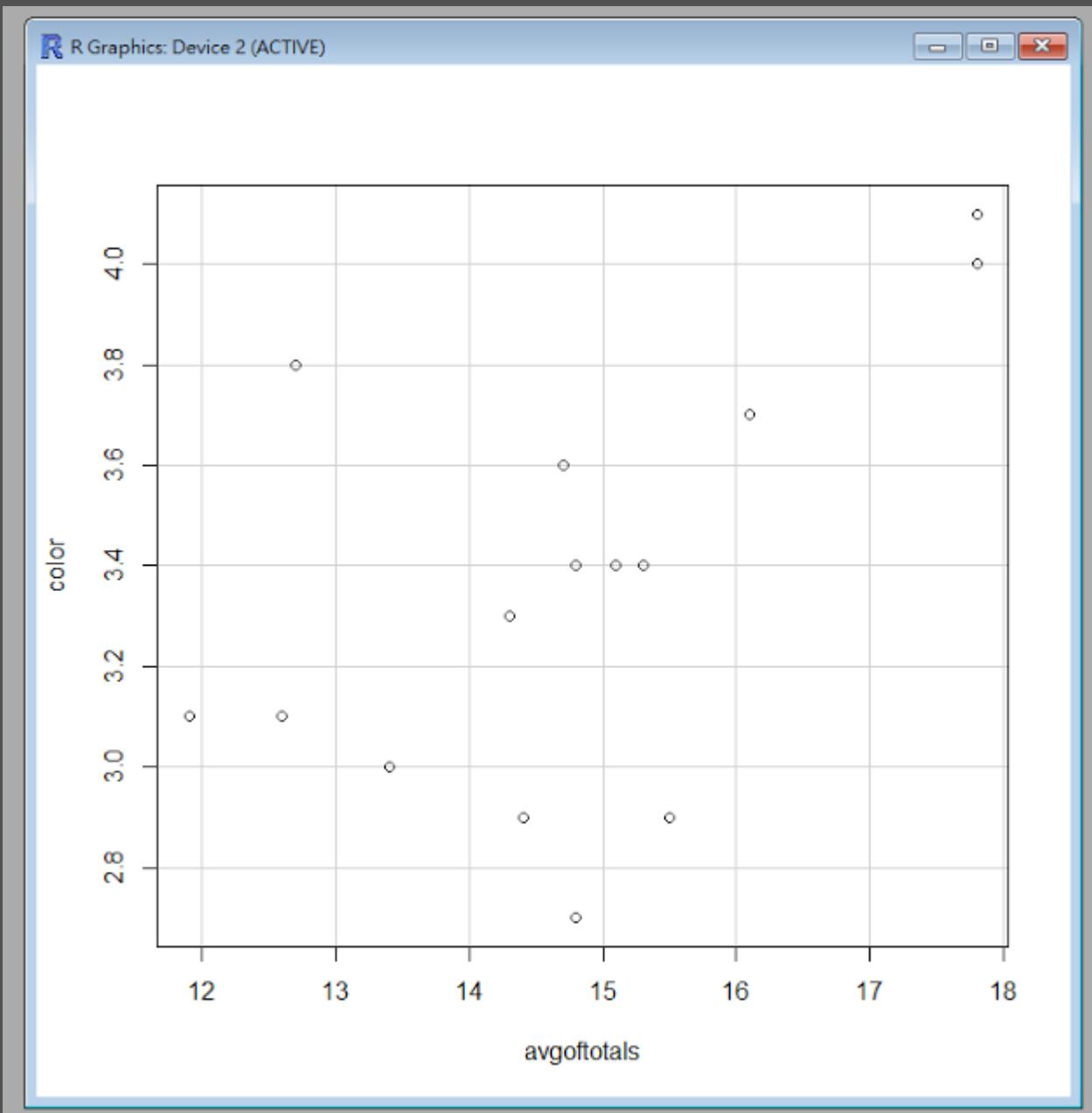
g2 + theme_wsj()



R RCommander 繪圖



R RCommander 繪圖



R ggplot 套件繪圖更換字體

參考書籍為 O'REILLY 的 R Graphics Cookbook

<http://www.cookbook-r.com/>

在此是用 extrafont 的套件，而範例則改用思源黑體來玩 !!!

```
# install.packages("extrafont")
library(extrafont)
```

```
> install.packages("extrafont")
Installing package into 'D:/USERDATA/Documents/R/win-library/3.3'
(as 'lib' is unspecified)
also installing the dependencies 'extrafontdb', 'Rttf2pt1'

嘗試 URL 'https://cloud.r-project.org/bin/windows/contrib/3.3/extrafontdb\_1.0.zip'
Content type 'application/zip' length 9975 bytes
downloaded 9975 bytes

嘗試 URL 'https://cloud.r-project.org/bin/windows/contrib/3.3/Rttf2pt1\_1.3.4.zip'
Content type 'application/zip' length 115960 bytes (113 KB)
downloaded 113 KB

嘗試 URL 'https://cloud.r-project.org/bin/windows/contrib/3.3/extrafont\_0.17.zip'
Content type 'application/zip' length 34323 bytes (33 KB)
downloaded 33 KB

package 'extrafontdb' successfully unpacked and MD5 sums checked
package 'Rttf2pt1' successfully unpacked and MD5 sums checked
package 'extrafont' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:\Users\USER\AppData\Local\Temp\RtmpYRODa6\downloaded_packages
> |
```

R ggplot 套件繪圖更換字體

這裡則是將電腦的字體備起來，這個過程會需要比較久的時間。
過程中會有該訊息請回 -> y

Importing fonts may take a few minutes, depending on the number of fonts and the speed of the system.
Continue? [y/n] y

font_import()

```
> font_import()
Importing fonts may take a few minutes, depending on the number of fonts and the speed of the system.
Continue? [y/n] y
Scanning ttf files in C:\WINDOWS\Fonts ...
Extracting .afm files from .ttf files...
C:\Windows\Fonts\21205_____.TTF => D:/USERDATA/Documents/R/win-library/3.3/extrafontdb/metrics/21205_____
C:\Windows\Fonts\2120A_____.TTF => D:/USERDATA/Documents/R/win-library/3.3/extrafontdb/metrics/2120A_____
C:\Windows\Fonts\35191_____.TTF => D:/USERDATA/Documents/R/win-library/3.3/extrafontdb/metrics/35191_____
C:\Windows\Fonts\40240_____.TTF => D:/USERDATA/Documents/R/win-library/3.3/extrafontdb/metrics/40240_____
C:\Windows\Fonts\46152_____.TTF => D:/USERDATA/Documents/R/win-library/3.3/extrafontdb/metrics/46152_____
C:\Windows\Fonts\50416_____.TTF => D:/USERDATA/Documents/R/win-library/3.3/extrafontdb/metrics/50416_____
```

R ggplot 套件繪圖更換字體

fonts()

```
> fonts()
[1] "Academy Engraved LET"
[5] "AR BLANCA"
[9] "AR CHRISTY"
[13] "AR DESTINE"
[17] "Arial"
[21] "AR JULIAN"
[25] "Bell MT"
[29] "Blackadder ITC"
[33] "Bodoni MT Condensed"
[37] "Bookshelf Symbol 7"
[41] "Broadway BT"
[45] "Calibri Light"
[49] "Cambria"
[53] "Cataneo BT"
[57] "Century Schoolbook"
[1] "Agency FB"
[5] "AR BONNIE"
[9] "AR DARLING"
[13] "AR ESSENCE"
[17] "Arial Narrow"
[21] "Arvo"
[25] "Berlin Sans FB"
[29] "Blackletter686 BT"
[33] "Bodoni MT Poster Compressed"
[37] "Bradley Hand ITC"
[41] "Brush Script MT"
[45] "Californian FB"
[49] "Candara"
[53] "Centaur"
[57] "Chiller"
[1] "Algerian"
[5] "AR CARTER"
[9] "AR DECODE"
[13] "AR HERMANN"
[17] "Arial Rounded MT Bold"
[21] "Baskerville Old Face"
[25] "Berlin Sans FB Demi"
[29] "Bodoni MT"
[33] "Book Antiqua"
[37] "Britannic Bold"
[41] "Caladea"
[45] "Calisto MT"
[49] "Carlito"
[53] "Century"
[57] "Colonna MT"
[1] "AR BERKLEY"
[5] "AR CENA"
[9] "AR DELANEY"
[13] "Arial Black"
[17] "Arial Unicode MS"
[21] "Bauhaus 93"
[25] "Bernard MT Condensed"
[29] "Bodoni MT Black"
[33] "Bookman Old Style"
[37] "Broadway"
[41] "Calibri"
[45] "Calligraph421 BT"
[49] "Castellar"
[53] "Century Gothic"
[57] "Comic Sans MS"
```

思源黑體

```
[173] "MV Boli"
[177] "Niagara Solid"
[181] "Noto Kufi Arabic"
[185] "Noto Sans Armenian"
[189] "Noto Sans Batak"
[193] "Noto Sans Buhid"
[197] "Noto Sans Cherokee"
[201] "Noto Sans Deseret"
[205] "Noto Sans Georgian"
[209] "Noto Sans Gurmukhi"
[213] "Noto Sans Inscriptional Pahlavi"
[217] "Noto Sans Kannada"
[1] "Myanmar Text"
[5] "Nirmala UI"
[9] "Noto Naskh Arabic"
[13] "Noto Sans Avestan"
[17] "Noto Sans Bengali"
[21] "Noto Sans Canadian Aboriginal"
[25] "Noto Sans Coptic"
[29] "Noto Sans Devanagari"
[33] "Noto Sans Glagolitic"
[37] "Noto Sans Hanunoo"
[41] "Noto Sans Inscriptional Parthian"
[45] "Noto Sans Kayah Li"
[1] "New Gillim"
[5] "Nirmala UI Semilight"
[9] "Noto Nastaliq Urdu"
[13] "Noto Sans Balinese"
[17] "Noto Sans Brahmi"
[21] "Noto Sans Carian"
[25] "Noto Sans Cuneiform"
[29] "Noto Sans Egyptian Hieroglyphs"
[33] "Noto Sans Gothic"
[37] "Noto Sans Hebrew"
[41] "Noto Sans Javanese"
[45] "Noto Sans Kharoshthi"
[1] "Niagara Engraved"
[5] "Noto Emoji"
[9] "Noto Sans"
[13] "Noto Sans Bamum"
[17] "Noto Sans Buginese"
[21] "Noto Sans Cham"
[25] "Noto Sans Cypriot"
[29] "Noto Sans Ethiopic"
[33] "Noto Sans Gujarati"
[37] "Noto Sans Imperial Aramaic"
[41] "Noto Sans Kaithi"
[45] "Noto Sans Khmer"
```

R ggplot 套件繪圖更換字體

匯入套件與載入字體。

```
> library(extrafont)
Registering fonts with R
> loadfonts("win")
Academy Engraved LET already registered with windowsFonts().
Agency FB already registered with windowsFonts().
Algerian already registered with windowsFonts().
AR BERKLEY already registered with windowsFonts().
AR BLANCA already registered with windowsFonts().
AR BONNIE already registered with windowsFonts().
AR CARTER already registered with windowsFonts().
AR CENA already registered with windowsFonts().
```

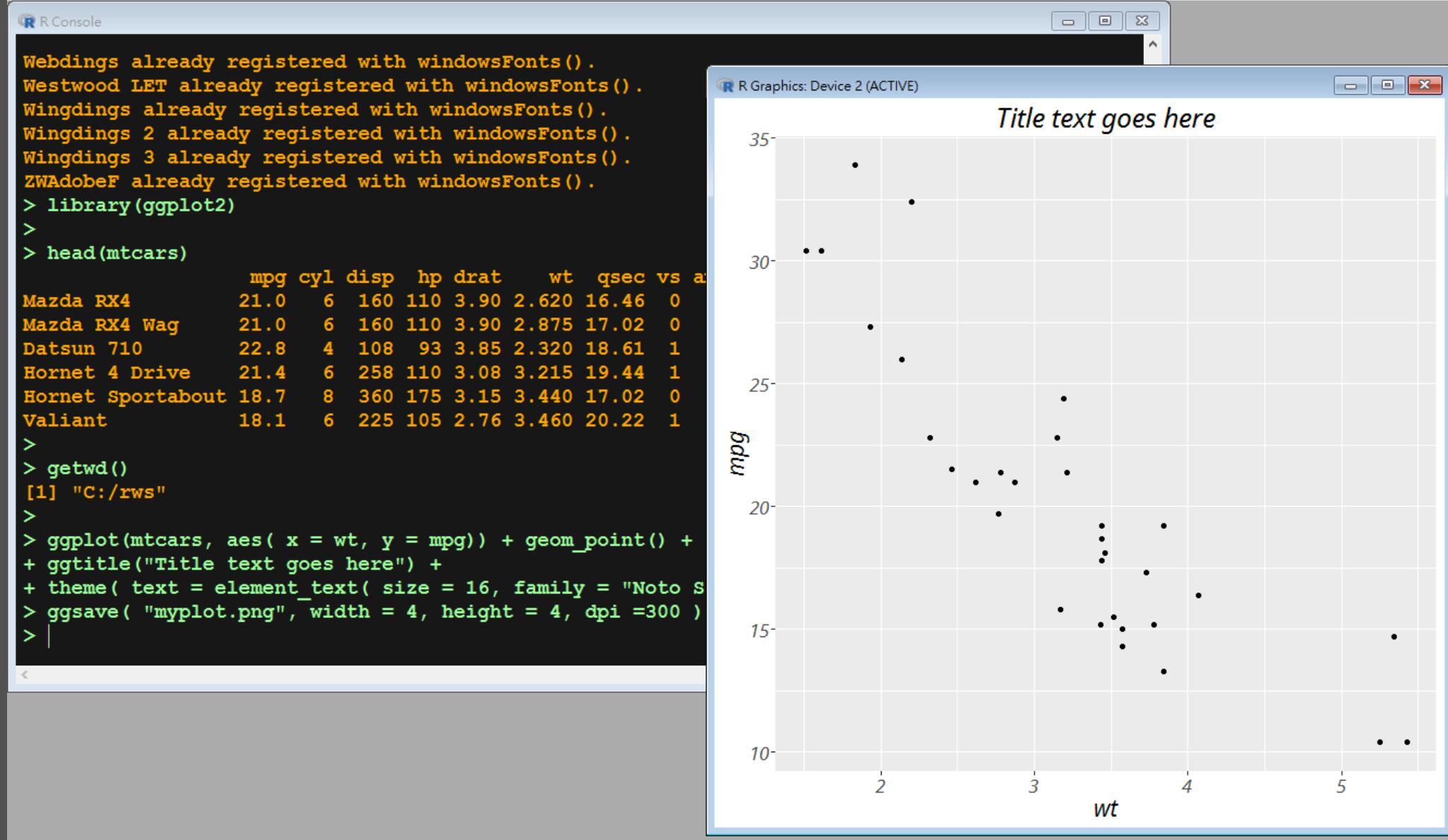
R ggplot 套件繪圖更換字體

匯入 ggplot2 的繪圖套件，在此我們使用內建的 mtcars 資料 進行繪圖。

```
> library(ggplot2)
>
> head(mtcars)
      mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4     21.0   6 160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag 21.0   6 160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710    22.8   4 108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive 21.4   6 258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8 360 175 3.15 3.440 17.02  0  0    3    2
Valiant       18.1   6 225 105 2.76 3.460 20.22  1  0    3    1
>
> getwd()
[1] "C:/rws"
>
> ggplot(mtcars, aes( x = wt, y = mpg)) + geom_point() +
+ ggttitle("Title text goes here") +
+ theme( text = element_text( size = 16, family = "Noto Sans", face = "italic"))
> ggsave( "myplot.png", width = 4, height = 4, dpi =300 )
> |
```

R ggplot 套件繪圖更換字體

結果

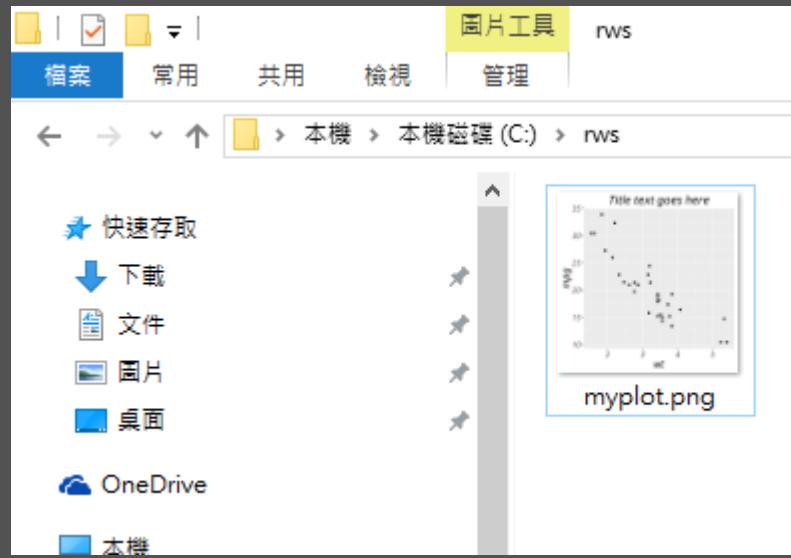


R ggplot 套件繪圖更換字體

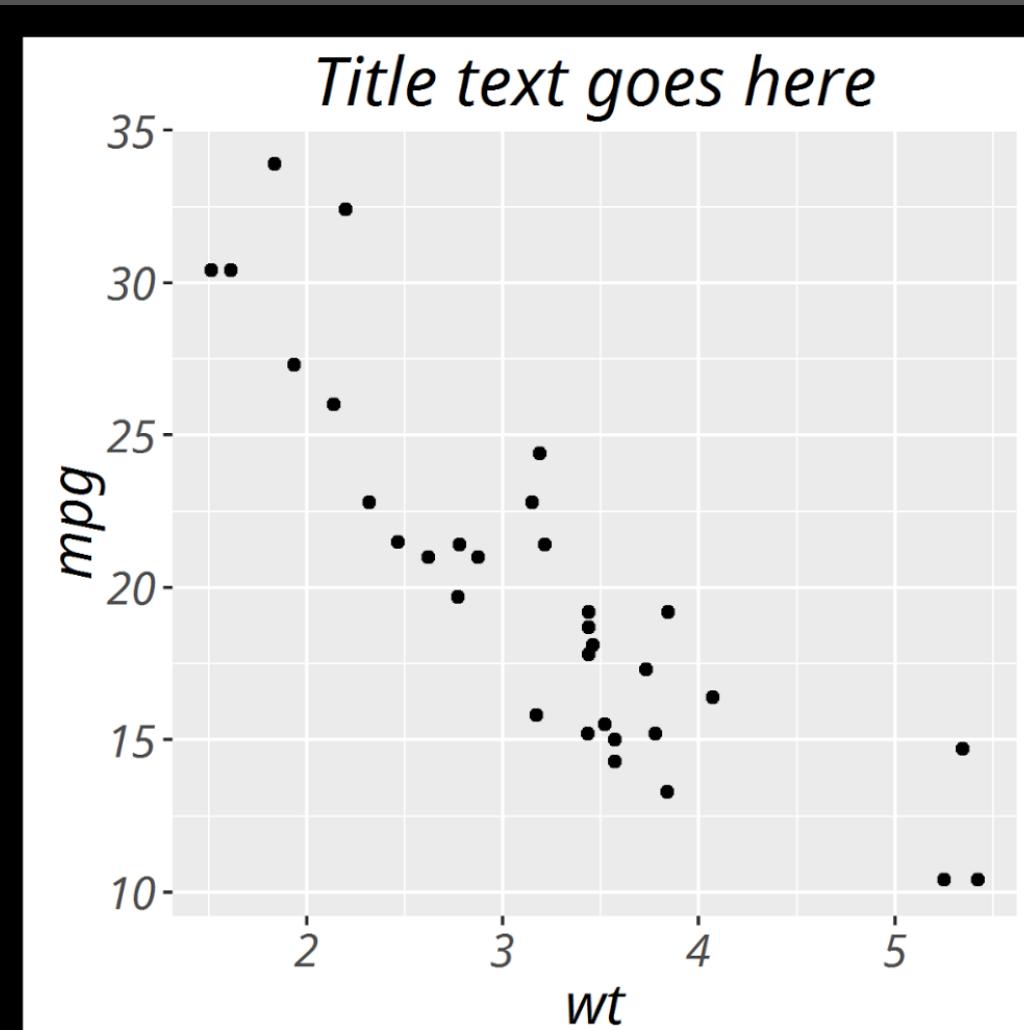
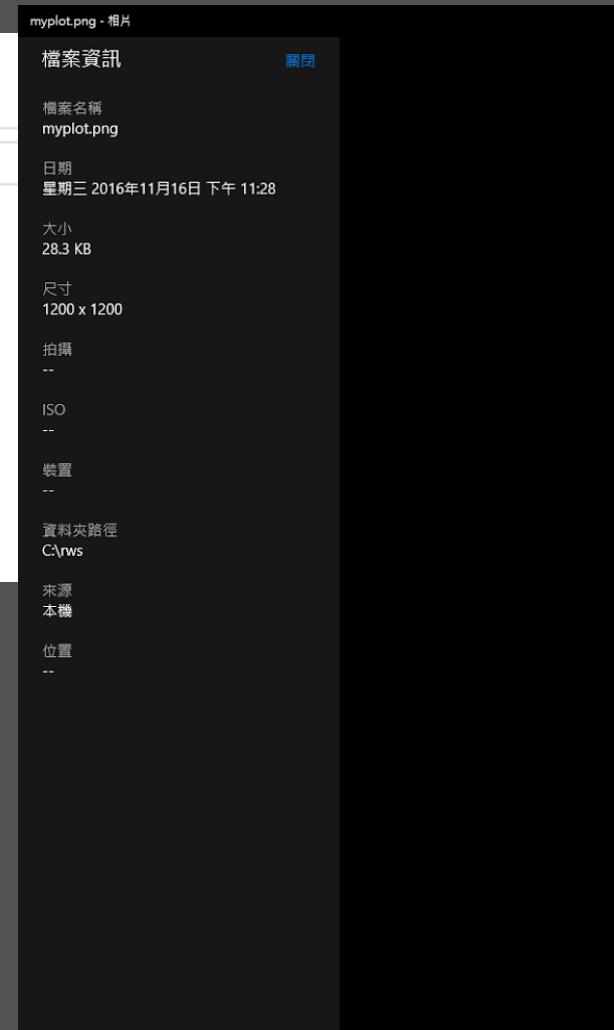
```
library(extrafont)
loadfonts("win")
library(ggplot2)
head(mtcars)
getwd()
ggplot(mtcars, aes( x = wt, y = mpg)) + geom_point() +
  ggtitle("Title text goes here") +
  theme( text = element_text( size = 16, family = "Noto Sans", face = "italic"))
ggsave( "myplot.png", width = 4, height = 4, dpi =300 )
```

R ggplot 套件繪圖更換字體

基本上輸出會是在預設的工作目錄上

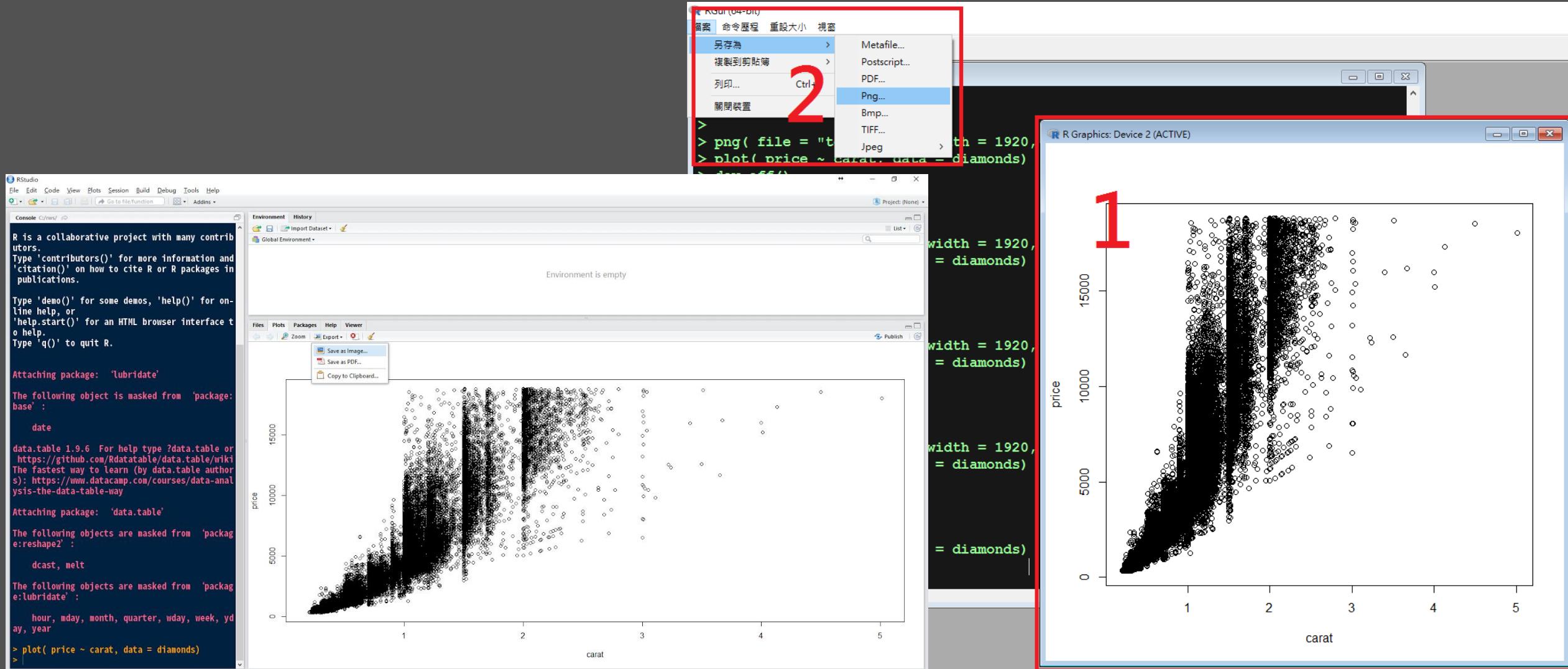


基本上輸出會是在預設的工作目錄上



R 圖像輸出

下圖為 R Console GUI 使用，而 Visual Studio R Tool 還可以直接受剪貼簿的方式，直接複製進 Office 用在 Word 等軟體內。



R 圖像輸出

1. 開啟輸出

規則上都是輸出的指定格式的函數 -> `png()`、`bmp()`、`jpeg()`、`tiff()`、`pdf()`.....等等，來開啟輸出。

2. 繪圖

這裡就是使用繪圖函數，此章則是用 R 內建的繪圖函數來玩，而 `ggplot` 的處理比較特別，要多一些動作，在此先不談。

3. 關閉輸出

`dev.off()`

基本上這幾個 `png()`、`bmp()`、`jpeg()`、`tiff()` 預設輸出多少類似，但在參數上還是有所不同，可以掌握的參數就是寬高與 `res`、`units`、`pointsize`。

各個圖形格式都有自己的特性，我們在此不詳述。當然目前最常用的圖形格式基本上都是 (*.png)，習慣上如果你的電腦作業系統，MS Windows 在 Win7 之後的版本、Linux、Mac 預設。

R 圖像輸出

```
 bmp(filename = "Rplot%03d.bmp",
      width = 480, height = 480, units = "px", pointsize = 12,
      bg = "white", res = NA, ...,
      type = c("cairo", "Xlib", "quartz"), antialias)

 jpeg(filename = "Rplot%03d.jpeg",
       width = 480, height = 480, units = "px", pointsize = 12,
       quality = 75,
       bg = "white", res = NA, ...,
       type = c("cairo", "Xlib", "quartz"), antialias)

 png(filename = "Rplot%03d.png",
      width = 480, height = 480, units = "px", pointsize = 12,
      bg = "white", res = NA, ...,
      type = c("cairo", "cairo-png", "Xlib", "quartz"), antialias)

 tiff(filename = "Rplot%03d.tiff",
       width = 480, height = 480, units = "px", pointsize = 12,
       compression = c("none", "rle", "lzw", "jpeg", "zip", "lzw+p", "zip+p"),
       bg = "white", res = NA, ...,
       type = c("cairo", "Xlib", "quartz"), antialias)
```

R 圖像輸出

```
win.metafile(filename = "", width = 7, height = 7, pointsize = 12,  
            family, restoreConsole = TRUE)  
  
pdf(file = if(onefile) "Rplots.pdf" else "Rplot%03d.pdf",  
     width, height, onefile, family, title, fonts, version,  
     paper, encoding, bg, fg, pointsize, pagecentre, colormodel,  
     useDingbats, useKerning, fillOddEven, compress)  
  
svg(filename = if(onefile) "Rplots.svg" else "Rplot%03d.svg",  
     width = 7, height = 7, pointsize = 12,  
     onefile = FALSE, family = "sans", bg = "white",  
     antialias = c("default", "none", "gray", "subpixel"))
```

R 圖像輸出(width & height)

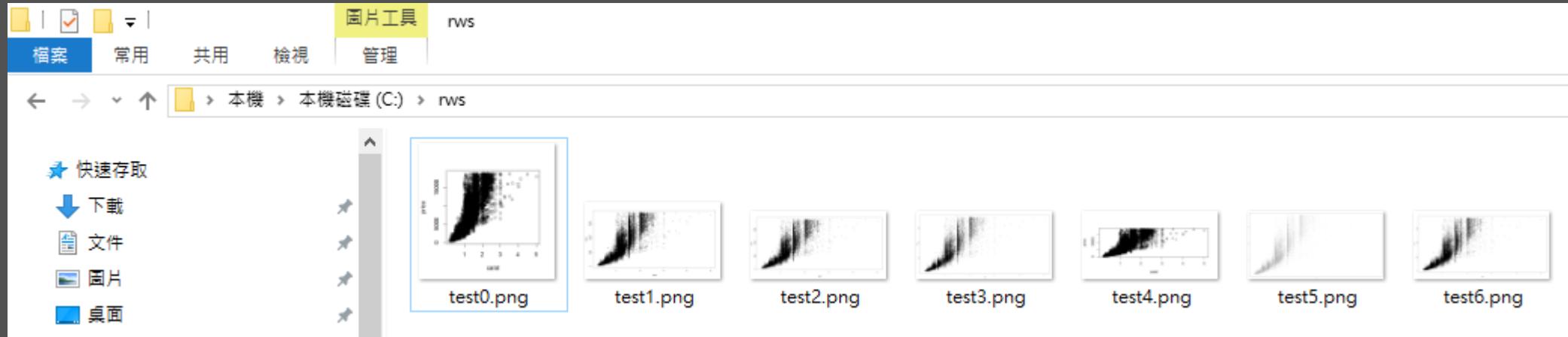
使用 ggplot2 套件下的 diamonds 資料。

```
library(ggplot2)
head(diamonds)
class(diamonds)
getwd()
```

```
> library(ggplot2)
> head(diamonds)
# A tibble: 6 × 10
  carat      cut color clarity depth table price     x     y     z
  <dbl>    <ord> <ord>   <ord> <dbl> <dbl> <int> <dbl> <dbl> <dbl>
1 0.23     Ideal     E     SI2  61.5    55    326  3.95  3.98  2.43
2 0.21     Premium   E     SI1  59.8    61    326  3.89  3.84  2.31
3 0.23     Good      E     VS1  56.9    65    327  4.05  4.07  2.31
4 0.29     Premium   I     VS2  62.4    58    334  4.20  4.23  2.63
5 0.31     Good      J     SI2  63.3    58    335  4.34  4.35  2.75
6 0.24 Very Good   J     VVS2  62.8    57    336  3.94  3.96  2.48
> class(diamonds)
[1] "tbl_df"     "tbl"        "data.frame"
> getwd()
[1] "C:/rws"
>
```

R 圖像輸出(width & height)

其圖檔匯出會在工作目錄。



R 圖像輸出 (width & height)

```
library(ggplot2)
head(diamonds)
class(diamonds)
getwd()
```

```
png( file = "test0.png", res = 120)
plot( price ~ carat, data = diamonds)
dev.off()
```

```
png( file = "test1.png", width = 1500, height = 850, res = 120)
plot( price ~ carat, data = diamonds)
dev.off()
```

```
png( file = "test2.png", width = 1920, height = 962, res =120)
plot( price ~ carat, data = diamonds)
dev.off()
```

```
png( file = "test3.png", width = 1920, height = 962)
plot( price ~ carat, data = diamonds)
dev.off()
```

```
png( file = "test4.png", width = 1920, height = 962, res = 300)
plot( price ~ carat, data = diamonds)
dev.off()
```

```
png( file = "test5.png", width = 1920, height = 962, res = 30)
plot( price ~ carat, data = diamonds)
dev.off()
```

```
png( file = "test6.png", width = 1920, height = 962, res = 90)
plot( price ~ carat, data = diamonds)
dev.off()
```

R 圖像輸出(7 種繪圖格式 & ggplot)

先匯入 ggplot2 套件，並利用當中的 diamonds 資料集與 R 本
身就有的 mtcars 資料，最後檢查工作目錄的路徑。

```
library(ggplot2)
head(mtcars)
head(diamonds)
getwd()
```

```
> library(ggplot2)
> head(mtcars)

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

> head(diamonds)
# A tibble: 6 × 10
  carat      cut color clarity depth table price     x     y     z
  <dbl>    <ord> <ord>    <ord> <dbl> <dbl> <int> <dbl> <dbl> <dbl>
1 0.23     Ideal     E     SI2    61.5    55    326  3.95  3.98  2.43
2 0.21     Premium   E     SI1    59.8    61    326  3.89  3.84  2.31
3 0.23     Good      E     VS1    56.9    65    327  4.05  4.07  2.31
4 0.29     Premium   I     VS2    62.4    58    334  4.20  4.23  2.63
5 0.31     Good      J     SI2    63.3    58    335  4.34  4.35  2.75
6 0.24 Very Good   J     VVS2   62.8    57    336  3.94  3.96  2.48

> getwd()
[1] "C:/rws"
> |
```

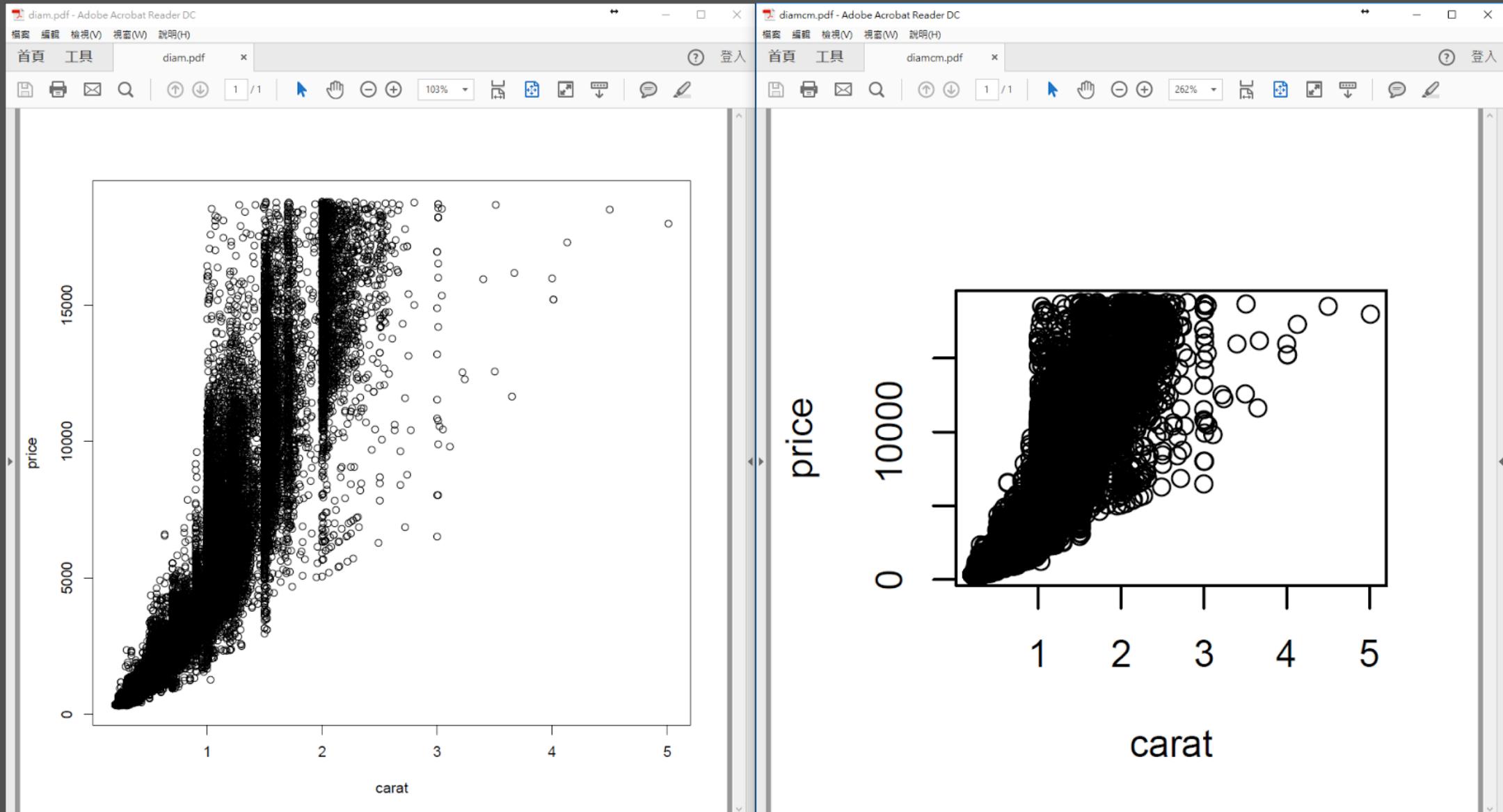
R 圖像輸出(7 種繪圖格式 & ggplot)

PDF 要注意預設的長寬單位

```
> # PDF 輸出，寬高皆為英寸
> pdf( file = "diam.pdf", width = 8, height = 8)
> plot( price ~ carat, data = diamonds)
> dev.off()
null device
      1
>
> # PDF 輸出，調整成公分
> pdf( file = "diamcm.pdf", width = 8/2.54, height = 8/2.54)
> plot( price ~ carat, data = diamonds)
> dev.off()
null device
      1
> |
```

R 圖像輸出(7 種繪圖格式 & ggplot)

PDF



R 圖像輸出(7 種繪圖格式 & ggplot)

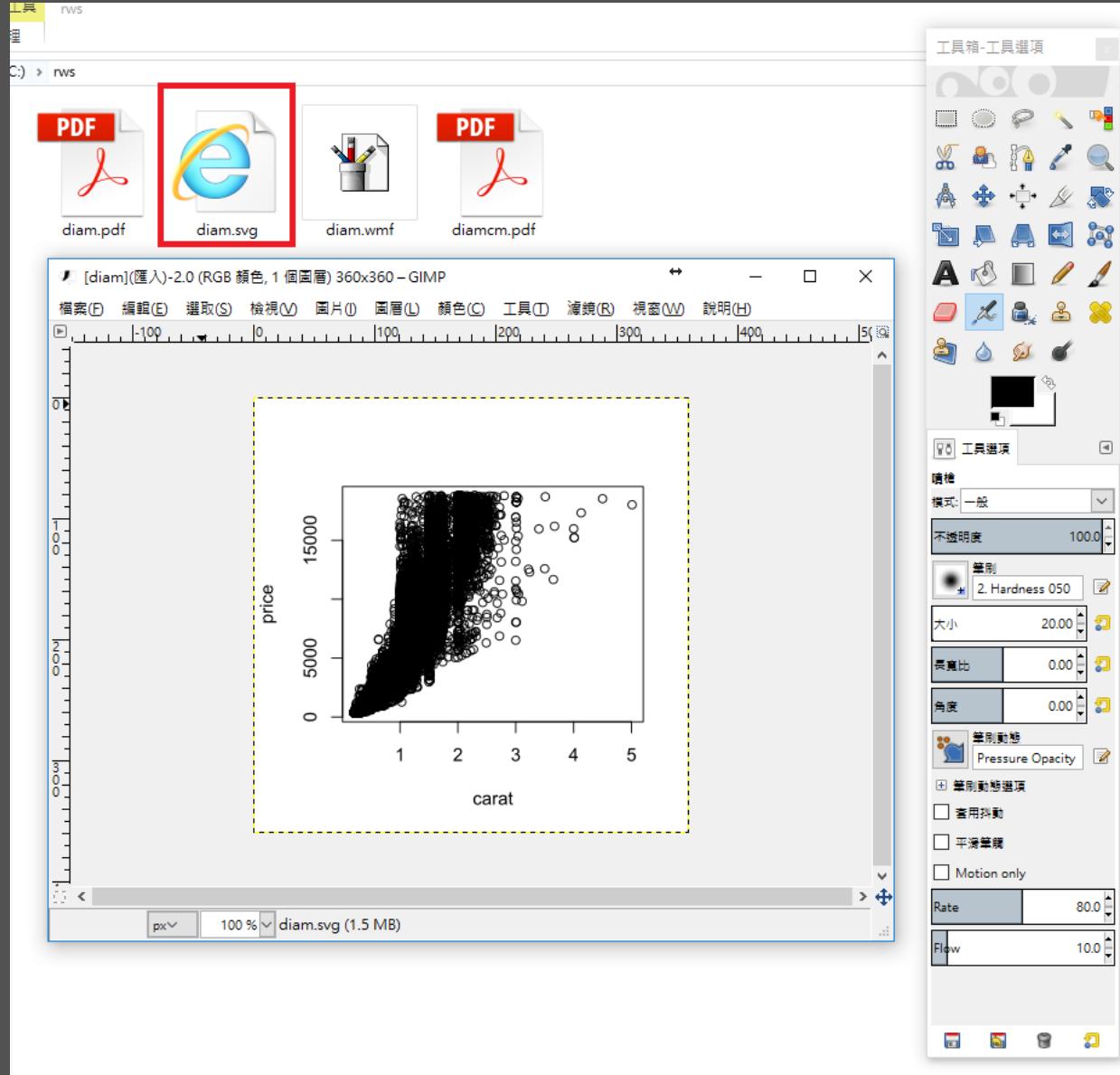
SVG 與 WMF

匯出的 SVG 可用 GIMP 來看，用 Adobe illustrator 開起來可能會跑掉。而 WMF 為 Windows 才能使用，畢竟是古早 MS 專屬的格式，Linux 與 Mac 也能使用。

```
> # SVG 輸出
> svg( file = "diam.svg", width = 4, height = 4)
> plot( price ~ carat, data = diamonds)
> dev.off()
null device
      1
>
> # WMF 輸出，僅能在Windows 上使用
> win.metafile( file = "diam.wmf", width = 4, height = 4)
> plot( price ~ carat, data = diamonds)
> dev.off()
null device
      1
> |
```

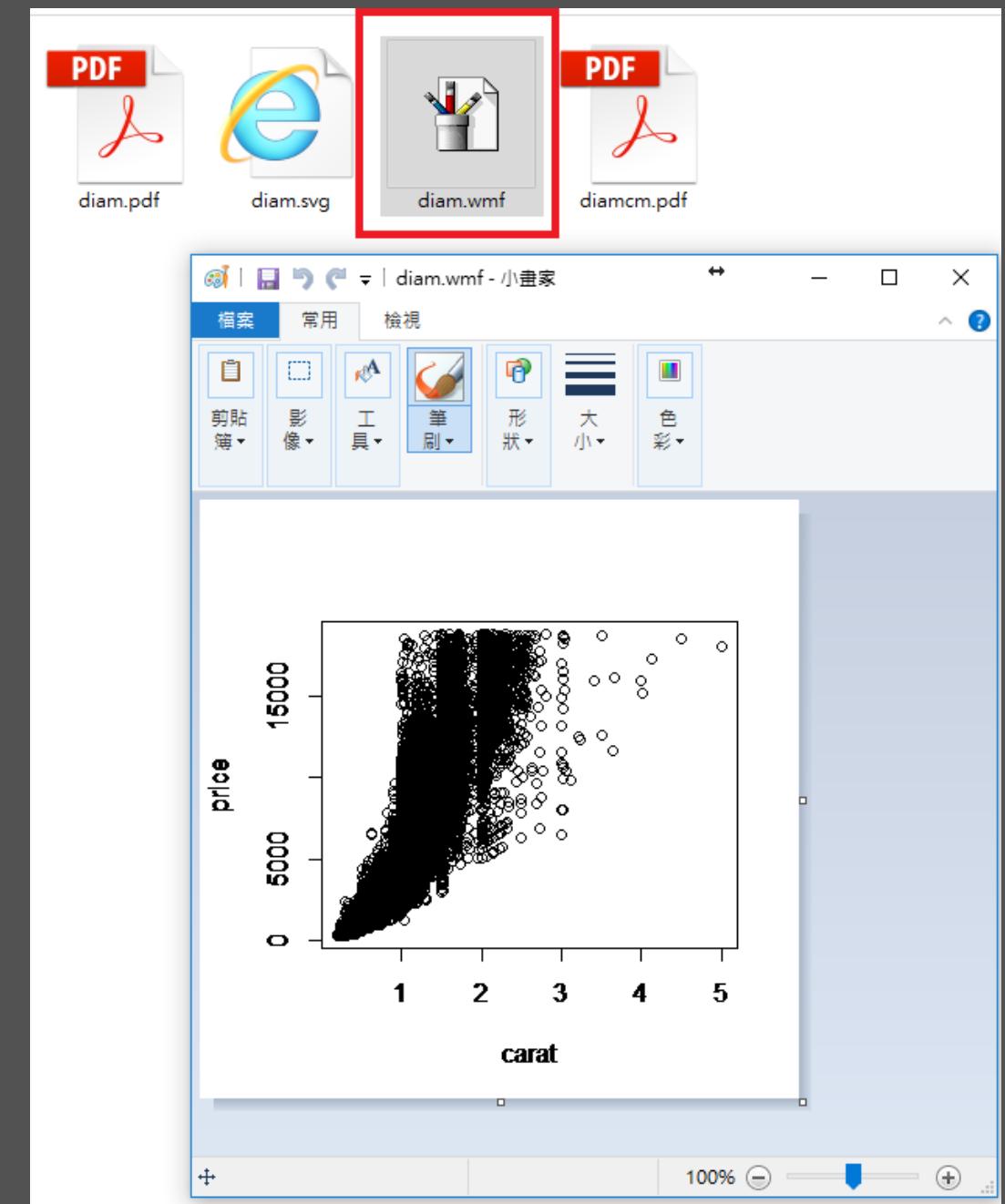
R 圖像輸出(7 種繪圖格式 & ggplot)

SVG



R 圖像輸出 (7 種繪圖格式 & ggplot)

WMF

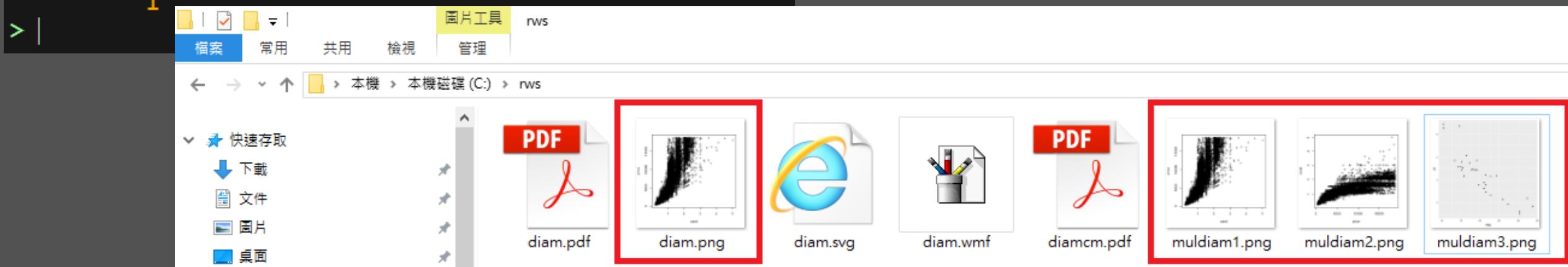


R 圖像輸出(7 種繪圖格式 & ggplot)

PNG 與 TIFF

當中用 %d 可以將中間繪圖的 plot() 指令依序輸出。

```
> # PNG 輸出，單位為像素
> png( file = "diam.png", width = 400, height = 400)
> plot( price ~ carat, data = diamonds)
> dev.off()
null device
    1
>
> # 多個 PNG 輸出，單位為像素
> png( file = "muldiam%d.png", width = 400, height = 400)
> plot( price ~ carat, data = diamonds)
> plot( carat ~ price, data = diamonds)
> print((ggplot(mtcars, aes(mpg, wt)) + geom_point()))
> dev.off()
null device
    1
```

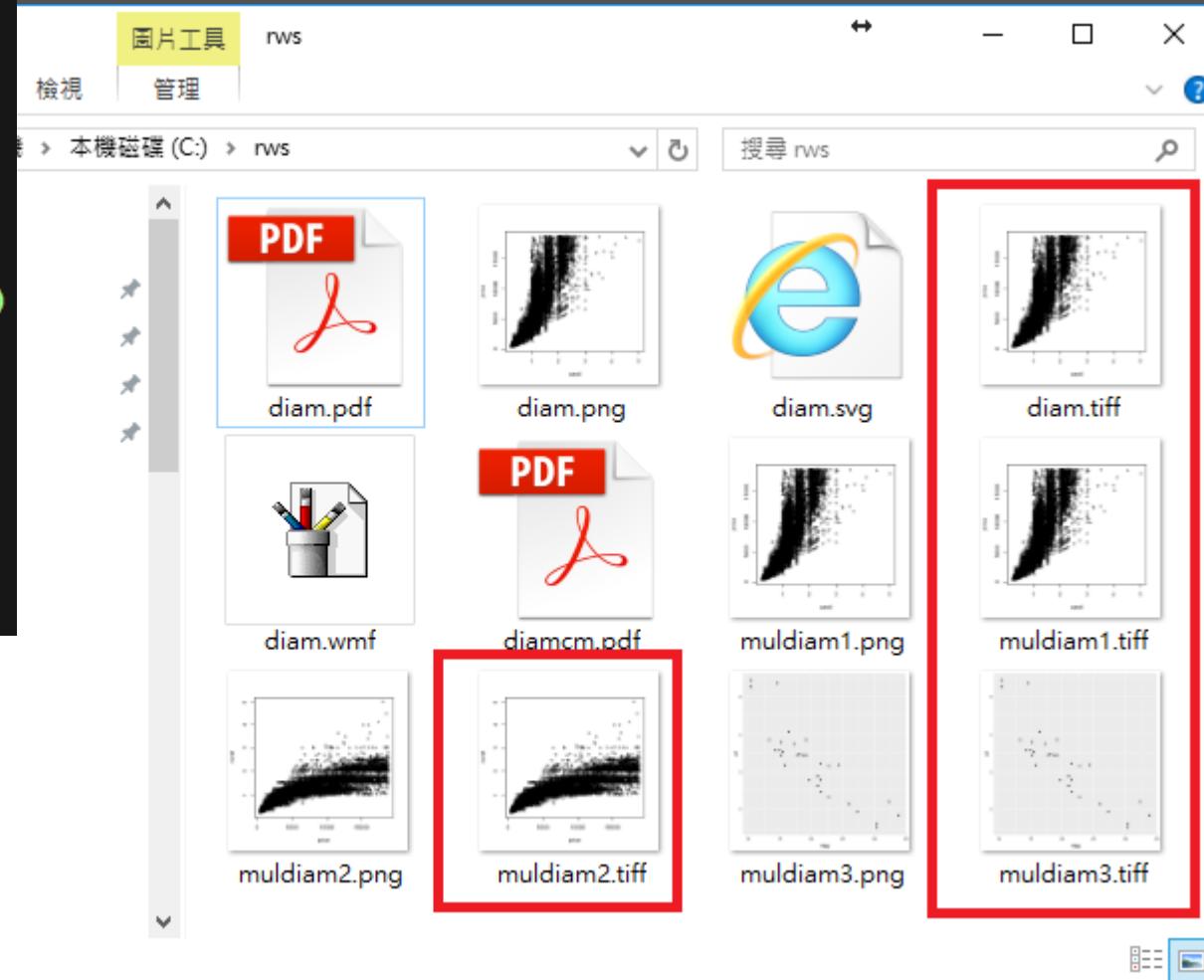
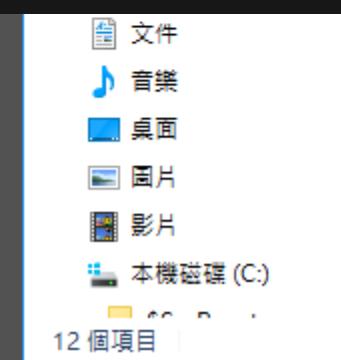


R 圖像輸出(7 種繪圖格式 & ggplot)

PNG 與 TIFF

當中用 %d 可以將中間繪圖的 plot() 指令依序輸出。

```
> # TIFF 輸出，單位為像素
> tiff( file = "diam.tiff", width = 400, height = 400)
> plot( price ~ carat, data = diamonds)
> dev.off()
null device
1
>
> # 多個 TIFF 輸出，單位為像素
> tiff( file = "muldiam%d.tiff", width = 400, height = 400)
> plot( price ~ carat, data = diamonds)
> plot( carat ~ price, data = diamonds)
> print((ggplot(mtcars, aes(mpg, wt)) + geom_point()))
> dev.off()
null device
1
> |
```



R 圖像輸出(7 種繪圖格式 & ggplot)

BMP

```

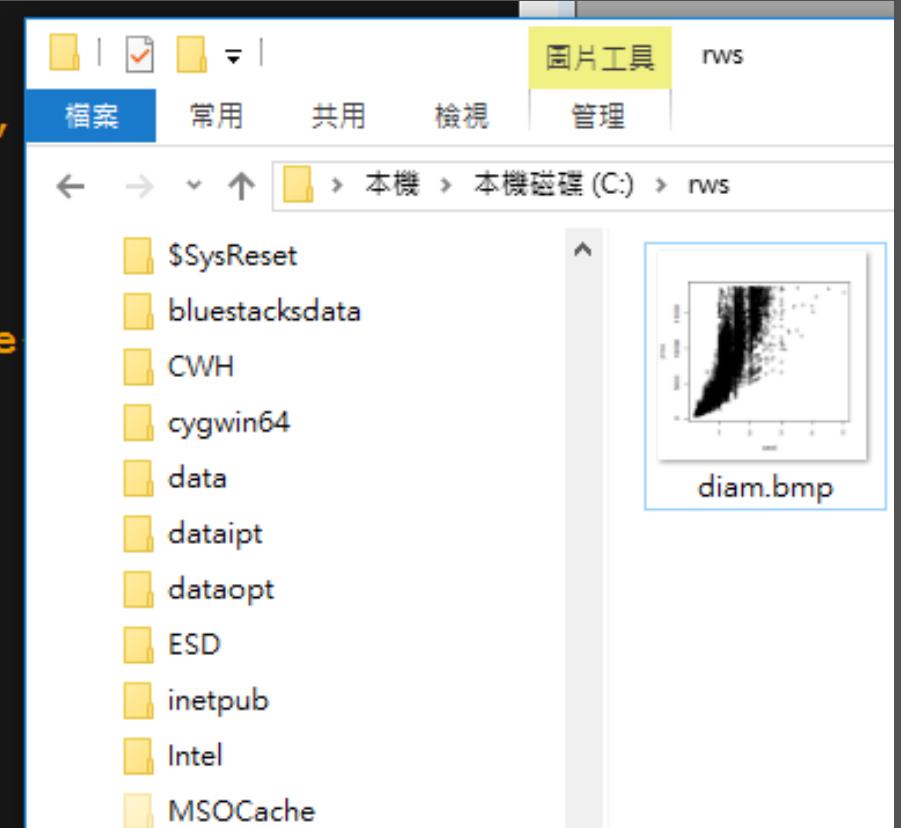
Attaching package: 'data.table'

The following objects are masked from 'package:reshape2'
  dcast, melt

The following objects are masked from 'package:lubridate'
  hour, mday, month, quarter, wday, week, yday, year

> # BMP 輸出，單位為像素
> bmp( file = "diam.bmp", width = 400, height = 400)
> plot( price ~ carat, data = diamonds)
> dev.off()
null device
      1

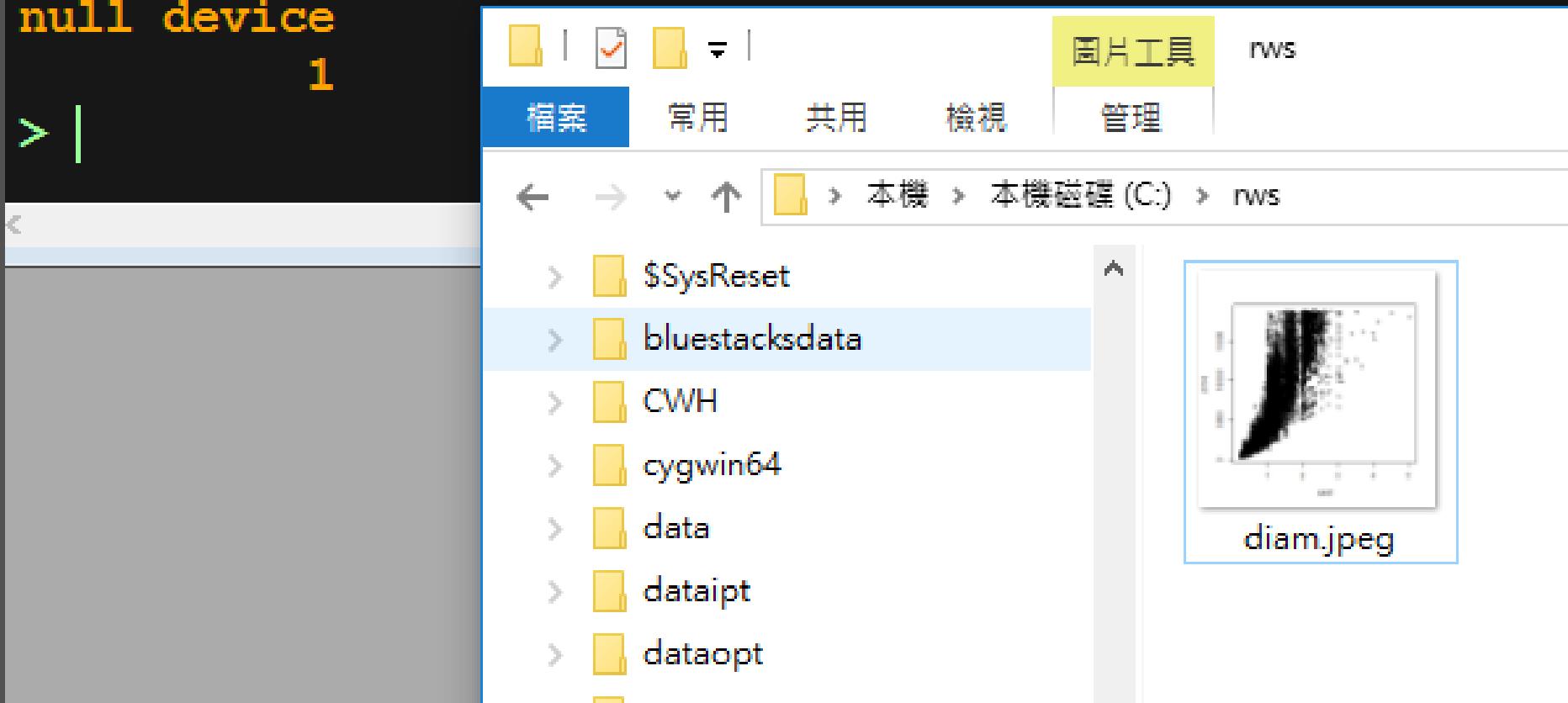
```



R 圖像輸出(7 種繪圖格式 & ggplot)

JPEG

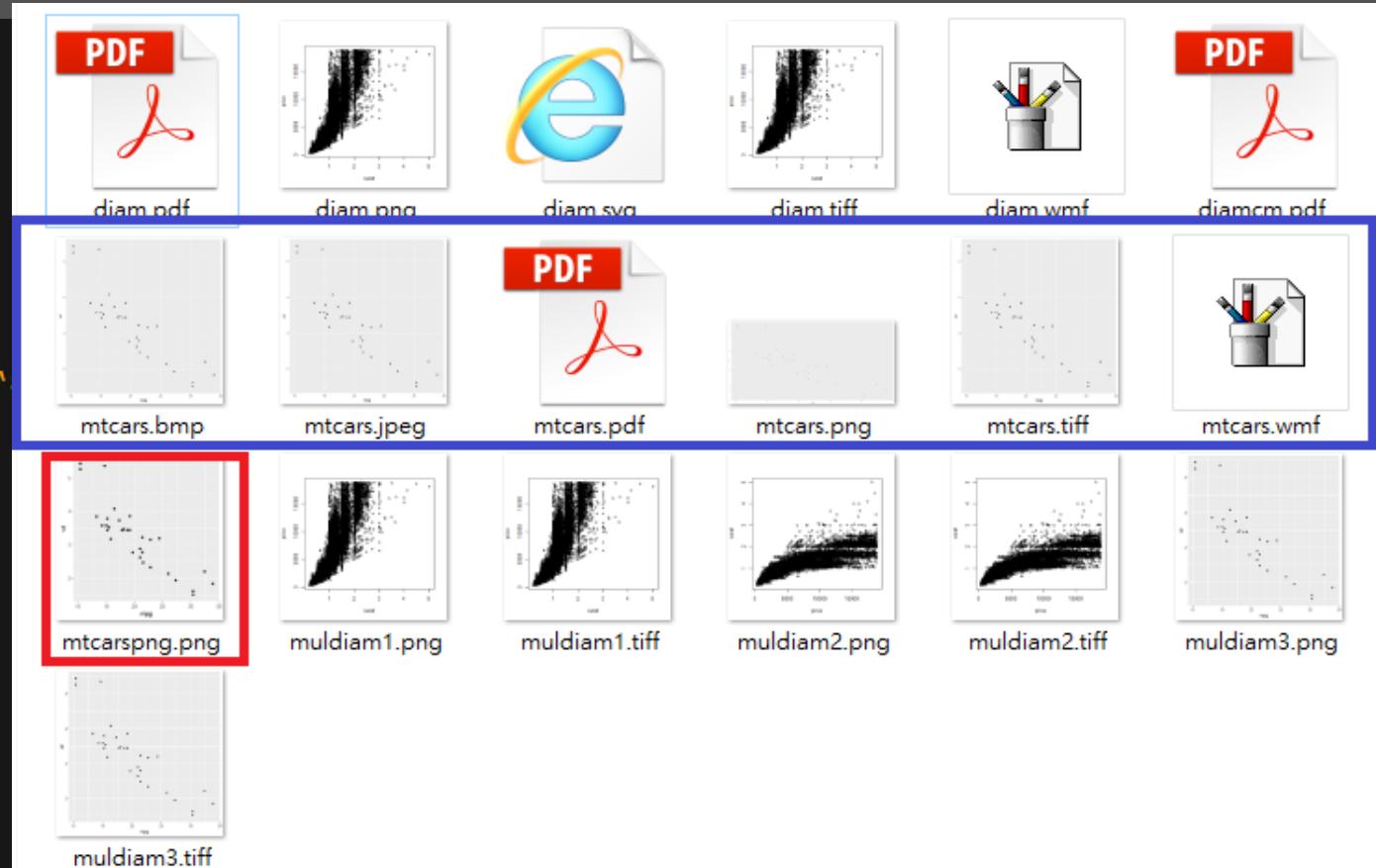
```
> #jpeg 輸出，單位為像素  
> jpeg( file = "diam.jpeg", width = 400, height = 400)  
> plot( price ~ carat, data = diamonds)  
> dev.off()  
null device  
1
```



R 圖像輸出(7 種繪圖格式 & ggplot)

一種是用 ggsave() 進行處理，只要前面用 ggplot() 將圖畫好就可以用，另外一種是用內建的匯出函數，建議在 ggplot() 上用 print() 包起來 -> print(ggplot())。

```
> # ggplot2 套件繪圖輸出之 ggsave()
>
> ggplot(mtcars, aes(mpg, wt)) + geom_point()
> ggsave("mtcars.png")
Saving 20 x 10 in image
> ggsave("mtcars.wmf")
Saving 7 x 6.99 in image
> ggsave("mtcars.svg")
Saving 7 x 6.99 in image
Error in loadNamespace(name) : there is no package called ' 
> ggsave("mtcars.bmp")
Saving 7 x 6.99 in image
> ggsave("mtcars.jpeg")
Saving 7 x 6.99 in image
> ggsave("mtcars.tiff")
Saving 7 x 6.99 in image
> ggsave("mtcars.pdf")
Saving 7 x 6.99 in image
> ggsave("mtcars.pdf", width = 4, height = 4)
> ggsave("mtcars.pdf", width = 4, height = 4, unit = "cm")
>
> # ggplot2 套件繪圖輸出之 print()
> png( file = "mtcarspng.png", res = 120)
> print(ggplot(mtcars, aes(mpg, wt)) + geom_point())
> dev.off()
windows
 2
> |
```



R 圖像輸出(7 種繪圖格式 & ggplot)

PDF 輸出，寬高皆為英寸

```
pdf( file = "diam.pdf", width = 8, height = 8)
plot( price ~ carat, data = diamonds)
dev.off()
```

PDF 輸出，調整成公分

```
pdf( file = "diamcm.pdf", width = 8/2.54, height = 8/2.54)
plot( price ~ carat, data = diamonds)
dev.off()
```

SVG 輸出

```
svg( file = "diam.svg", width = 4, height = 4)
plot( price ~ carat, data = diamonds)
dev.off()
```

WMF 輸出，僅能在Windows 上使用

```
win.metafile( file = "diam.wmf", width = 4, height = 4)
plot( price ~ carat, data = diamonds)
dev.off()
```

PNG 輸出，單位為像素

```
png( file = "diam.png", width = 400, height = 400)
plot( price ~ carat, data = diamonds)
dev.off()
```

多個 PNG 輸出，單位為像素

```
png( file = "muldiam%d.png", width = 400, height = 400)
plot( price ~ carat, data = diamonds)
plot( carat ~ price, data = diamonds)
print(ggplot(mtcars, aes(mpg, wt)) + geom_point())
dev.off()
```

TIFF 輸出，單位為像素

```
tiff( file = "diam.tiff", width = 400, height = 400)
plot( price ~ carat, data = diamonds)
dev.off()
```

R 圖像輸出(7 種繪圖格式 & ggplot)

多個 TIFF 輸出，單位為像素

```
tiff( file = "muldiam%d.tiff", width = 400, height = 400)
plot( price ~ carat, data = diamonds)
plot( carat ~ price, data = diamonds)
print(ggplot(mtcars, aes(mpg, wt)) + geom_point())
dev.off()
```

BMP 輸出，單位為像素

```
bmp( file = "diam.bmp", width = 400, height = 400)
plot( price ~ carat, data = diamonds)
dev.off()
```

#jpeg 輸出，單位為像素

```
jpeg( file = "diam.jpeg", width = 400, height = 400)
plot( price ~ carat, data = diamonds)
dev.off()
```

ggplot2 套件繪圖輸出之 ggsave()

```
ggplot(mtcars, aes(mpg, wt)) + geom_point()
ggsave("mtcars.png")
ggsave("mtcars.wmf")
ggsave("mtcars.svg")
ggsave("mtcars.bmp")
ggsave("mtcars.jpeg")
ggsave("mtcars.tiff")
ggsave("mtcars.pdf")
ggsave("mtcars.pdf", width = 4, height = 4)
ggsave("mtcars.pdf", width = 4, height = 4, unit = "cm")
```

ggplot2 套件繪圖輸出之 print()

```
png( file = "mtcarspng.png", res = 120)
print(ggplot(mtcars, aes(mpg, wt)) + geom_point())
dev.off()
```

R ggplot2 多圖合併

主函式

```

multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  library(grid)
  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)
  numPlots = length(plots)
  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from # of cols
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                    ncol = cols, nrow = ceiling(numPlots/cols))
  }
  if (numPlots==1) {
    print(plots[[1]])
  } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))
    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}

```

R ggplot2 多圖合併

繪出 4 張 ggplot 圖形

後面要合併時，假設有 3 張圖 `multiplot(,,,
cols =)` 則後面的 `cols` 為 2 (因為 2^2)

類推有 7 張圖 `cols` 為 2 會不夠，需要用
`cols = 3` (因為 $3^3 = 9$)

```
library(ggplot2)

# This example uses the ChickWeight dataset, which comes with ggplot2
# First plot
p1 <- ggplot(ChickWeight, aes(x=Time, y=weight, colour=Diet, group=Chick)) +
  geom_line() +
  ggtitle("Growth curve for individual chicks")

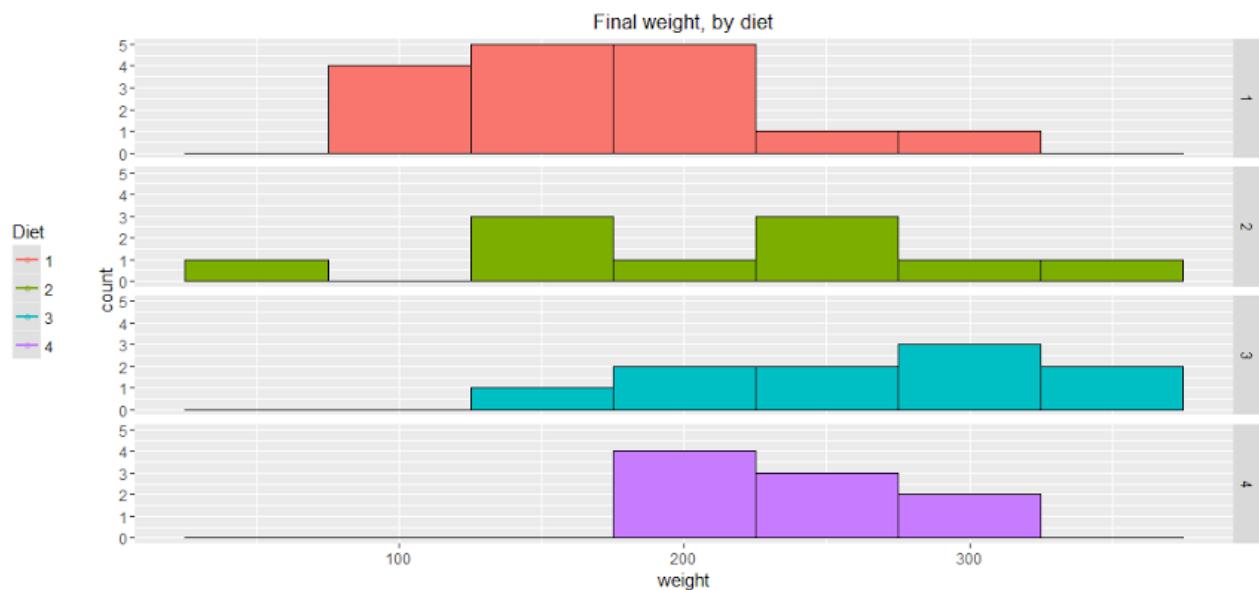
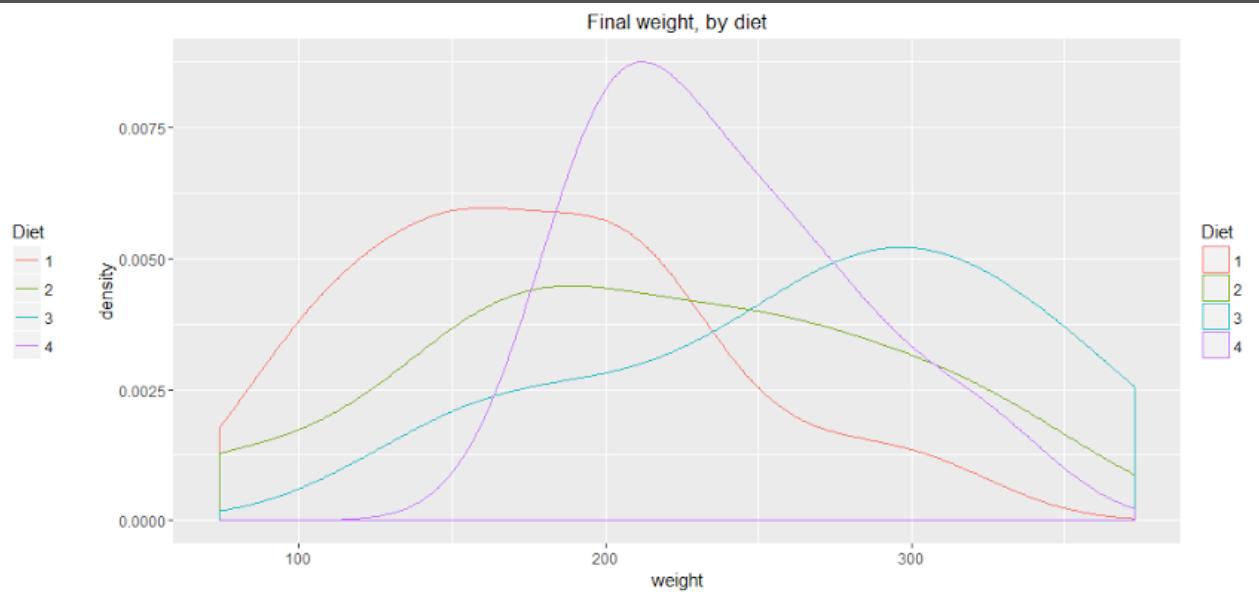
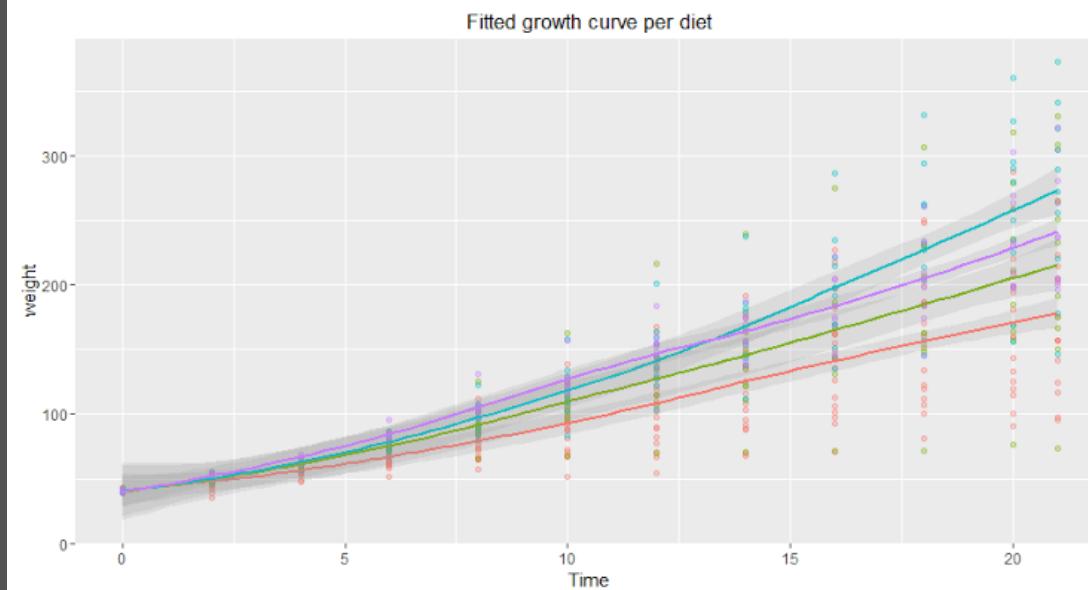
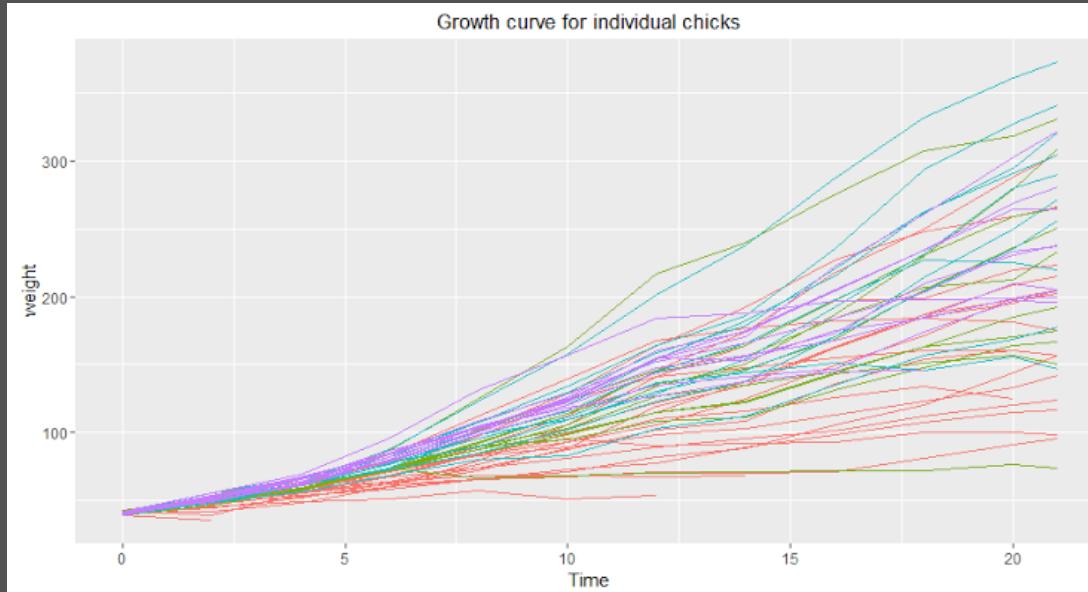
# Second plot
p2 <- ggplot(ChickWeight, aes(x=Time, y=weight, colour=Diet)) +
  geom_point(alpha=.3) +
  geom_smooth(alpha=.2, size=1) +
  ggtitle("Fitted growth curve per diet")

# Third plot
p3 <- ggplot(subset(ChickWeight, Time==21), aes(x=weight, colour=Diet)) +
  geom_density() +
  ggtitle("Final weight, by diet")

# Fourth plot
p4 <- ggplot(subset(ChickWeight, Time==21), aes(x=weight, fill=Diet)) +
  geom_histogram(colour="black", binwidth=50) +
  facet_grid(Diet ~ .) +
  ggtitle("Final weight, by diet") +
  theme(legend.position="none")      # No legend (redundant in this graph)

multiplot(p1, p2, p3, p4, cols=2)
```

R ggplot2 多圖合併



R par 多圖合併

```

> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2   setosa
2          4.9         3.0          1.4         0.2   setosa
3          4.7         3.2          1.3         0.2   setosa
4          4.6         3.1          1.5         0.2   setosa
5          5.0         3.6          1.4         0.2   setosa
6          5.4         3.9          1.7         0.4   setosa
> class(iris)
[1] "data.frame"
> NROW(iris)
[1] 150
>

> iris1.par = par(mai=c(0.3, 0.3, 0.3, 0.3), mfrow=c(3, 2))
> boxplot(iris$Sepal.Length, ylim=c(-1, 8))
> hist(iris$Sepal.Width, xlim=c(-1, 8))
> plot(iris$Petal.Length)
> plot(iris$Petal.Width)
> hist(iris$Petal.Width, xlim=c(-1, 8))
> hist(iris$Petal.Length, xlim=c(-1, 8))
> par(iris1.par)
> par(mai=c(0.3, 0.3, 0.3, 0.3), mfcol=c(2, 3))
>

```

在此隨意利用 IRIS 資料集來做個小測試。
而第一個測試為根據 iris 產生的 3×2 多圖合併。

```

# iris
head(iris)
class(iris)
NROW(iris)

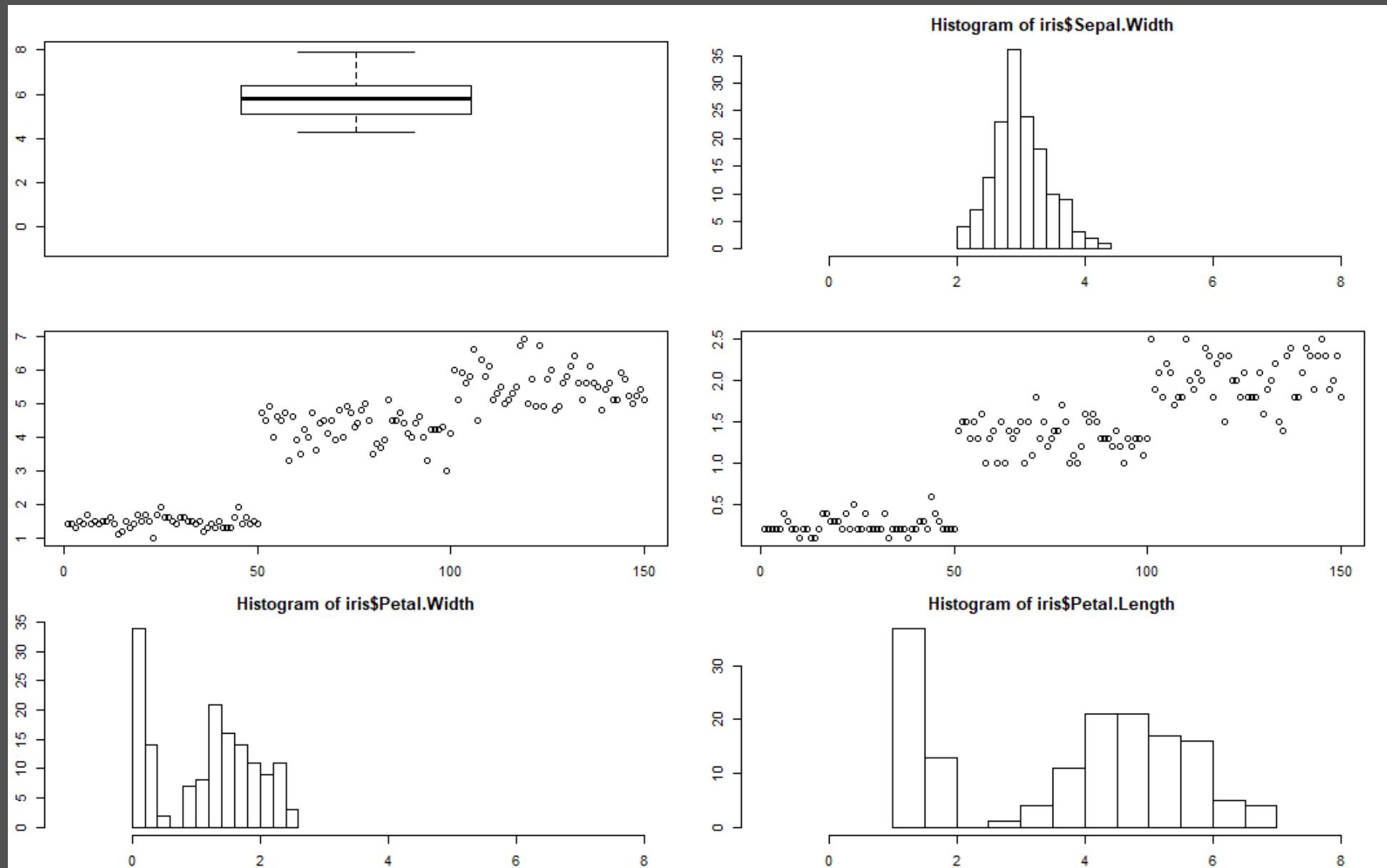
```

```

iris1.par = par(mai=c(0.3, 0.3, 0.3, 0.3),
mfrow=c(3, 2))
boxplot(iris$Sepal.Length, ylim=c(-1, 8))
hist(iris$Sepal.Width, xlim=c(-1, 8))
plot(iris$Petal.Length)
plot(iris$Petal.Width)
hist(iris$Petal.Width, xlim=c(-1, 8))
hist(iris$Petal.Length, xlim=c(-1, 8))
par(iris1.par)
par(mai=c(0.3, 0.3, 0.3, 0.3), mfcol=c(2, 3))

```

R par 多圖合併



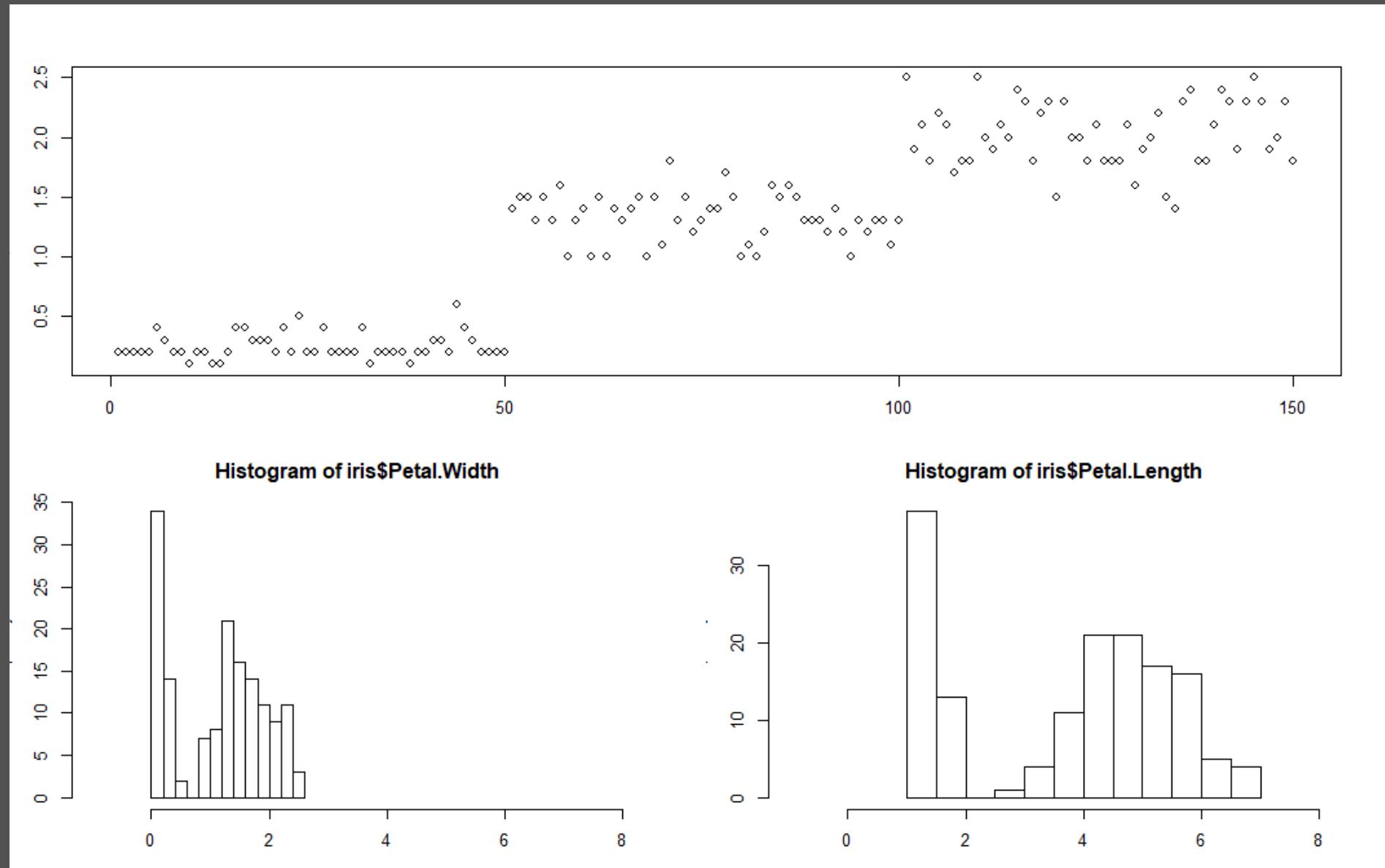
R par 多圖合併

```
> iris2.par = par(mai=c(0.5, 0.5, 0.5, 0.5))
> (irismat1 = matrix(c(1,2,1,3), 2, 2))
 [,1] [,2]
[1,]    1    1
[2,]    2    3
> layout(irismat1)
> plot(iris$Petal.Width)
> hist(iris$Petal.Width, xlim=c(-1, 8))
> hist(iris$Petal.Length, xlim=c(-1, 8))
> par(iris2.par)
> |
```

第二個測試為利用矩陣所編號的三圖合併。

```
iris2.par = par(mai=c(0.5, 0.5, 0.5, 0.5))
(irismat1 = matrix(c(1,2,1,3), 2, 2))
layout(irismat1)
plot(iris$Petal.Width)
hist(iris$Petal.Width, xlim=c(-1, 8))
hist(iris$Petal.Length, xlim=c(-1, 8))
par(iris2.par)
```

R par 多圖合併



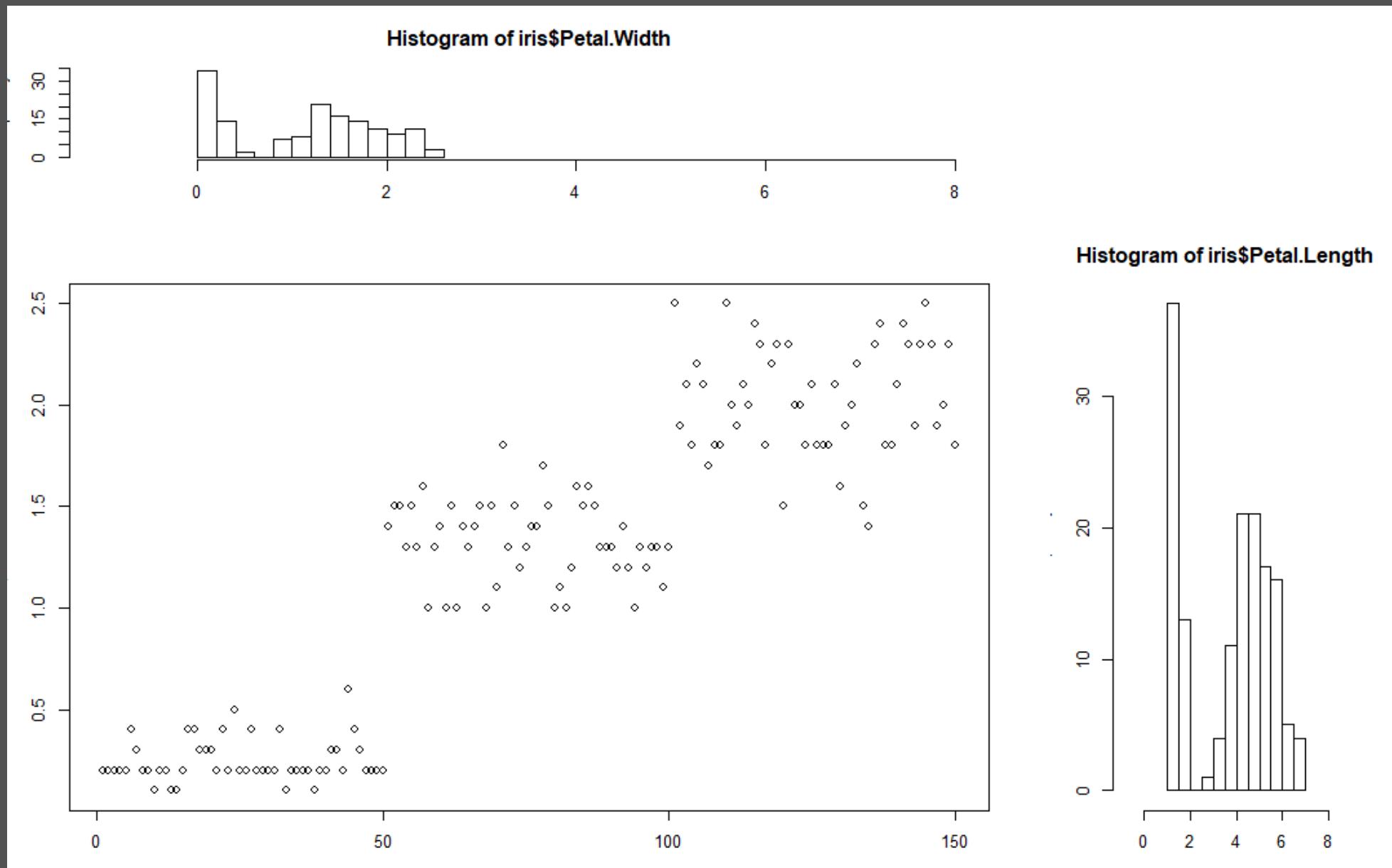
R par 多圖合併

```
> iris3.par = par(mai=c(0.5, 0.5, 0.5, 0.5))
> (irismat2 = matrix(c(2,1,0,3), 2, 2))
 [,1] [,2]
[1,]    2    0
[2,]    1    3
> layout(irismat2, widths=c(3, 1), heights=c(1, 3))
> plot(iris$Petal.Width)
> hist(iris$Petal.Width, xlim=c(-1, 8))
> hist(iris$Petal.Length, xlim=c(-1, 8))
> par(iris3.par)
> |
```

第三個範例可以看到矩陣編號為零的不會顯示圖片，並且會配合 layout() 調整。

```
iris3.par = par(mai=c(0.5, 0.5, 0.5, 0.5))
(irismat2 = matrix(c(2,1,0,3), 2, 2))
layout(irismat2, widths=c(3, 1),
heights=c(1, 3))
plot(iris$Petal.Width)
hist(iris$Petal.Width, xlim=c(-1, 8))
hist(iris$Petal.Length, xlim=c(-1, 8))
par(iris3.par)
```

R par 多圖合併



R par 多圖合併

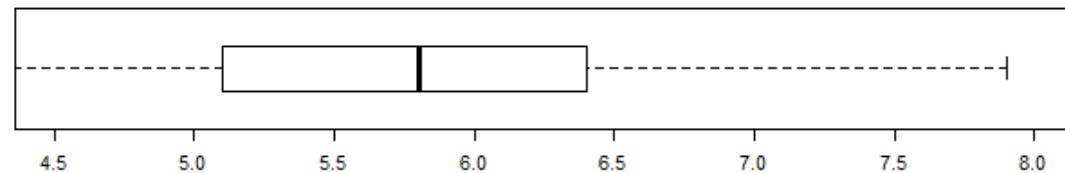
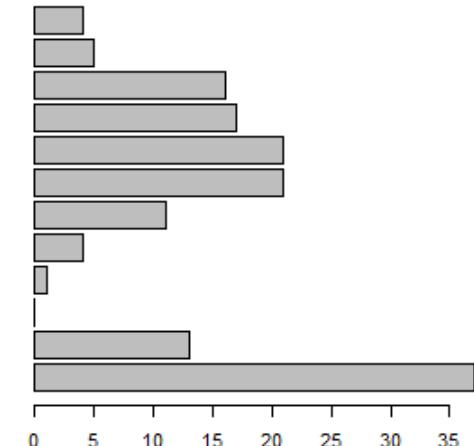
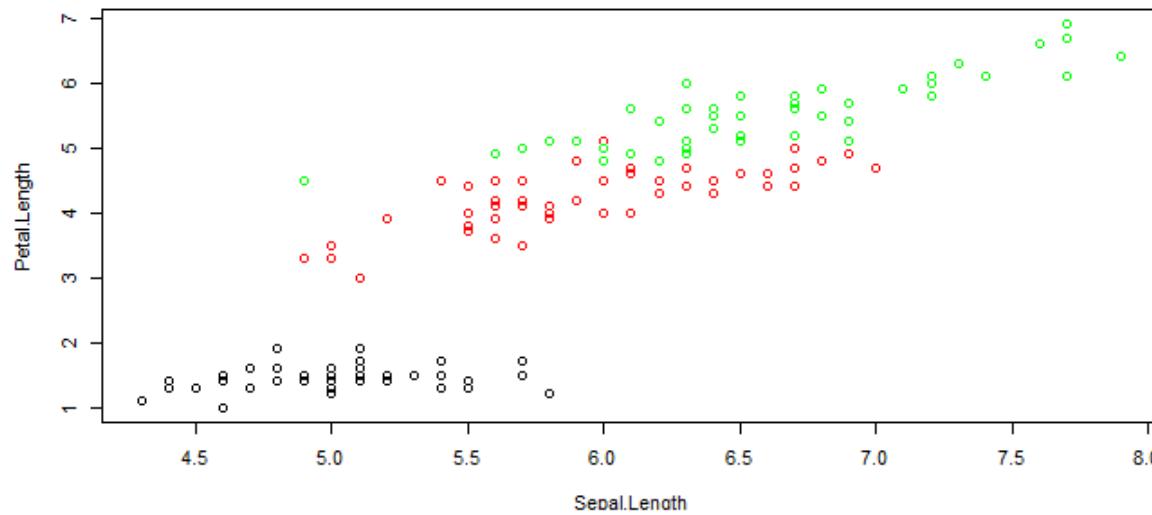
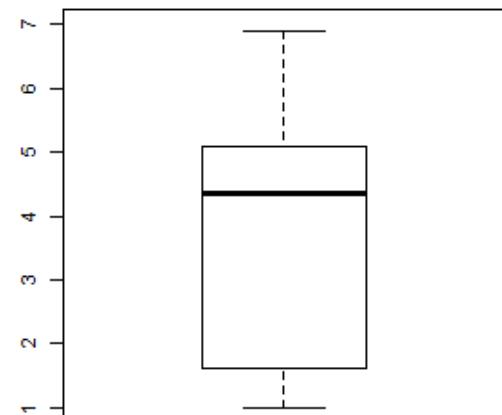
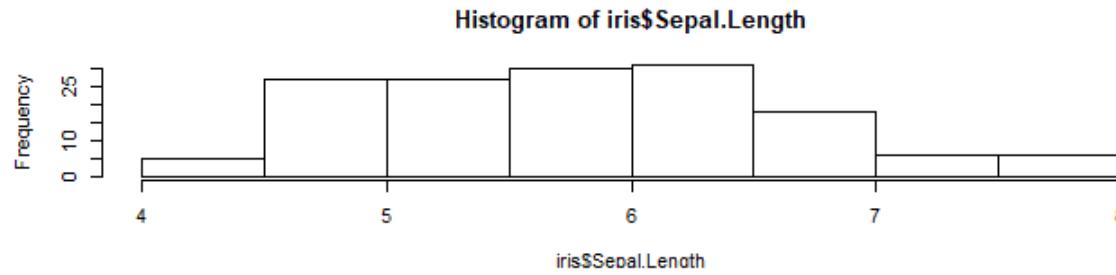
67

第四個範例。

```
iris4.par = par(mai=c(0.5, 0.5, 0.5, 0.5))
(irismat3 = matrix(c( 0, 1, 0, 2, 3, 4, 0, 5, 0), 3, 3))
layout(irismat3, widths=c( 2, 4, 2), heights=c(2, 4, 2))
boxplot(iris$Petal.Length, ylim = c( 1, 7))
hist(iris$Sepal.Length, xlim = c( 4, 8))
Sepal.Length = iris$Sepal.Length
Petal.Length = iris$Petal.Length
plot(Sepal.Length, Petal.Length)
points(Sepal.Length [1:50], Petal.Length[1:50])
points(Sepal.Length [51:100], Petal.Length[51:100], col="red")
points(Sepal.Length [101:150], Petal.Length[101:150], col="green")
boxplot(iris$Sepal.Length, horizontal=T, ylim = c( 4.5, 8.0))
templ = hist(Petal.Length, plot = FALSE)$count
barplot(templ, horiz=TRUE)
par(iris4.par)
```

```
> iris4.par = par(mai=c(0.5, 0.5, 0.5, 0.5))
> (irismat3 = matrix(c( 0, 1, 0, 2, 3, 4, 0, 5, 0), 3, 3))
      [,1] [,2] [,3]
[1,]    0    2    0
[2,]    1    3    5
[3,]    0    4    0
> layout(irismat3, widths=c( 2, 4, 2), heights=c(2, 4, 2))
> boxplot(iris$Petal.Length, ylim = c( 1, 7))
> hist(iris$Sepal.Length, xlim = c( 4, 8))
>
> Sepal.Length = iris$Sepal.Length
> Petal.Length = iris$Petal.Length
> plot(Sepal.Length, Petal.Length)
> points(Sepal.Length [1:50], Petal.Length[1:50])
> points(Sepal.Length [51:100], Petal.Length[51:100], col="red")
> points(Sepal.Length [101:150], Petal.Length[101:150], col="green")
> boxplot(iris$Sepal.Length, horizontal=T, ylim = c( 4.5, 8.0))
> templ = hist(Petal.Length, plot = FALSE)$count
> barplot(templ, horiz=TRUE)
>
> par(iris4.par)
> |
```

R par 多圖合併



R 散佈圖標籤處理

```
> install.packages("ggrepel")
Installing package into 'D:/USERDATA/Documents/R/win-library/3.3'
(as 'lib' is unspecified)
嘗試 URL 'https://cloud.r-project.org/bin/windows/contrib/3.3/ggrepel_0.6.5.zip'
Content type 'application/zip' length 893561 bytes (872 KB)
downloaded 872 KB

package 'ggrepel' successfully unpacked and MD5 sums checked
```

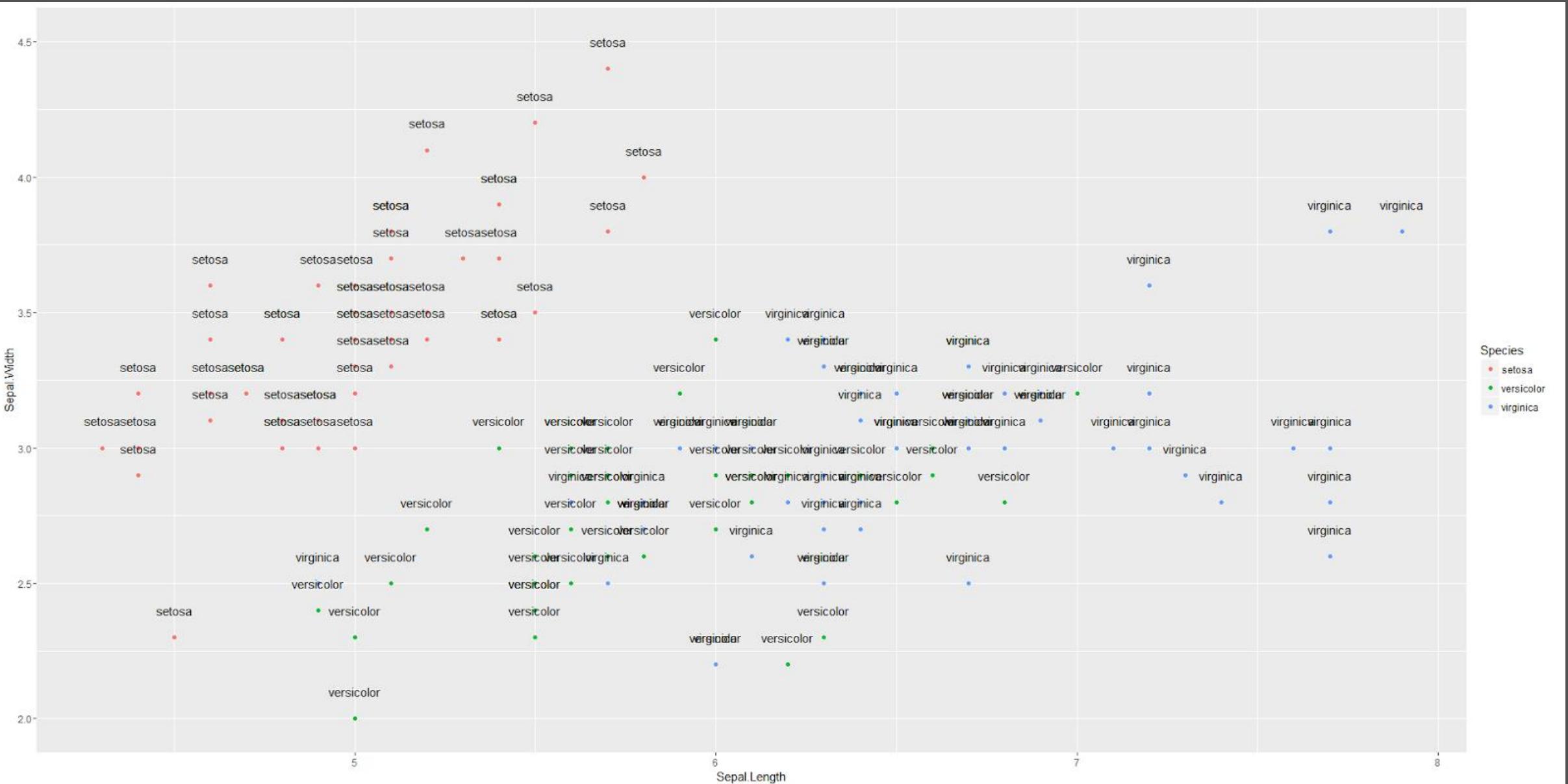
第一種方式

```
# 原始繪圖
library(ggplot2)
ggplot(iris)+
```

```
  geom_point(aes(x=Sepal.Length,y=Sepal.Width,colour=Species))+
  geom_text(aes(x=Sepal.Length,y=Sepal.Width+0.1,label=Species))
```

```
# install.packages("ggrepel")
library(ggrepel)
ggplot(iris) +
  geom_point(aes( x = Sepal.Length, y = Sepal.Width, colour = Species )) +
  geom_text_repel(aes( x = Sepal.Length, y = Sepal.Width, label = Species ))
```

R 散佈圖標籤處理



R 散佈圖標籤處理

```
> library(ggplot2)
> mydf <- data.frame(class=rep(c("dbms", "mis"), each=5),
+                      club=rep(paste0("clu.", 1:5), 2),
+                      len=c(80, 76, 65, 80, 81, 76, 85, 60, 78, 60))
> mydf
  class club len
1 dbms clu.1  80
2 dbms clu.2  76
3 dbms clu.3  65
4 dbms clu.4  80
5 dbms clu.5  81
6 mis  clu.1  76
7 mis  clu.2  85
8 mis  clu.3  60
9 mis  clu.4  78
10 mis  clu.5  60
> |
```

第二種方式

```
library(ggplot2)
mydf<- data.frame(class=rep(c("dbms", "mis"), each=5),
                    club=rep(paste0("clu.", 1:5), 2),
                    len=c(80, 76, 65, 80, 81, 76, 85, 60, 78, 60))
```

R 散佈圖標籤處理

```
> row.names(mydf) <- LETTERS[1:nrow(mydf)]  
> mydf  
  class   club len  
A dbms clu.1 80  
B dbms clu.2 76  
C dbms clu.3 65  
D dbms clu.4 80  
E dbms clu.5 81  
F mis clu.1 76  
G mis clu.2 85  
H mis clu.3 60  
I mis clu.4 78  
J mis clu.5 60  
> |
```

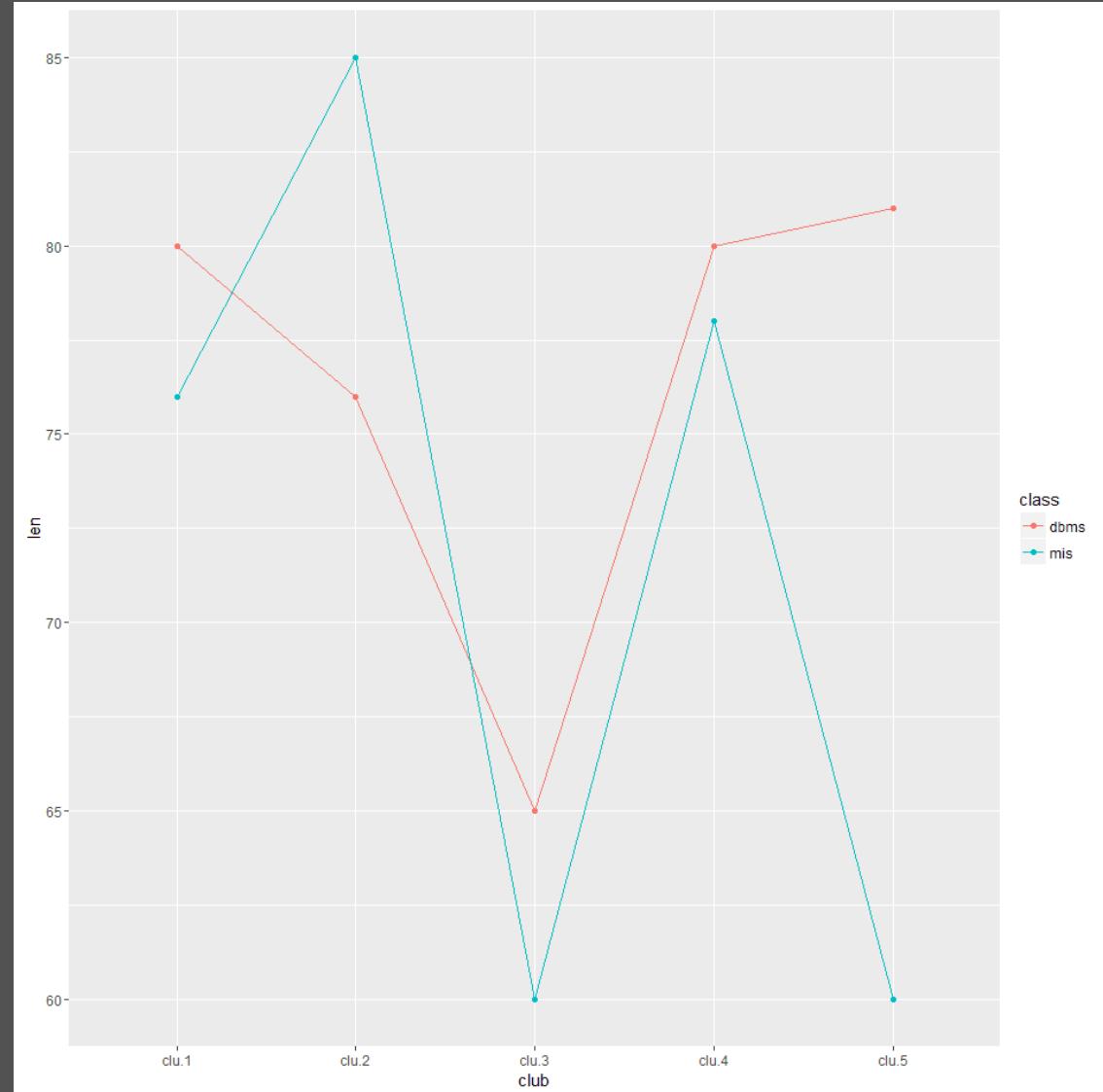
產生大寫的英文字母作為欄位編號。

```
row.names(mydf) <- LETTERS[1:nrow(mydf)]
```

R 散佈圖標籤處理

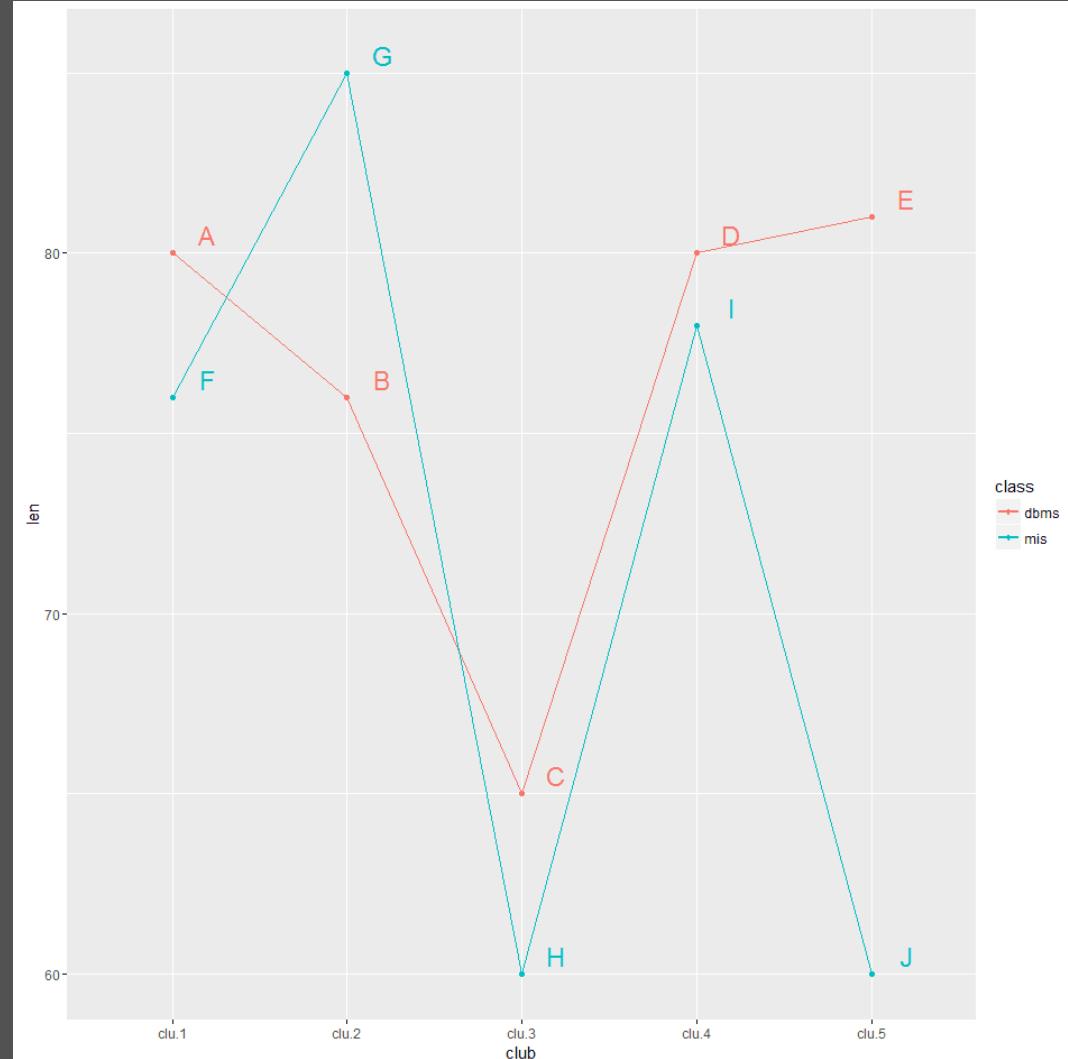
```
p <- ggplot(data=mydf, aes(x=club, y=len, group=class, label=rownames(mydf))) +  
  geom_line(aes(color=class)) +  
  geom_point(aes(color=class))
```

```
p
```



R 散佈圖標籤處理

```
p + geom_text(size=6, nudge_x=0.2, nudge_y=0.5, aes(color=class)) +  
guides(colour=guide_legend(override.aes = list(size=1,linetype=1)))
```



R 開啟多個繪圖視窗

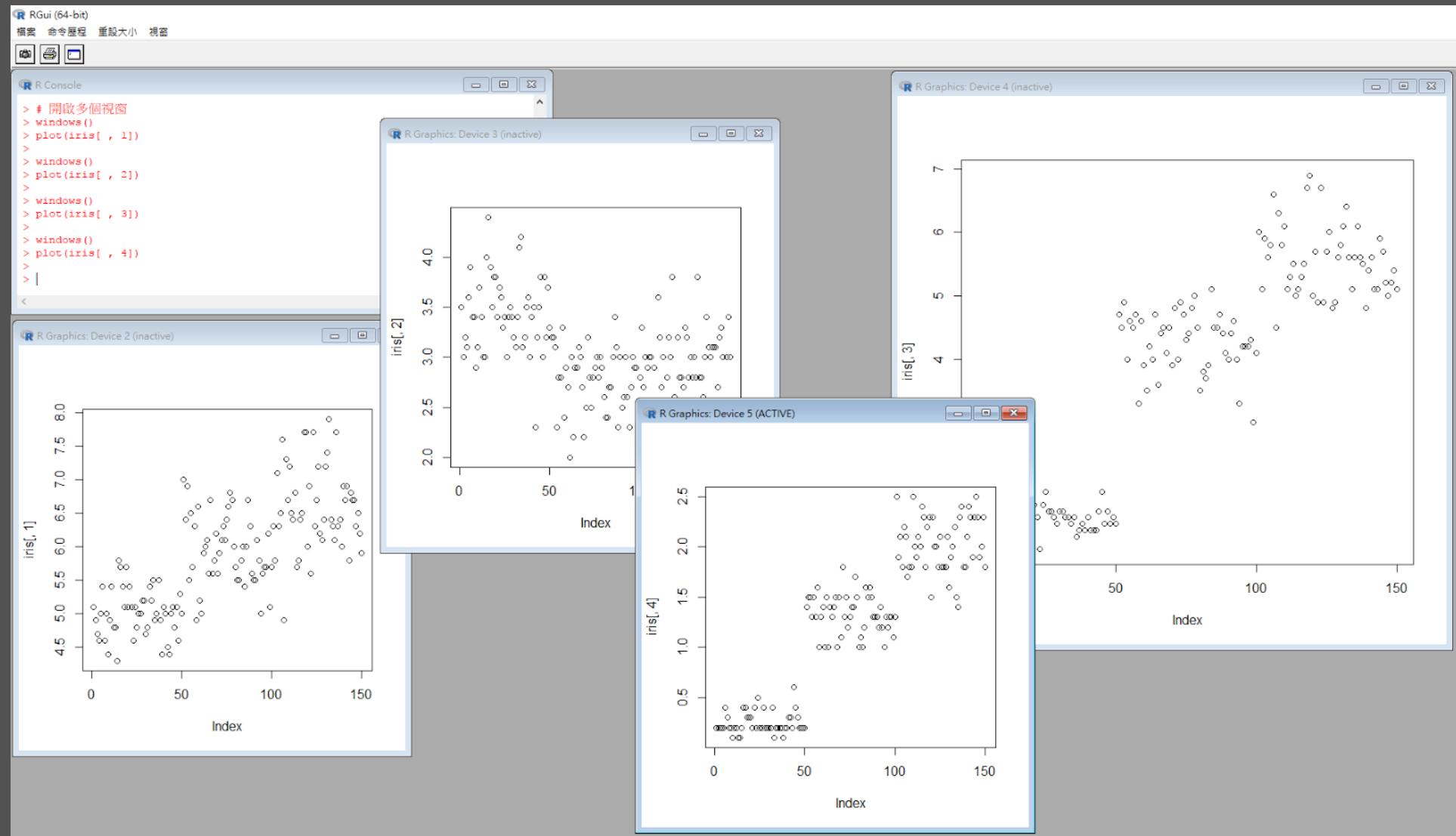
一個重要的函數為
`windows()`，可以開多個
 繪圖的視窗。

```
# 開啟多個視窗
windows()
plot(iris[, 1])
```

```
windows()
plot(iris[, 2])
```

```
windows()
plot(iris[, 3])
```

```
windows()
plot(iris[, 4])
```



R 修改繪圖字體

76

根據 R 內建的繪圖方式，修改繪圖的字體，安裝
showtext 套件。並利用思源黑體的 Thin。

```
# install.packages("showtext")
```

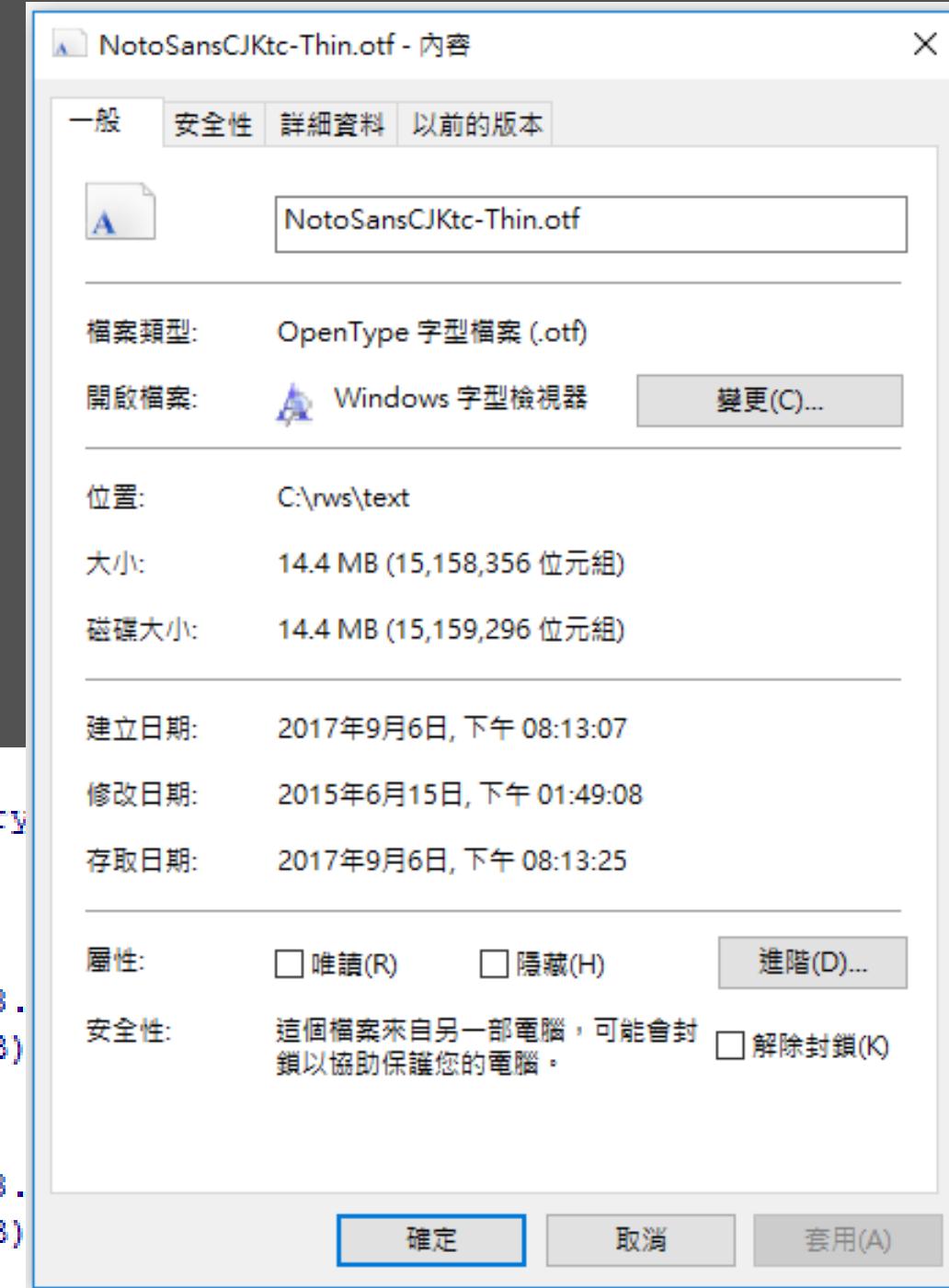
R 修改繪圖字體

在此將思源黑體的檔案用 `setwd()` 設定來源的工作目錄，將字型的檔案抓進去。

```
> install.packages("showtext")
Installing package into 'D:/USERDATA/Documents/R/win-library
(as 'lib' is unspecified)
also installing the dependencies 'sysfonts', 'showtextdb'
```

```
嘗試 URL 'https://cloud.r-project.org/bin/windows/contrib/3.
Content type 'application/zip' length 1673256 bytes (1.6 MB)
downloaded 1.6 MB
```

```
嘗試 URL 'https://cloud.r-project.org/bin/windows/contrib/3.
Content type 'application/zip' length 1992546 bytes (1.9 MB)
downloaded 1.9 MB
```



R 修改繪圖字體

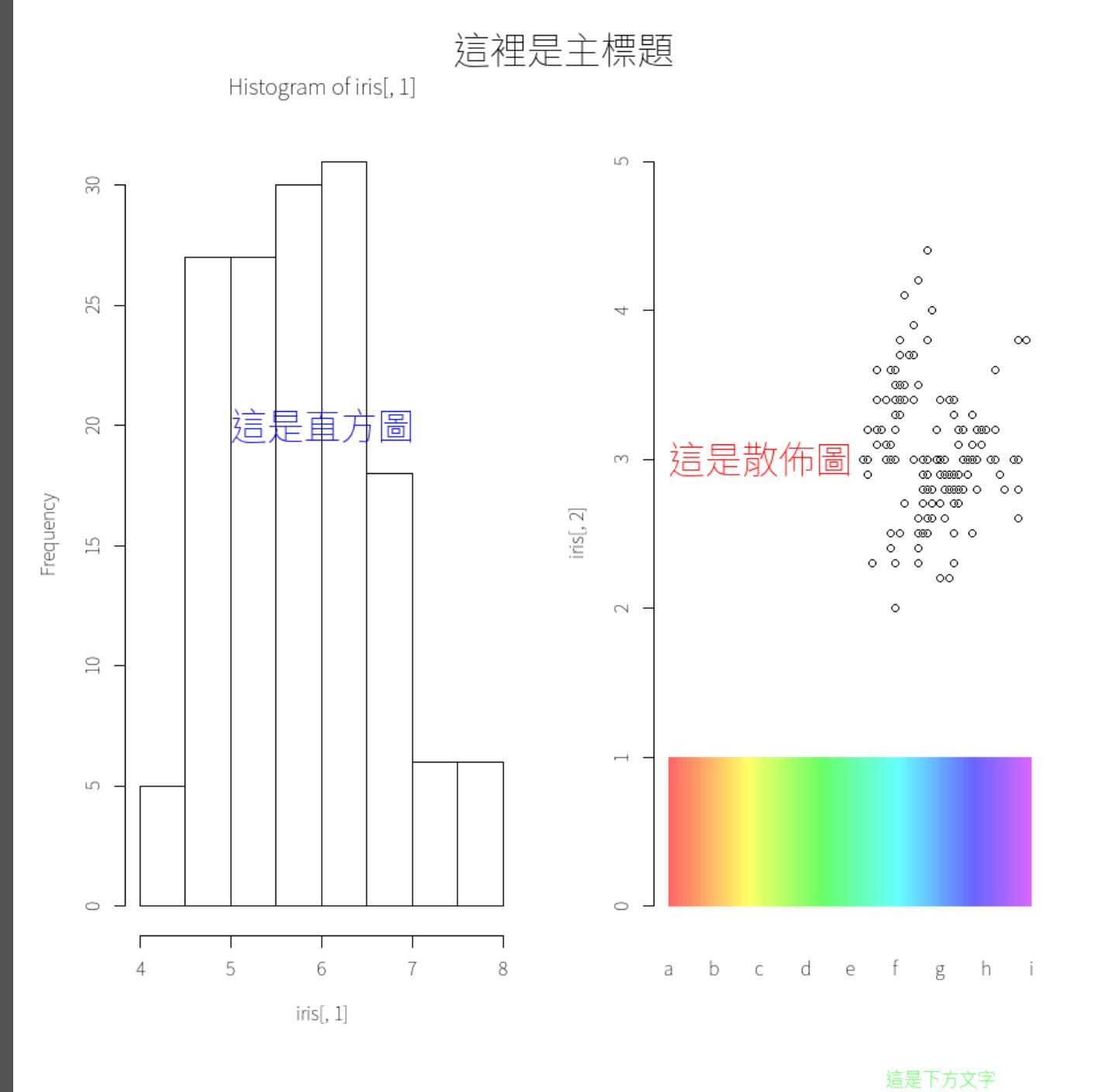
78

```
> library(showtext)
> setwd("C:/rws/text")
> dir()
[1] "NotoSansCJKtc-Black.otf"      "NotoSansCJKtc-Bold.otf"
[3] "NotoSansCJKtc-DemiLight.otf"  "NotoSansCJKtc-Light.otf"
[5] "NotoSansCJKtc-Medium.otf"    "NotoSansCJKtc-Regular.otf"
[7] "NotoSansCJKtc-Thin.otf"
> font.add("NSCT", "NotoSansCJKtc-Thin.otf")
> showtext.auto(enable = TRUE)
> par( family = 'NSCT')
> par(mfrow = c( 1, 2), oma = c( 2, 1, 2, 1))
> hist(iris[ , 1])
> text(6, 20, "這是直方圖", col = "blue", cex = 2)
> plot( iris[ , 1], iris[ , 2],
+ xaxt = "n", bty = "n",
+ xlab = "", xlim = c(0, 8),
+ ylim = c( 0, 5))
> axis( 1, at = 0:8 ,
+ labels = letters[ 1: 9],
+ tick = FALSE)
> text( 2, 3, "這是散佈圖",
+ col = "red",
+ cex = 2)
> title( '這裡是主標題',
+ family = "NSCT",
+ outer = TRUE,
+ line = -1,
+ cex.main = 2)
> mtext("這是下方文字", side=1,
+ line = 6, at = 6, col = "green")
> mycolor = rainbow( 100, alpha = 0.6)[1:80]
> rasterImage(t(mycolor), 0, 0, 8, 1, interpolate = FALSE)
```

```
library("showtext")
setwd("path dir")
dir()
font.add("NSCT", "NotoSansCJKtc-Thin.otf")
showtext.auto(enable = TRUE)
par(family = 'NSCT')
par(mfrow = c( 1, 2), oma = c( 2, 1, 2, 1))
hist(iris[ , 1])
text( 6, 20, "這是直方圖", col = "blue", cex = 2)
plot( iris[ , 1], iris[ , 2],
xaxt = "n", bty = "n",
xlab = "", xlim = c( 0, 8),
ylim = c( 0, 5))
axis( 1, at = 0:8,
labels = letters[1:9],
tick = FALSE)
text( 2, 3, "這是散佈圖",
col = "red",
cex = 2)
title( '這裡是主標題', family = "NSCT",
outer = TRUE,
line = -1,
cex.main = 2)
mtext("這是下方文字", side = 1,
line = 6, at = 6, col = "green")
mycolor = rainbow( 100, alpha = 0.6)[1:80]
rasterImage(t(mycolor), 0, 0, 8, 1, interpolate = FALSE)
```

R 修改繪圖字體

79

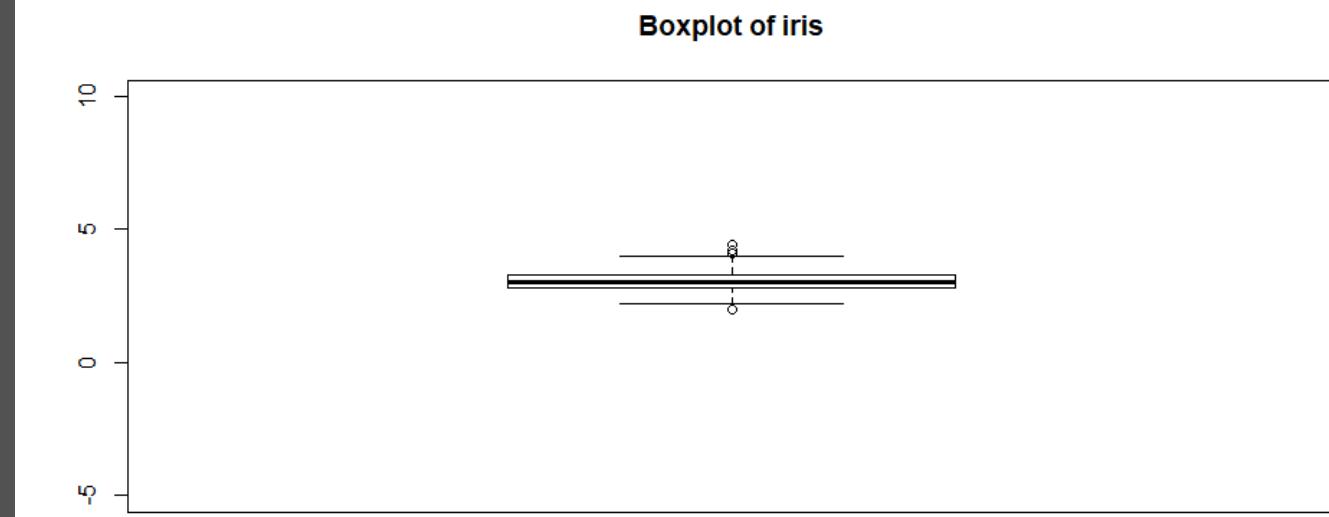
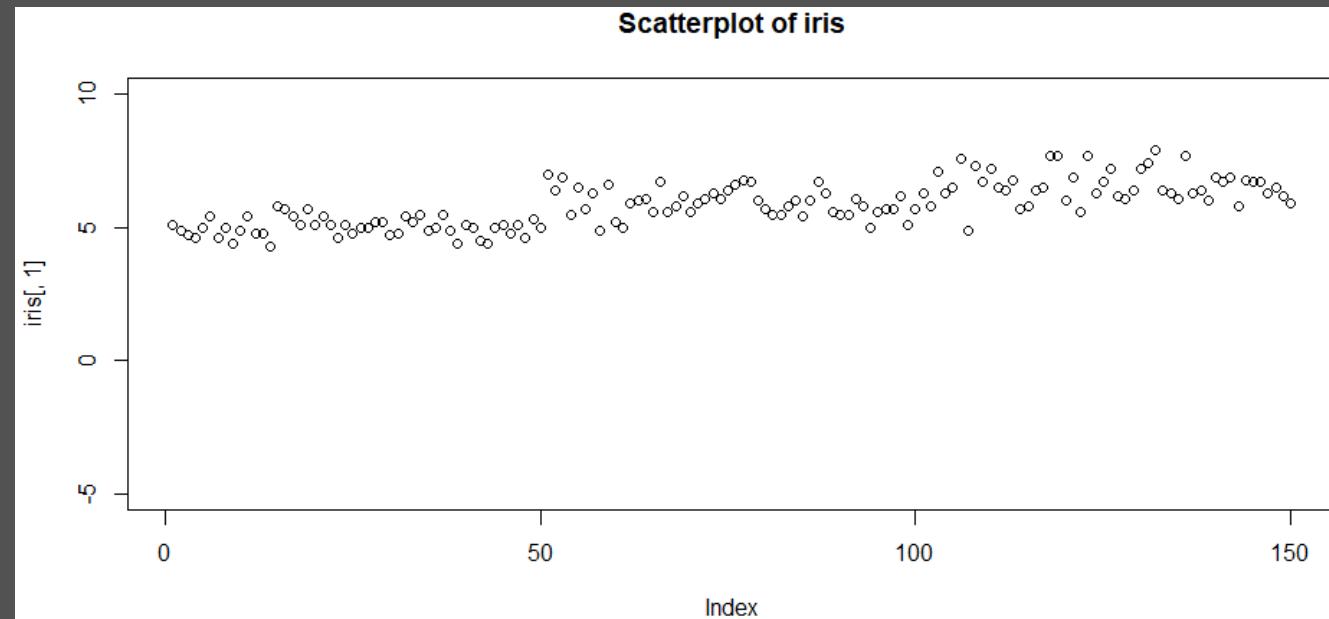


R 繪圖參數設定與劃線 - plot

1. 改變 y 軸與 x 軸的刻度
 第一張圖的 y 軸範圍是 -5 ~ 10。

```
windows()
par(mfrow = c(2,1))
s.title = "Scatterplot of iris"
plot(iris[,1], ylim = c(-5, 10), main = s.title)
b.title = "Boxplot of iris"
boxplot(iris[,2], ylim = c(-5, 10), main = b.title)
```

```
> windows()
> par(mfrow = c(2,1))
> s.title = "Scatterplot of iris"
> plot(iris[,1], ylim = c(-5, 10), main = s.title)
> b.title = "Boxplot of iris"
> boxplot(iris[,2], ylim = c(-5, 10), main = b.title)
> |
```



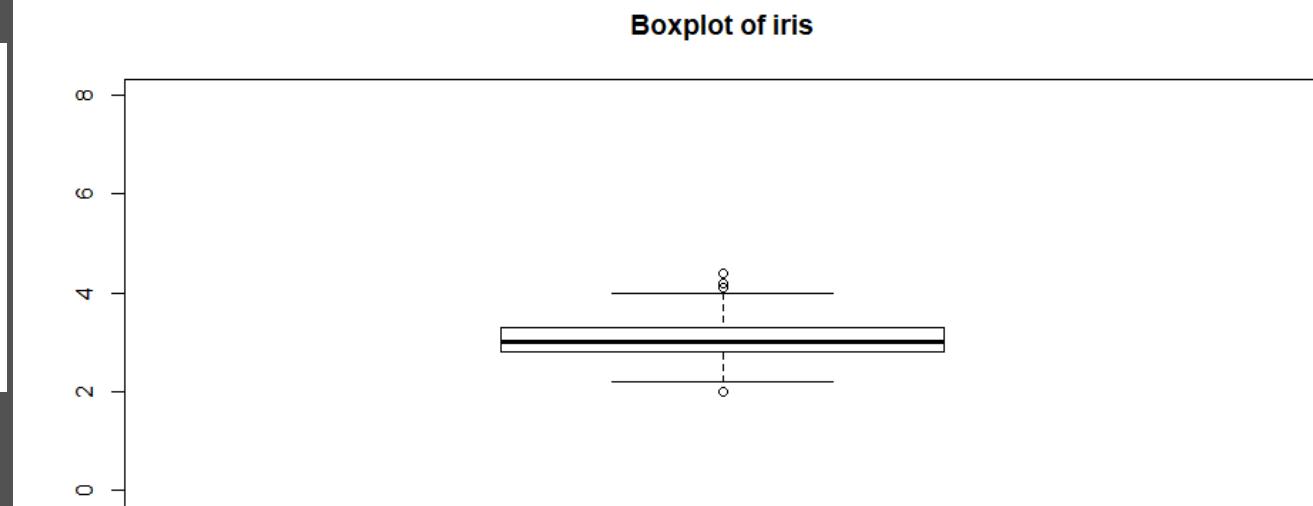
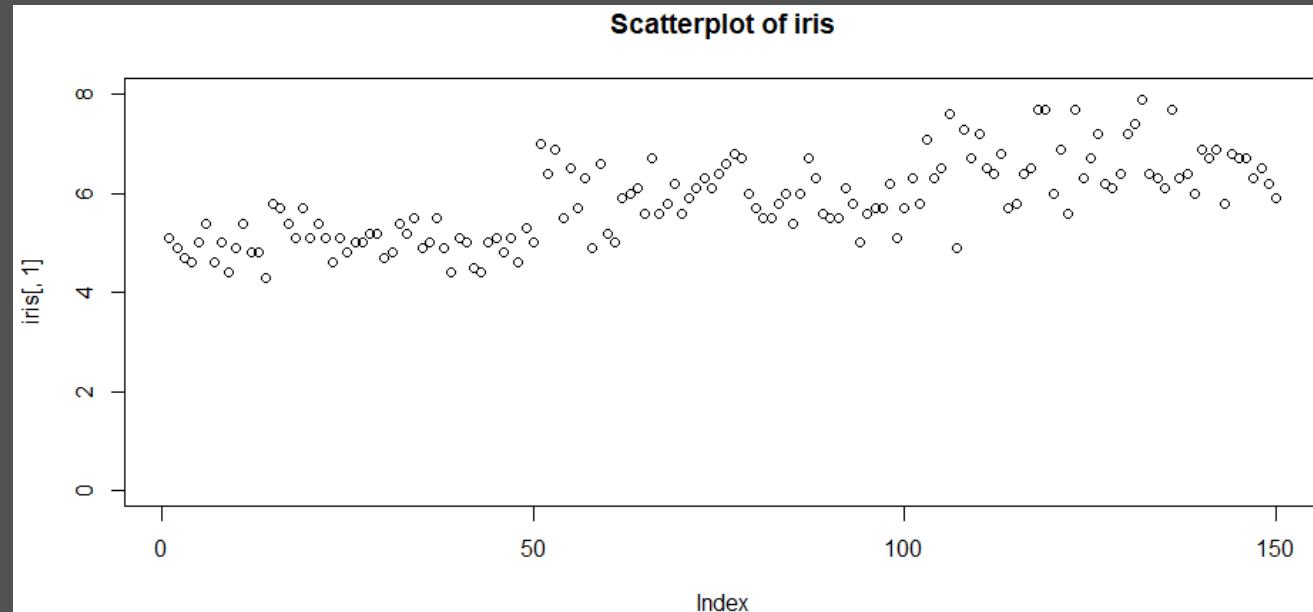
R 繪圖參數設定與劃線 - plot

1. 改變 y 軸與 x 軸的刻度

第二張圖的 y 軸範圍則是 0 ~ 8。

```
windows()
par(mfrow = c(2,1))
s.title = "Scatterplot of iris"
plot(iris[,1], ylim = c(0, 8), main = s.title)
b.title = "Boxplot of iris"
boxplot(iris[,2], ylim = c(0, 8), main = b.title)
```

```
> windows()
> par(mfrow = c(2,1))
> s.title = "Scatterplot of iris"
> plot(iris[,1], ylim = c(0, 8), main = s.title)
> b.title = "Boxplot of iris"
> boxplot(iris[,2], ylim = c(0, 8), main = b.title)
> |
```



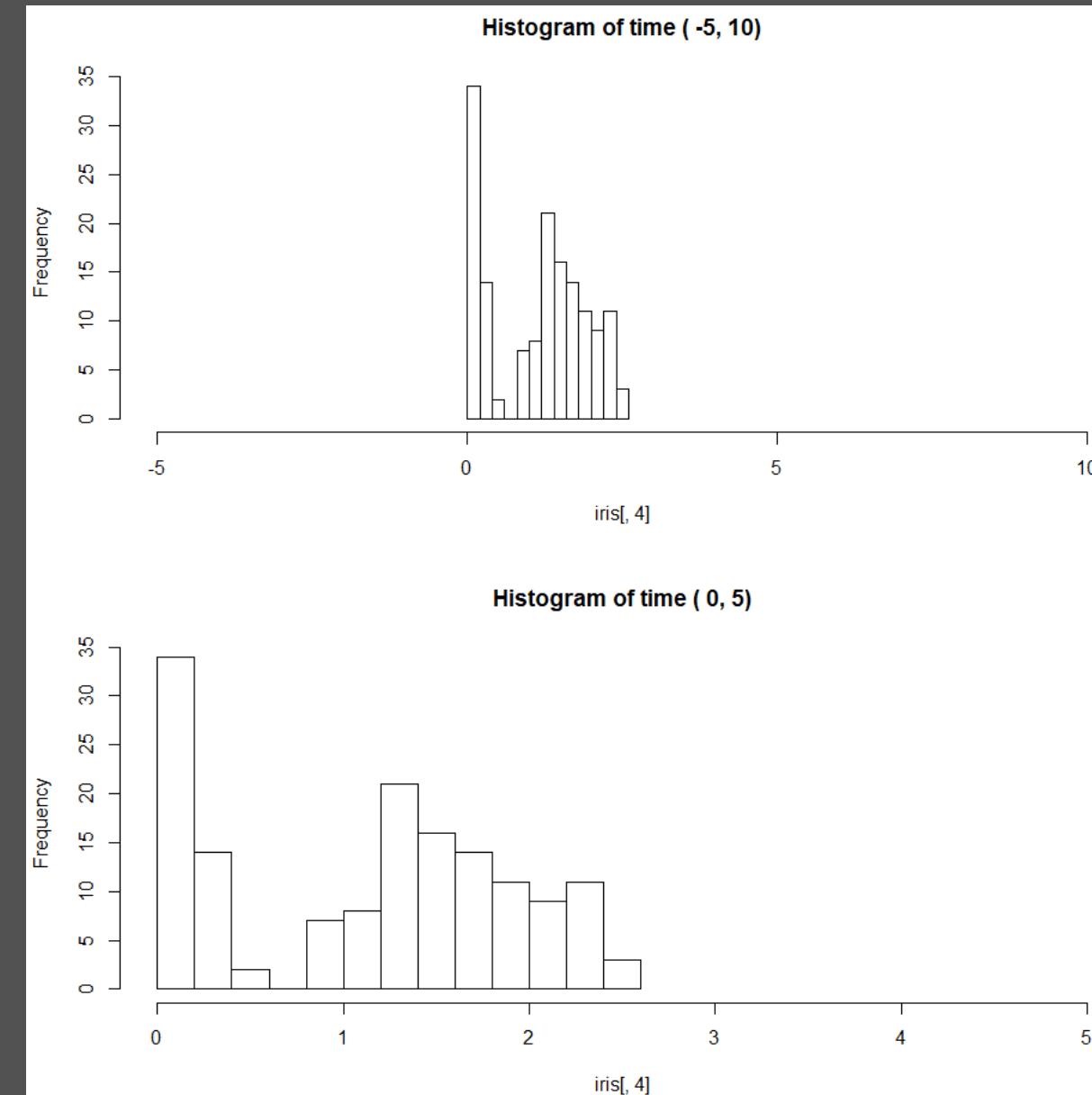
R 繪圖參數設定與劃線 - plot

1. 改變 y 軸與 x 軸的刻度

第三張圖則是上半部為 x 軸 $-5 \sim 10$ ，下半部 x 軸為 $0 \sim 5$ 。

```
windows()
par(mfrow = c(2,1))
hist(iris[,4], xlim = c(-5, 10), main = "")
title("Histogram of time (-5, 10)")
hist(iris[,4], xlim = c(0, 5), main = "")
title("Histogram of time (0, 5)")
```

```
> windows()
> par(mfrow = c(2,1))
> hist(iris[,4], xlim = c(-5, 10), main = "")
> title("Histogram of time (-5, 10)")
> hist(iris[,4], xlim = c(0, 5), main = "")
> title("Histogram of time (0, 5)")
> |
```

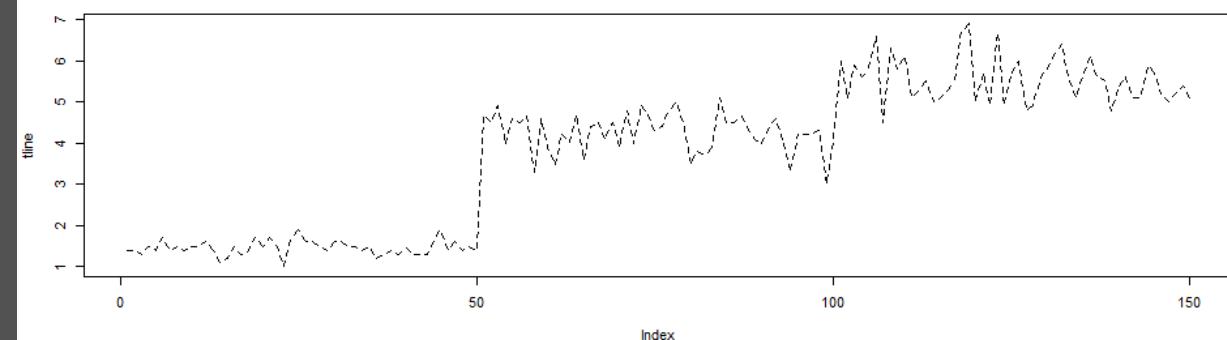
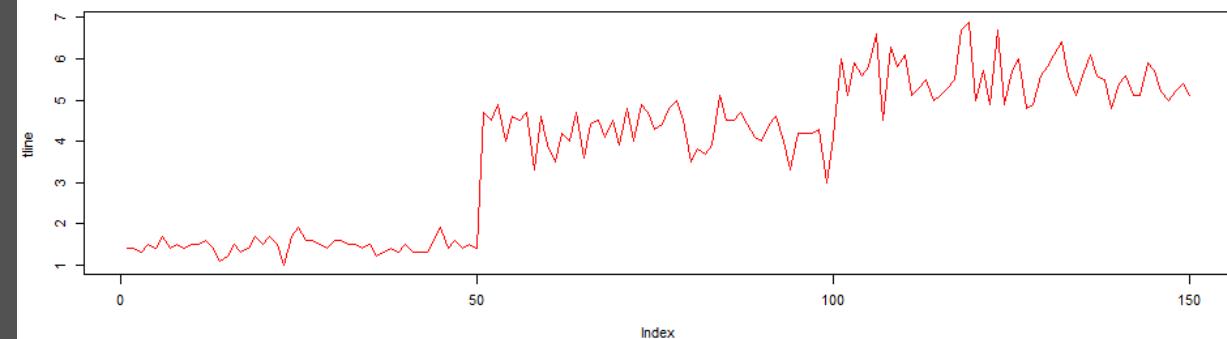
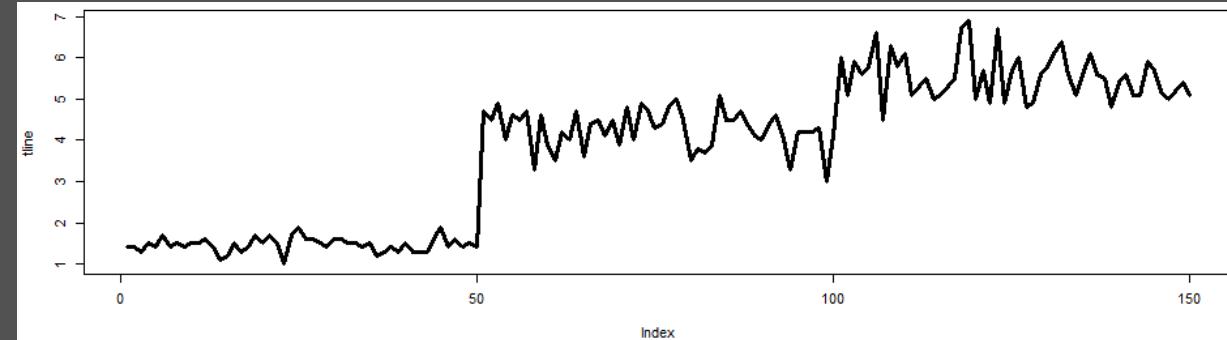


R 繪圖參數設定與劃線 - plot

2. 折線圖在線上的細節

- (1) lwd - 設定線的寬度
- (2) col - 設定線的顏色
- (3) lty - 設定線的型態

```
tline = iris[,3]
windows()
par(mfrow = c(3,1))
plot(tline, type = "l", lwd = 3)
plot(tline, type = "l", col = "red")
plot(tline, type = "l", lty = "dashed")
```



```
> tline = iris[,3]
> windows()
> par(mfrow = c(3,1))
> plot(tline, type = "l", lwd = 3)
> plot(tline, type = "l", col = "red")
> plot(tline, type = "l", lty = "dashed")
> |
```

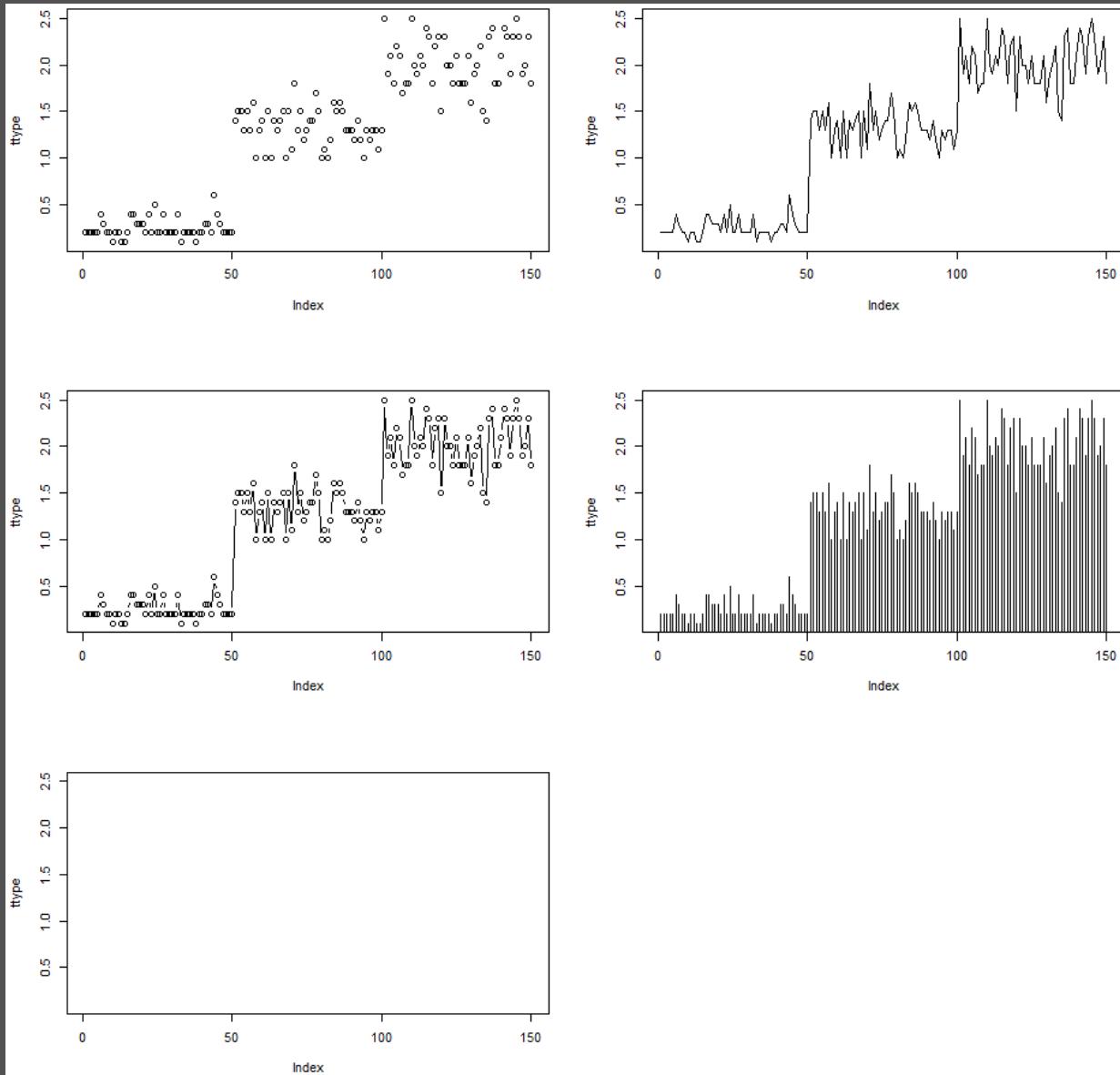
R 繪圖參數設定與劃線 - plot

3. plot 設定圖的類別

可以根據 type 產生不同的資料。

```
ttype = iris[,4]
windows()
par(mfrow = c(3,2))
# points
plot(ttype, type = "p")
# lines
plot( ttype, type = "l")
# both
plot( ttype, type = "b")
# histogram - like
plot( ttype, type = "h")
# none
plot( ttype, type = "n")
```

```
> ttype = iris[,4]
> windows()
> par(mfrow = c(3,2))
> # points
> plot(ttype, type = "p")
> # lines
> plot( ttype, type = "l")
> # both
> plot( ttype, type = "b")
> # histogram - like
> plot( ttype, type = "h")
> # none
> plot( ttype, type = "n")
> |
```



R 繪圖參數設定與劃線 - plot

4. 控制繪圖的線

abline 函數為控制繪圖的線，可以將資料進行迴歸後輸入。

- (1) abline - 函數為控制繪圖的線
- (2) col - 為控制顏色
- (3) h - 為根據 y 軸的水平線
- (4) v - 為根據 x 軸的垂直線
- (5) lwd - 寬度
- (6) a 與 b 之間是相互關連，a 為 y 軸起始，b 則為斜率

```
> windows()
> plot(iris$Sepal.Length, iris$Petal.Length,
+       xlim = c(-1, max(iris$Sepal.Length)),
+       ylim = c(-1, max(iris$Petal.Length)))
> # abline 函數為控制繪圖的線
> # 可以將資料進行迴歸後輸入
> abline(lm(iris$Petal.Length ~ iris$Sepal.Length), col = "black")
> # col 為控制顏色
> abline(h = 0, col = "grey")
> abline(v = 0, col = "grey")
> # h 為根據 y 軸的水平線
> abline(h = 2, col = "red", lty = 2)
> # v 為根據 x 軸的垂直線
> abline(v = 5.5, col = "blue", lty = 3)
> # lwd 寬度
> # a 與 b 之間是相互關連，a 為 y 軸起始，b 則為斜率
> abline(a = 1, b = 0.7, col = "green", lty = 4, lwd = 2)
```

windows()

```
plot(iris$Sepal.Length, iris$Petal.Length,
      xlim = c(-1, max(iris$Sepal.Length)),
      ylim = c(-1, max(iris$Petal.Length)))
```

abline 函數為控制繪圖的線

可以將資料進行迴歸後輸入

```
abline(lm(iris$Petal.Length ~ iris$Sepal.Length), col = "black")
```

col 為控制顏色

```
abline(h = 0, col = "grey")
```

```
abline(v = 0, col = "grey")
```

h 為根據 y 軸的水平線

```
abline(h = 2, col = "red", lty = 2)
```

v 為根據 x 軸的垂直線

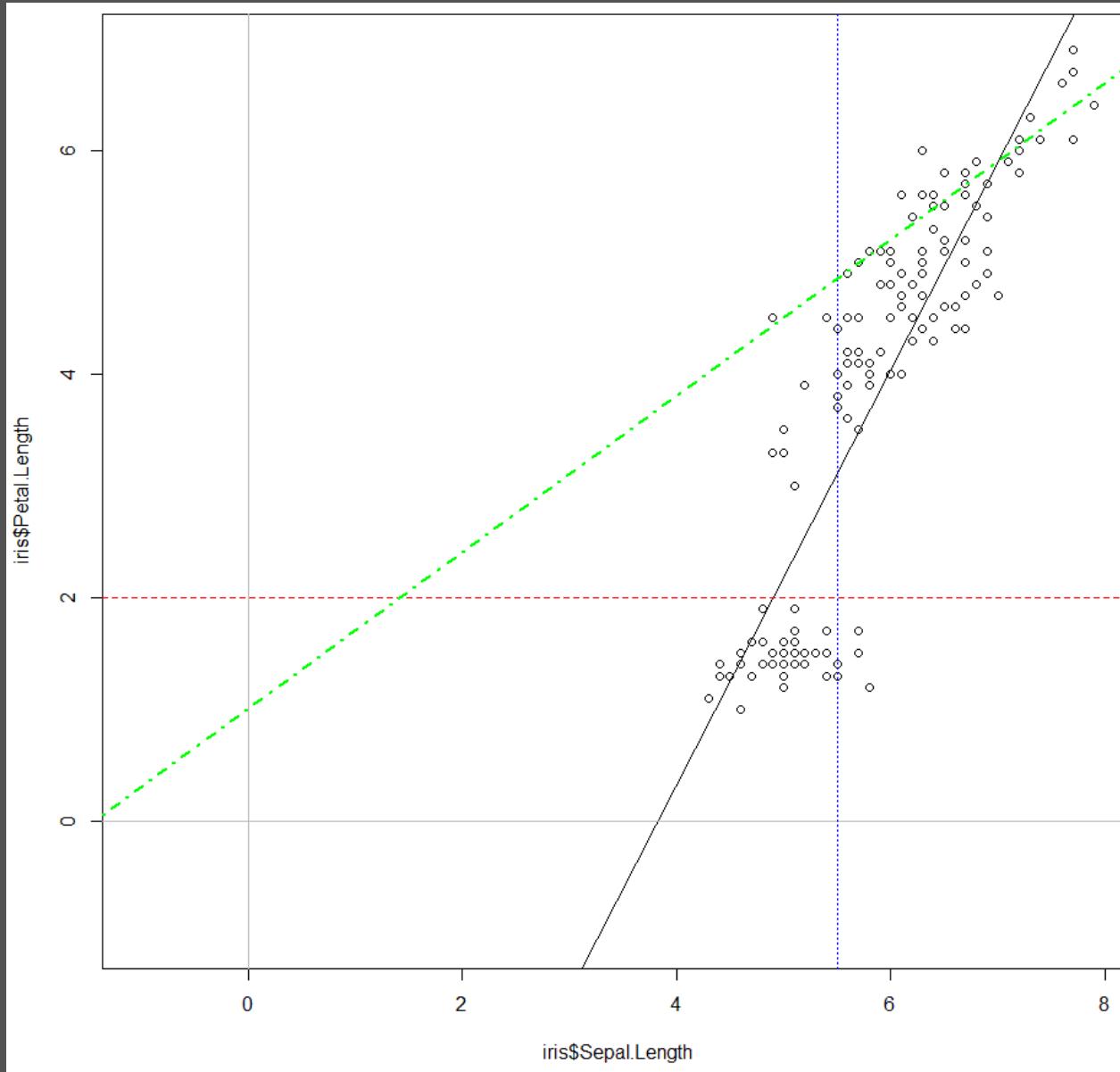
```
abline(v = 5.5, col = "blue", lty = 3)
```

lwd 寬度

a 與 b 之間是相互關連，a 為 y 軸起始，b 則為斜率

```
abline(a = 1, b = 0.7, col = "green", lty = 4, lwd = 2)
```

R 繪圖參數設定與劃線 - plot



R 繪圖參數設定與劃線 - plot

5. 散佈圖上點間的劃線

- (1) pch - 為控制散佈的點圖形
- (2) lines - 會將散佈的點連起來
- (3) order - 產生排序後數列資料

這裡要注意的事情在於 order 會產生指定資料的排序向量資料。

```
> tlidt1 = iris[, 1]
> tlidt2 = iris[, 3]
> windows()
> par(mfrow = c(2,1))
> # pch 為控制散佈的點圖形
> plot( tlidt1, tlidt2, pch = 14, col = "blue")
> # lines 會將散佈的點連起來
> lines( tlidt1, tlidt2)
> title("Line")
> # order 產生排序後數列資料
> sequence = order(tlidt1)
> plot( tlidt1, tlidt2, pch = 16, col = "blue")
> lines( tlidt1[sequence], tlidt2[sequence])
```

```
tlidt1 = iris[, 1]
tlidt2 = iris[, 3]
windows()
par(mfrow = c(2,1))
# pch 為控制散佈的點圖形
plot( tlidt1, tlidt2, pch = 14, col = "blue")
# lines 會將散佈的點連起來
lines( tlidt1, tlidt2)
title("Line")
# order 產生排序後數列資料
sequence = order(tlidt1)
plot( tlidt1, tlidt2, pch = 16, col = "blue")
lines( tlidt1[sequence], tlidt2[sequence])
```

R 繪圖參數設定與劃線 - plot

6. 指定點的劃線處裡

- (1) arrows - 產生箭號的直線
- (2) segments - 產生連結的直線

```
> asdtx = runif(12)
> asdty = rnorm(12)
> asdtx
[1] 0.97140110 0.79700670 0.05078871 0.29912886 0.84373687 0.50369670
[7] 0.98098230 0.38386720 0.17789458 0.50882285 0.19059442 0.90874910
> asdty
[1] 0.2301323 -0.8536572 2.5545825 0.1277021 -0.1817337 0.5664868
[7] -1.2473631 0.1287396 -1.6001798 -0.4431681 -0.5764739 0.9994687
> windows()
> plot( asdtx, asdty, main = "Arrows & Segments")
> arrows( asdtx[1], asdty[1],
+ asdtx[2], asdty[2],
+ col = "black", length = 0.2)
> segments(asdtx[3], asdty[3],
+ asdtx[4], asdty[4], col = "red")
> segments(asdtx[3:4], asdty[3:4],
+ asdtx[5:6], asdty[5:6], col = c("blue", "green"))
```

```
asdtx = runif(12)
asdty = rnorm(12)
asdtx
asdty
windows()
plot( asdtx, asdty, main = "Arrows & Segments")
arrows( asdtx[1], asdty[1],
asdtx[2], asdty[2],
col = "black", length = 0.2)
segments(asdtx[3], asdty[3],
asdtx[4], asdty[4], col = "red")
segments(asdtx[3:4], asdty[3:4],
asdtx[5:6], asdty[5:6], col = c("blue", "green"))
```

R 繪圖參數設定與劃線 - plot

7. 繪圖的標籤外框等設定

- (1) col - 設定點顏色
- (2) fg - 設定圖表外框顏色
- (3) col.axis - 設定 x 軸與 y 軸的字體顏色
- (4) col.main - 設定標題的字體顏色
- (5) col.lab - 設定 x 軸與 y 軸在標題的字體顏色
- (6) col.sub - 設定副標題的字體顏色
- (7) cex.lab - 設定 x 軸與 y 軸在標題的字體大小

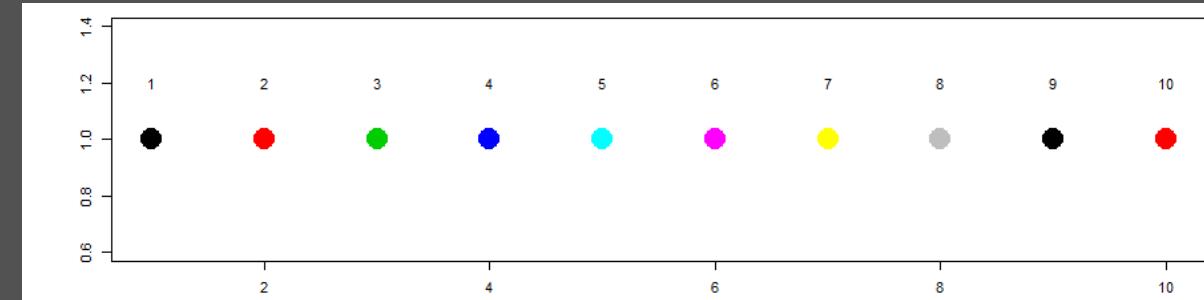
```
> windows()
> colr.p = par(mai=c(0.7, 0.7, 0.7, 0.7))
> (colr.m = matrix(c( 1, 2, 3, 1, 4, 5), 3, 2))
      [,1] [,2]
[1,]    1    1
[2,]    2    4
[3,]    3    5
> layout(colr.m)
> plot( 1:10, rep( 1, 10), pch = 20,
+       col = 1:10, cex = 5,
+       xlab = "", ylab = "")
> text( 1:10, rep(1.2, 10), labels = 1:10)
> plot( 1, col = 1:10, cex = 5,
+       xlab = "x", ylab = "y", type = "n",
+       main = "fg", fg = "blue")
> plot( 1, col = 1:10, cex = 5,
+       xlab = "x", ylab = "y", type = "n",
+       main = "col.axis(red) & col.main(blue)",
+       col.axis = "red",
+       col.main="blue")
> plot( 1, col = 1:10, cex = 5,
+       xlab = "x", ylab = "y", type = "n",
+       main = "col.lab(green)", col.lab = "green")
> plot( 1, col = 1:10, cex = 5,
+       xlab = "x", ylab = "y", type = "n")
> title( main = "My Title", col.main = "red",
+         sub = "My Sub-title", col.sub = "blue",
+         xlab = "My X label", ylab = "My Y label",
+         col.lab = "green", cex.lab = 2)
> par(colr.p)
> |
```

R 繪圖參數設定與劃線 - plot

```

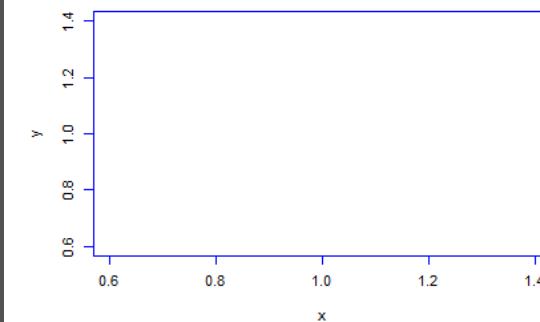
windows()
colr.p = par(mai=c(0.7, 0.7, 0.7, 0.7))
(colr.m = matrix(c( 1, 2, 3, 1, 4, 5), 3, 2))
layout(colr.m)
plot(1:10, rep( 1, 10), pch = 20,
      col = 1:10, cex = 5,
      xlab = "", ylab = "")
text( 1:10, rep(1.2, 10), labels = 1:10)
plot( 1, col = 1:10, cex = 5,
      xlab = "x", ylab = "y", type = "n",
      main = "fg", fg = "blue")
plot( 1, col = 1:10, cex = 5,
      xlab = "x", ylab = "y", type = "n",
      main = "col.axis(red) & col.main(blue)",
      col.axis = "red",
      col.main="blue")
plot( 1, col = 1:10, cex = 5,
      xlab = "x", ylab = "y", type = "n",
      main = "col.lab(green)", col.lab = "green")
plot( 1, col = 1:10, cex = 5,
      xlab = "x", ylab = "y", type = "n")
title( main = "My Title", col.main = "red",
       sub = "My Sub-title", col.sub = "blue",
       xlab = "My X label", ylab = "My Y label",
       col.lab = "green", cex.lab = 2)
par(colr.p)

```

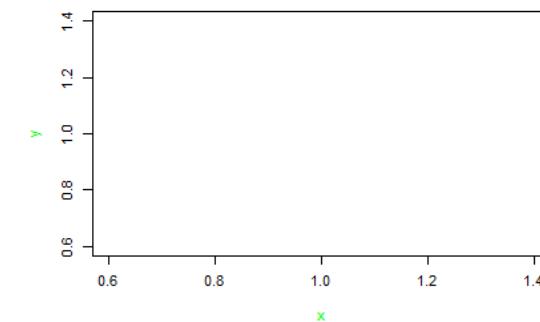


fg

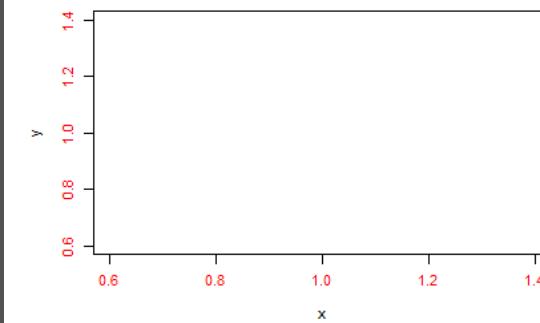
col.lab(green)



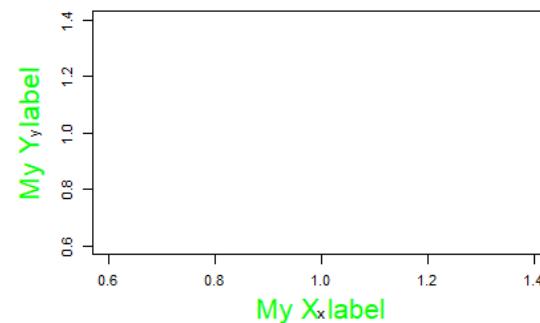
col.axis(red) & col.main(blue)



My Title



My Y_label

My Xlabel
My Sub-title

R 繪製熱圖與階層式集群的呈現

heatmap 會將資料轉變為熱圖的方式進行呈現，行列間可以建立預設為階層式的集群樹。

`heatmap {stats}`

R Documentation

Draw a Heat Map

Description

A heat map is a false color image (basically `image(t(x))`) with a dendrogram added to the left side and to the top. Typically, reordering of the rows and columns according to some set of values (row or column means) within the restrictions imposed by the dendrogram is carried out.

Usage

```
heatmap(x, Rowv = NULL, Colv = if(symm)"Rowv" else NULL,
        distfun = dist, hclustfun = hclust,
        reorderfun = function(d, w) reorder(d, w),
        add.expr, symm = FALSE, revC = identical(Colv, "Rowv"),
        scale = c("row", "column", "none"), na.rm = TRUE,
        margins = c(5, 5), ColSideColors, RowSideColors,
        cexRow = 0.2 + 1/log10(nr), cexCol = 0.2 + 1/log10(nc),
        labRow = NULL, labCol = NULL, main = NULL,
        xlab = NULL, ylab = NULL,
        keep.dendro = FALSE, verbose = getOption("verbose"), ...)
```

hclustfun

function used to compute the hierarchical clustering when `Rowv` or `Colv` are not dendrograms. Defaults to `hclust`. Should take as argument a result of `distfun` and return an object to which `as.dendrogram` can be applied.

R 繪製熱圖與階層式集群的呈現

在此利用 R 內部的 mtcars 資料。

head(mtcars)

class(mtcars)

NROW(mtcars)

rownames(mtcars)

colnames(mtcars)

```
> head(mtcars)
      mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4     21.0   6 160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag 21.0   6 160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710    22.8   4 108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive 21.4   6 258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8 360 175 3.15 3.440 17.02  0  0    3    2
Valiant       18.1   6 225 105 2.76 3.460 20.22  1  0    3    1
> class(mtcars)
[1] "data.frame"
> NROW(mtcars)
[1] 32
> rownames(mtcars)
[1] "Mazda RX4"           "Mazda RX4 Wag"        "Datsun 710"
[4] "Hornet 4 Drive"      "Hornet Sportabout"   "Valiant"
[7] "Duster 360"          "Merc 240D"          "Merc 230"
[10] "Merc 280"            "Merc 280C"          "Merc 450SE"
[13] "Merc 450SL"          "Merc 450SLC"         "Cadillac Fleetwood"
[16] "Lincoln Continental" "Chrysler Imperial"   "Fiat 128"
[19] "Honda Civic"         "Toyota Corolla"      "Toyota Corona"
[22] "Dodge Challenger"    "AMC Javelin"        "Camaro Z28"
[25] "Pontiac Firebird"    "Fiat X1-9"          "Porsche 914-2"
[28] "Lotus Europa"        "Ford Pantera L"      "Ferrari Dino"
[31] "Maserati Bora"       "Volvo 142E"          >
> colnames(mtcars)
[1] "mpg"   "cyl"   "disp"  "hp"    "drat"  "wt"    "qsec" "vs"    "am"
[10] "gear"  "carb" >
```

R 繪製熱圖與階層式集群的呈現

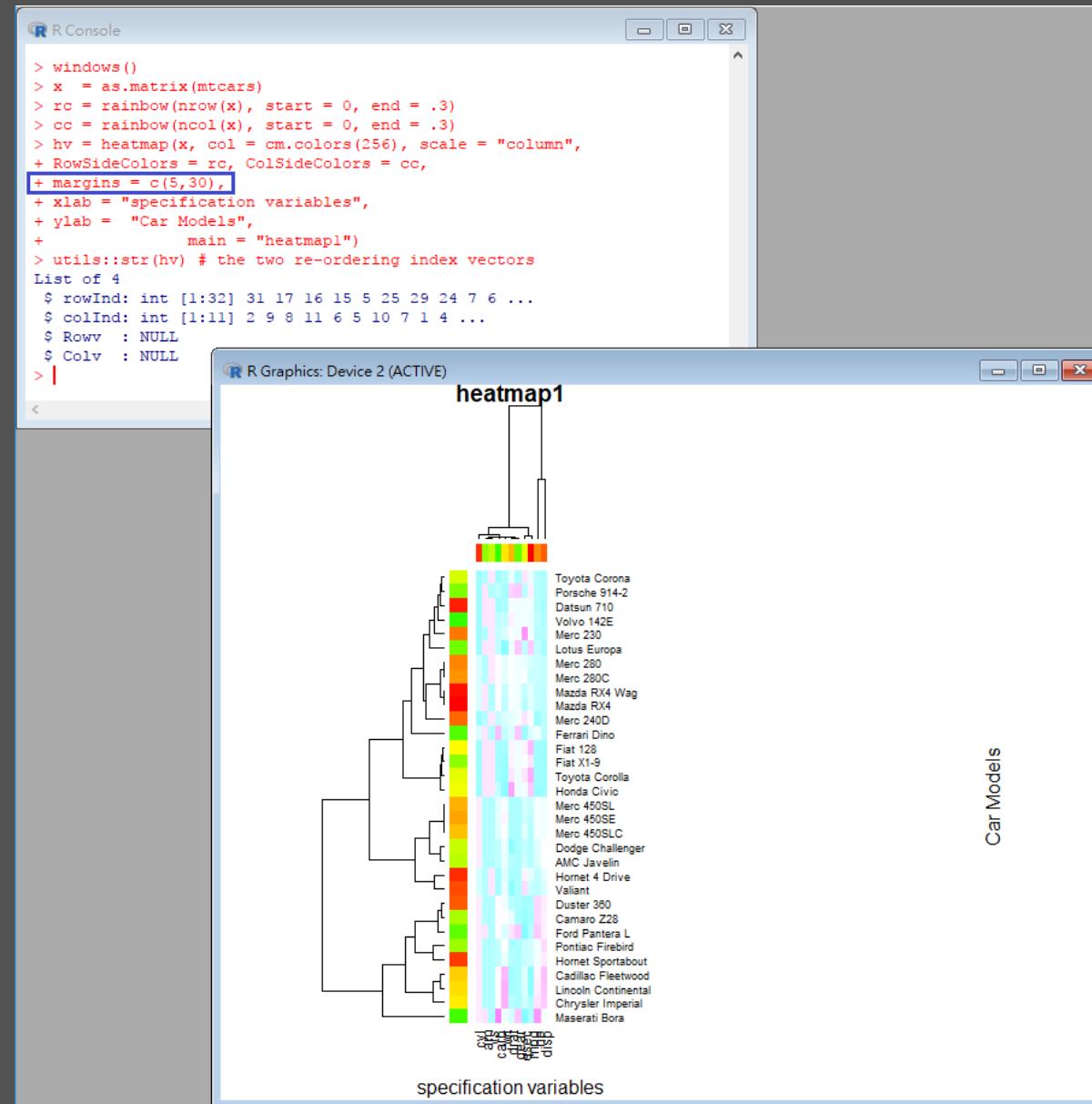
```
windows()
x = as.matrix(mtcars)
rc = rainbow(nrow(x), start = 0, end = .3)
cc = rainbow(ncol(x), start = 0, end = .3)
hv = heatmap(x, col = cm.colors(256), scale = "column",
             RowSideColors = rc, ColSideColors = cc,
             margins = c(5,10),
             xlab = "specification variables",
             ylab = "Car Models",
             main = "heatmap1")
utils::str(hv) # the two re-ordering index vectors
```

將 mtcars 轉成 Matrix 型態存入 x。rc、cc 物件，則是用 rainbows 函數則是根據 mtcars 的行列建立起連續的 RGB 顏色資料，之後存入 RowSideColors 與 ColSideColors 的參數，margin 為 heatmap 的比例，而 heatmap 的 col 參數，則是根據 cm.colors()，建立起256 的連續顏色。

```
> windows()
> x = as.matrix(mtcars)
> rc = rainbow(nrow(x), start = 0, end = .3)
> cc = rainbow(ncol(x), start = 0, end = .3)
> hv = heatmap(x, col = cm.colors(256), scale = "column",
+ RowSideColors = rc, ColSideColors = cc,
+ margins = c(5,10),
+ xlab = "specification variables",
+ ylab = "Car Models",
+           main = "heatmap1")
> utils::str(hv) # the two re-ordering index vectors
List of 4
$ rowInd: int [1:32] 31 17 16 15 5 25 29 24 7 6 ...
$ colInd: int [1:11] 2 9 8 11 6 5 10 7 1 4 ...
$ Rowv   : NULL
$ Colv   : NULL
> |
```

R 繪製熱圖與階層式集群的呈現

下圖將 margin 修改後的呈現，為影響 heatmap 繪圖的比例。



R 繪製熱圖與階層式集群的呈現

```

> head(rc)
[1] "#FF0000FF" "#FF0F00FF" "#FF1E00FF" "#FF2C00FF" "#FF3B00FF"
[6] "#FF4A00FF"
> NROW(rc)
[1] 32
> NROW(mtcars)
[1] 32
>
> head(cc)
[1] "#FF0000FF" "#FF2E00FF" "#FF5C00FF" "#FF8A00FF" "#FFB800FF"
[6] "#FFE500FF"
> NROW(cc)
[1] 11
> NCOL(mtcars)
[1] 11
>
> head(cm.colors(256))
[1] "#80FFFFFF" "#80FFFFFF" "#81FFFFFF" "#82FFFFFF" "#83FFFFFF"
[6] "#84FFFFFF"
> NROW(cm.colors(256))
[1] 256
> |

```

下圖為 rc、cc、cm.colors，所建立的 RGB 顏色資料。

```

head(rc)
NROW(rc)
NROW(mtcars)

```

```

head(cc)
NROW(cc)
NCOL(mtcars)

```

```

head(cm.colors(256))
NROW(cm.colors(256))

```

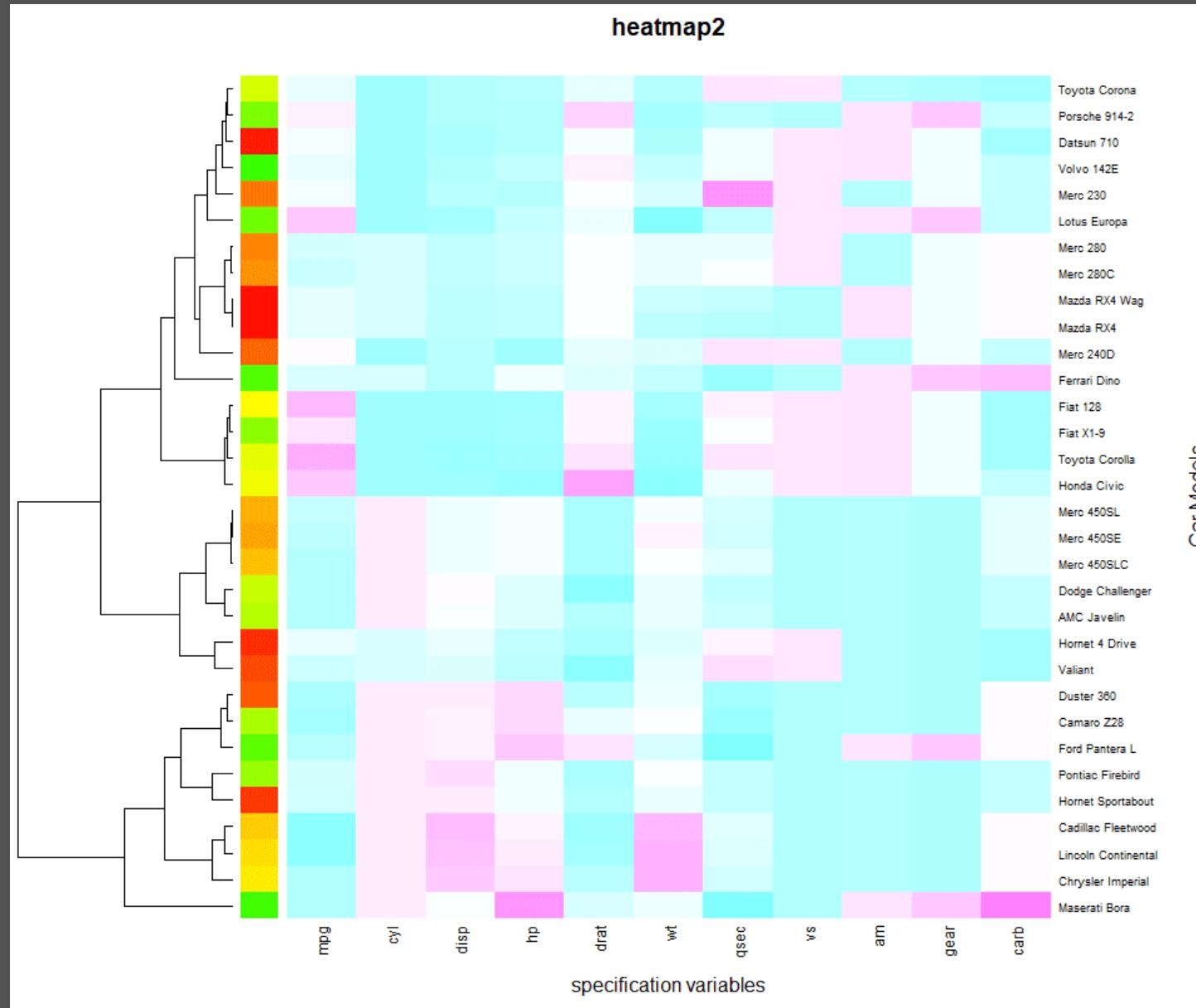
R 繪製熱圖與階層式集群的呈現

下圖為說明 Colv 參數的呈現，當為 NA 時則去掉上方的 Tree。

```
> windows()
> ## no column dendrogram (nor reordering) at all:
> heatmap(x, Colv = NA, col = cm.colors(256), scale = "column",
+ RowSideColors = rc, margins = c(5,10),
+ xlab = "specification variables", ylab = "Car Models",
+ main = "heatmap2")
> |
```

```
windows()
## no column dendrogram (nor reordering) at all:
heatmap(x, Colv = NA, col = cm.colors(256), scale = "column",
RowSideColors = rc, margins = c(5,10),
xlab = "specification variables", ylab = "Car Models",
main = "heatmap2")
```

R 繪製熱圖與階層式集群的呈現

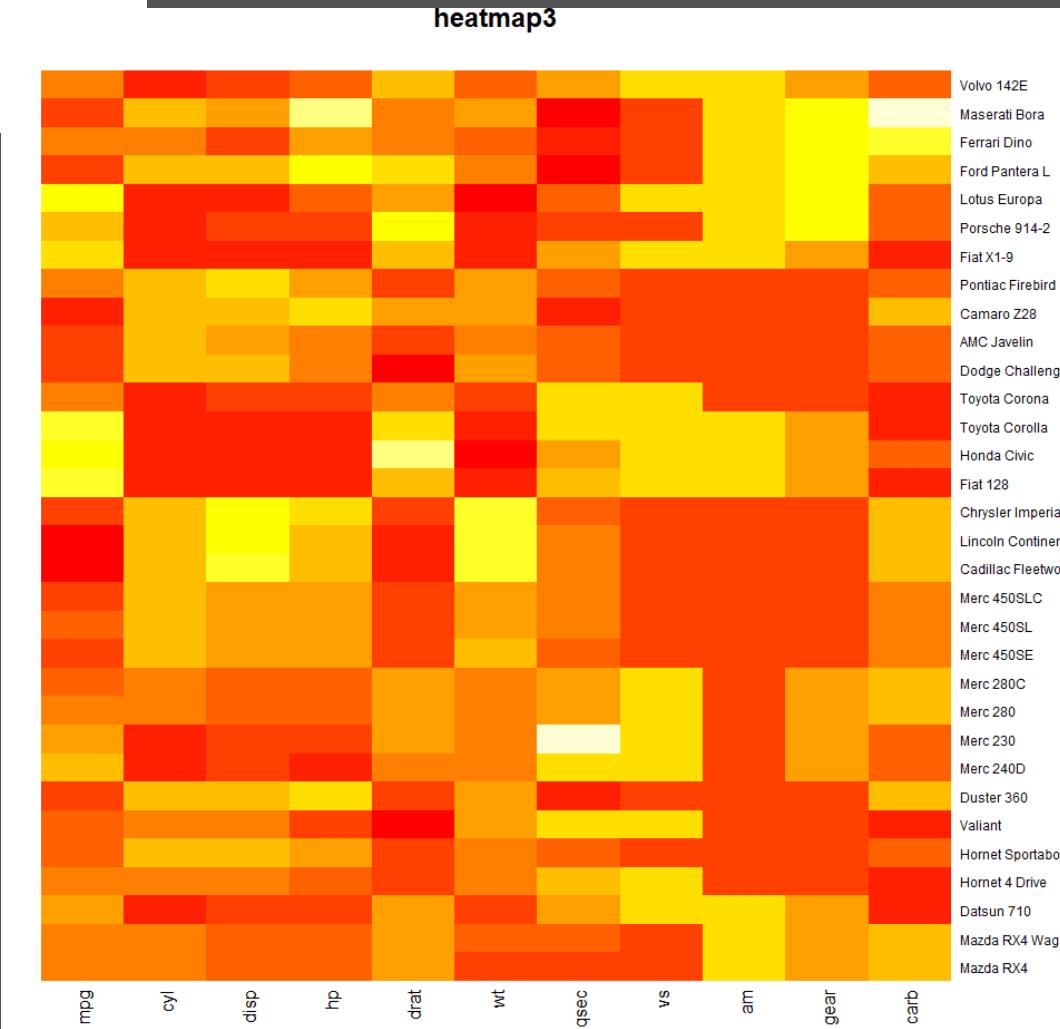


R 繪製熱圖與階層式集群的呈現

下圖為 Rowv 跟 Colv 的呈現，可以看到兩個 Tree 都消掉了。

```
> windows()
> heatmap(x, Rowv = NA, Colv = NA, scale = "column",
+ main = "heatmap3")
>
```

```
windows()
heatmap(x, Rowv = NA, Colv = NA, scale = "column",
main = "heatmap3")
```



R 繪製熱圖與階層式集群的呈現

下圖為 cm.colors()、rainbow() 函數文件。

Palettes {grDevices}

R Documentation

Color Palettes

Description

Create a vector of n contiguous colors.

Usage

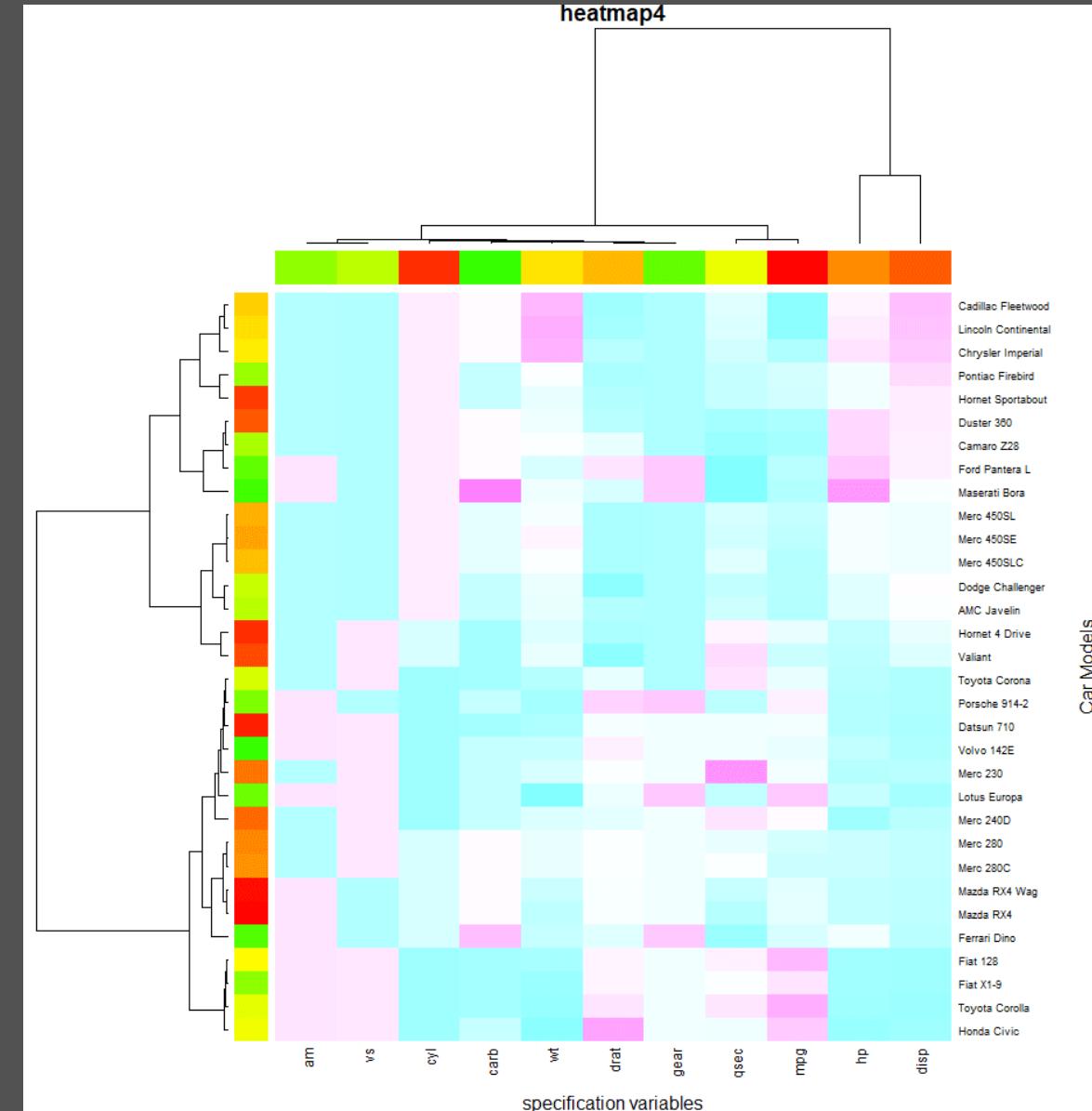
```
rainbow(n, s = 1, v = 1, start = 0, end = max(1, n - 1)/n, alpha = 1)
heat.colors(n, alpha = 1)
terrain.colors(n, alpha = 1)
topo.colors(n, alpha = 1)
cm.colors(n, alpha = 1)
```

R 繪製熱圖與階層式集群的呈現

在此修改範例將繪圖的 Tree，指定為階層式集群的華德法，距離為歐式距離。

```
> windows()
> hv2 = heatmap(x, col = cm.colors(256), scale = "column",
+ distfun = function(x) dist(x, method = "euclidean"),
+ hclustfun = function(x) hclust(x, method = "ward.D2"),
+ RowSideColors = rc, ColSideColors = cc,
+ margins = c(5,10),
+ xlab = "specification variables",
+ ylab = "Car Models",
+ main = "heatmap4")
> |
```

```
windows()
hv2 = heatmap(x, col = cm.colors(256), scale = "column",
  distfun = function(x) dist(x, method = "euclidean"),
  hclustfun = function(x) hclust(x, method = "ward.D2"),
  RowSideColors = rc, ColSideColors = cc,
  margins = c(5,10),
  xlab = "specification variables",
  ylab = "Car Models",
  main = "heatmap4")
```



R 集群分析的環形樹狀圖

1. 安裝 dendextend、circlize 套件

```
> install.packages("dendextend")
Installing package into 'D:/USERDATA/Documents/R/win-library/3.4'
(as 'lib' is unspecified)
also installing the dependencies 'modeltools', 'DEoptimR', 'mclust', 'flexmi
嘗試 URL 'https://cloud.r-project.org/bin/windows/contrib/3.4/modeltools_0.2
Content type 'application/zip' length 138817 bytes (135 KB)
downloaded 135 KB
```

```
> install.packages("circlize")
Installing package into 'D:/USERDATA/Documents/R/win-library/3.4'
(as 'lib' is unspecified)
also installing the dependencies 'GlobalOptions', 'shape'
嘗試 URL 'https://cloud.r-project.org/bin/windows/contrib/3.4/GlobalOptions_
Content type 'application/zip' length 75275 bytes (73 KB)
downloaded 73 KB
```

R 集群分析的環形樹狀圖

2. 匯入

```
> library("circlize")
> library("dendextend")

-----
Welcome to dendextend version 1.5.2
Type citation('dendextend') for how to cite the package.

Type browseVignettes(package = 'dendextend') for the package vignette.
The github page is: https://github.com/talgalili/dendextend/

Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/
Or contact: <tal.galili@gmail.com>

      To suppress this message use: suppressPackageStartupMessages(library(dendextend))
-----

Attaching package: 'dendextend'

The following object is masked from 'package:stats':
  cutree

The following object is masked from 'package:data.table':
  set
```

R 集群分析的環形樹狀圖

3. IRIS 與集群分析

在根據 iris 資料集作為集群分析的範例。

```
> iris = datasets::iris
> iris2 = iris[,-5]
> species_labels = iris[,5]
> library(colorspace)
> species_col = rev(rainbow_hcl(3))[as.numeric(species_labels)]
> d_iris = dist(iris2)
> hc_iris = hclust(d_iris, method = "complete")
> iris_species = rev(levels(iris[,5]))
>
> dend = as.dendrogram(hc_iris)
> dend = rotate(dend, 1:150)
>
> dend = color_branches(dend, k=3) #, groupLabels=iris_species)
>
> labels_colors(dend) =
+ rainbow_hcl(3)[sort_levels_values(
+ as.numeric(iris[,5])[order.dendrogram(dend)] +
+ )]
>
> labels(dend) = paste(as.character(iris[,5])[order.dendrogram(dend)],
+ ("",labels(dend),""),
+ sep = ""))
> dend = hang.dendrogram(dend,hang_height=0.1)
> dend = set(dend, "labels_cex", 0.5)
>
```

R 集群分析的 環形樹狀圖

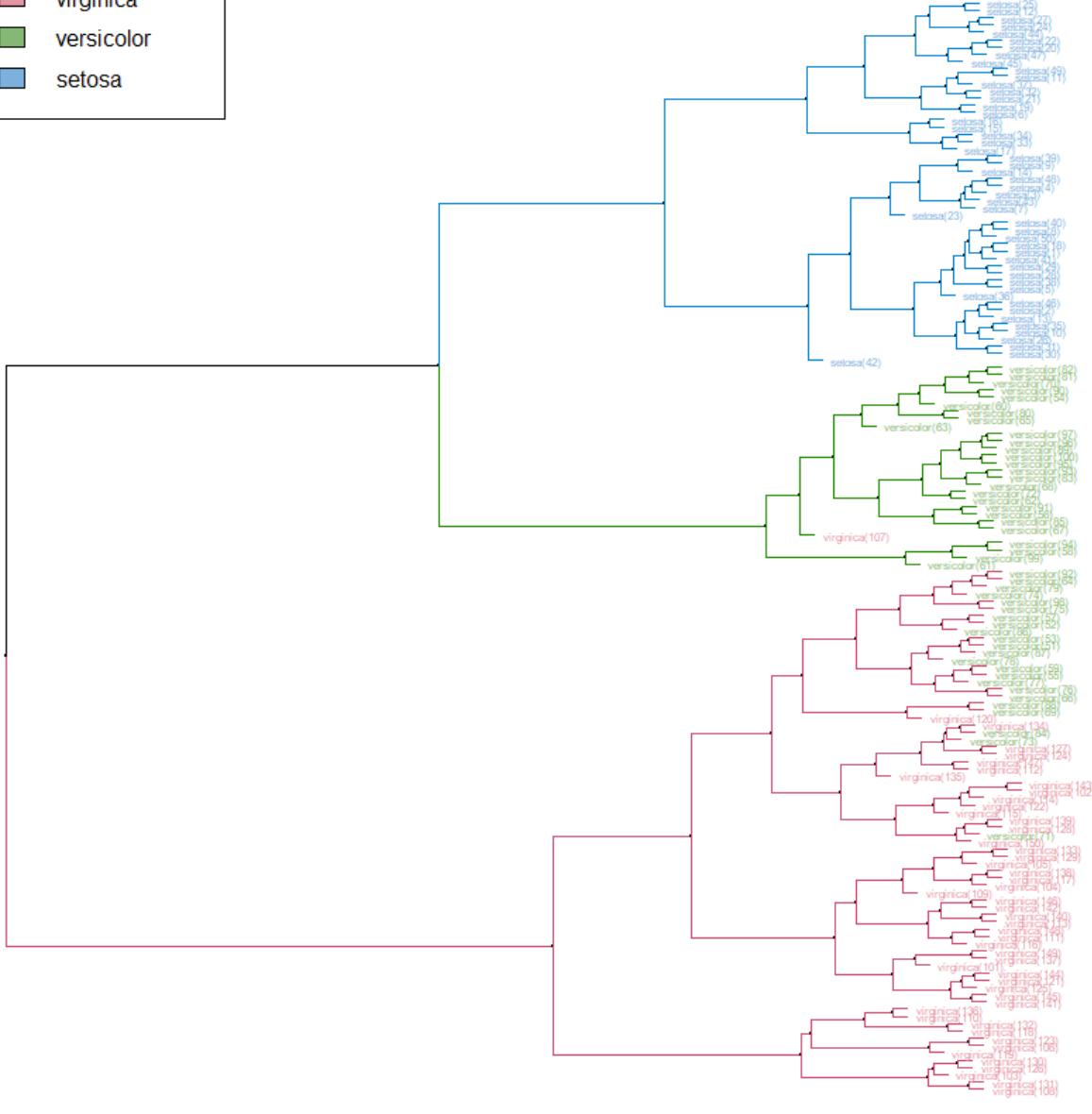
4. 進行樹狀的繪圖

```
> windows()
> par(mar = c(3,3,3,7))
> plot(dend,
+ main = "Clustered Iris data set",
+ (the labels give the true flower species)",
+ horiz = TRUE, nodePar = list(cex = .007))
> legend("topleft", legend = iris_species, fill = rainbow_hcl(3))
> |
```

Clustered Iris data set
(the labels give the true flower species)



7 6 5 4 3 2 1 0

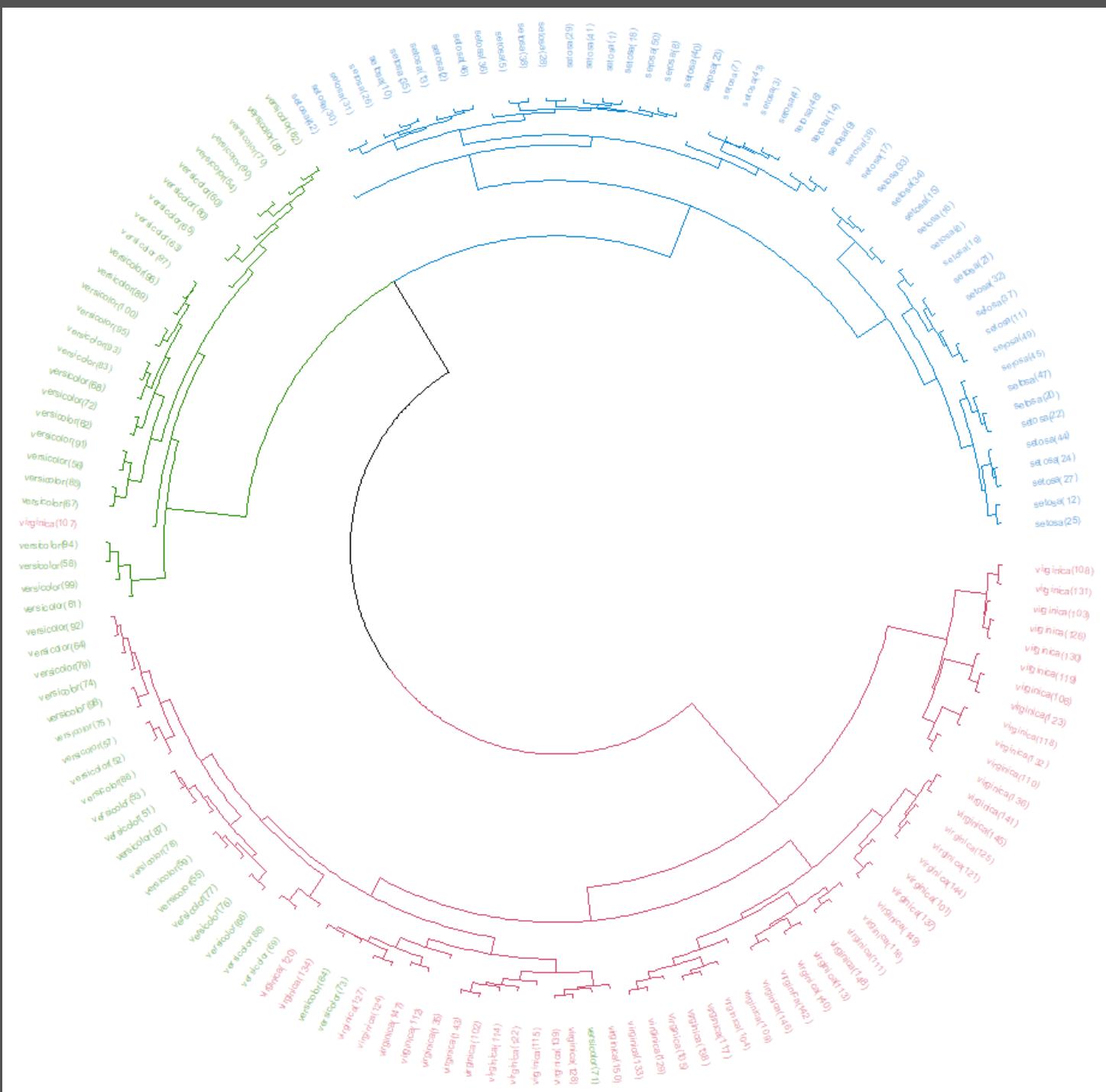


R 集群分析的 環形樹狀圖

105

5. 環狀繪圖

```
> windows()  
> par(mar = rep(0,4))  
> circlize_dendrogram(dend)  
> |
```



R 集群分析的環形樹狀圖

```

# install
install.packages("circlize")
install.packages("dendextend")
# library
library("circlize")
library("dendextend")
iris = datasets::iris
iris2 = iris[,-5]
species_labels = iris[,5]
library(colorspace)
species_col = rev(rainbow_hcl(3))[as.numeric(species_labels)]
d_iris = dist(iris2)
hc_iris = hclust(d_iris, method = "complete")
iris_species = rev(levels(iris[,5]))
dend = as.dendrogram(hc_iris)
dend = rotate(dend, 1:150)

dend = color_branches(dend, k=3) #, groupLabels=iris_species)

labels_colors(dend) =
rainbow_hcl(3)[sort_levels_values(
as.numeric(iris[,5])[order.dendrogram(dend)] )
]

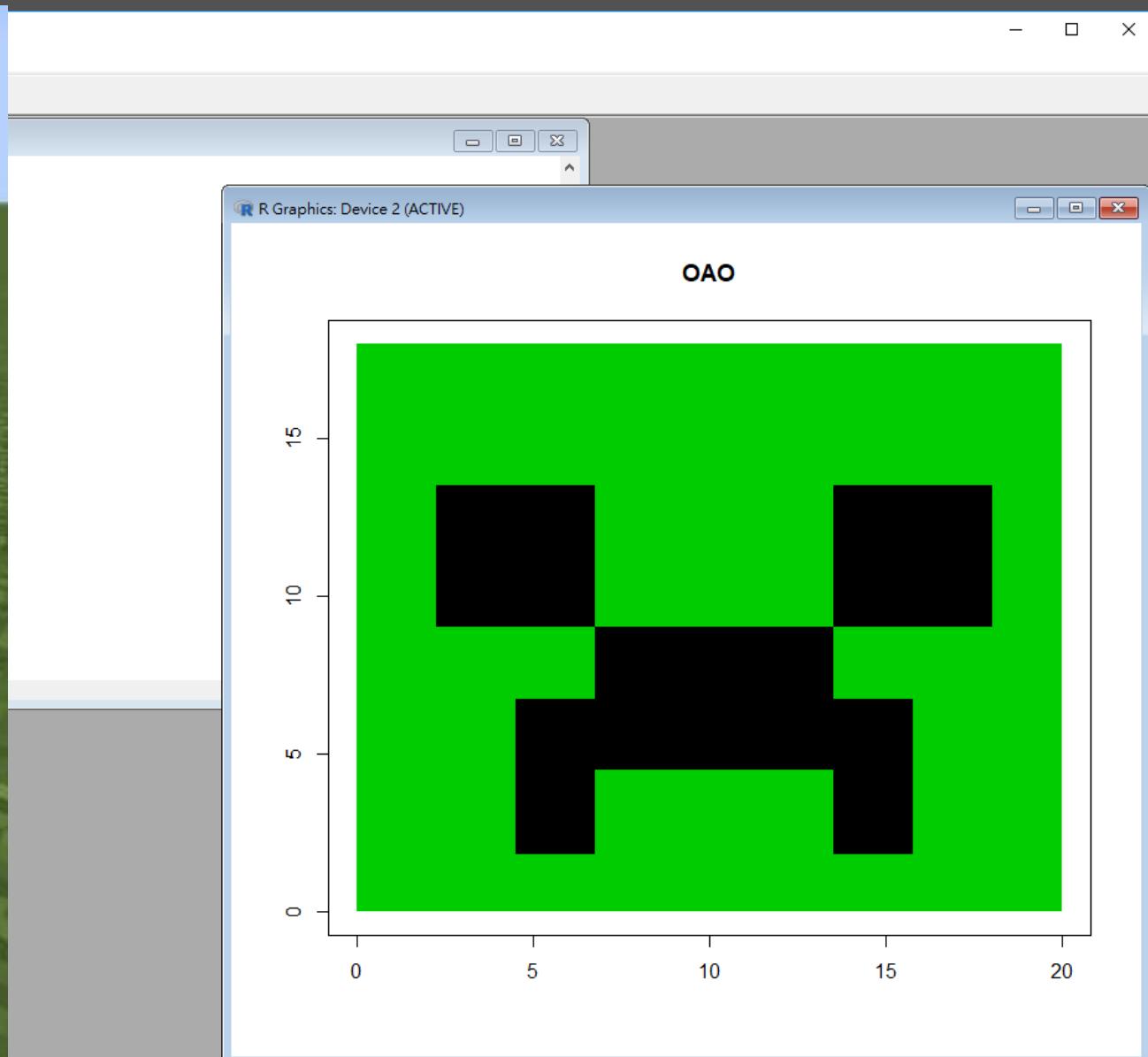
labels(dend) = paste(as.character(iris[,5])[order.dendrogram(dend)],
"(,labels(dend),)",
sep = "")
dend = hang.dendrogram(dend,hang_height=0.1)
dend = set(dend, "labels_cex", 0.5)

windows()
par(mar = rep(0,4))
circlize_dendrogram(dend)

windows()
par(mar = c(3,3,3,7))
plot(dend,
main = "Clustered Iris data set",
(the labels give the true flower species)",
horiz = TRUE, nodePar = list(cex = .007))
legend("topleft", legend = iris_species, fill = rainbow_hcl(3))

```

R 方形與圓形繪圖 - plot



R 方形與圓形繪圖 - plot

建立一個多圖合併。

```
> windows()
> bcp.par = par(mai=c(0.7, 0.7, 0.7, 0.7))
> (bcp.mat = matrix(c( 1, 2, 1, 3), 2, 2))
     [,1] [,2]
[1,]    1    1
[2,]    2    3
> layout(bcp.mat)
```

第一張是苦力怕圖，使用的函數為 rect，其實是根據兩個座標來畫方形，一個是左下座標跟右上座標。

```
> plot( 0, xlim = c( 0,20), ylim = c( 0, 18),
+ type = "n", xlab = "", ylab = "",
+ main = "OAO")
> rect( 0, 0, 20, 18, border = 0, lwd = 4 , col = 3)
> rect( 18 * 3 / 8,
+ 18 * 2/8,
+ 18 * 6/8,
+ 18 * 4/8, border = 0, lwd = 4 , col = 1)
> rect( 18 * 1 / 8,
+ 18 * 4/8,
+ 18 * 3/8,
+ 18 * 6/8, border = 0, lwd = 4 , col = 1)
> rect( 18 * 6 / 8,
+ 18 * 4/8,
+ 18 * 8/8,
+ 18 * 6/8, border = 0, lwd = 4 , col = 1)
> rect( 18 * 2 / 8,
+ 18 * 0.8/8,
+ 18 * 3/8,
+ 18 * 3/8, border = 0, lwd = 4 , col = 1)
> rect( 18 * 6 / 8,
+ 18 * 0.8/8,
+ 18 * 7/8,
+ 18 * 3/8, border = 0, lwd = 4 , col = 1)
>
```

R 方形與圓形繪圖 - plot

建立一個 0~6 的向量資料，逐一塞進 rect 函數。

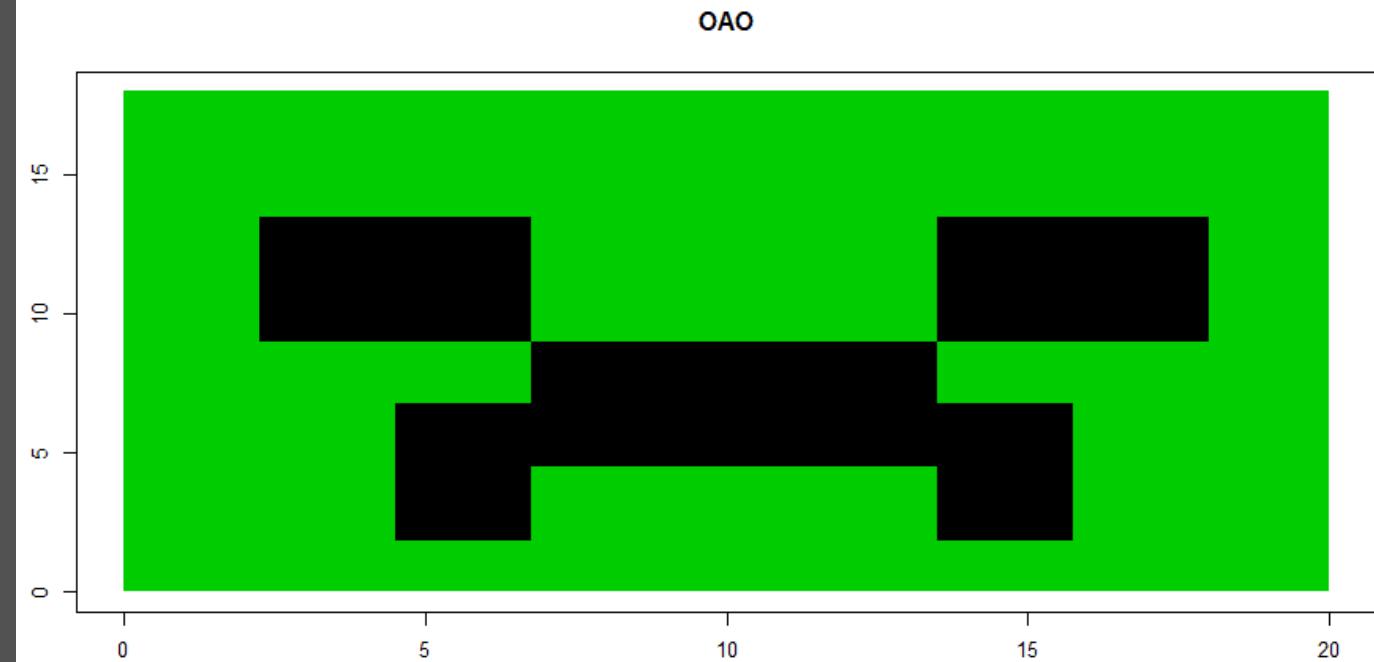
```
> plot( 0, xlim = c( 0,15), ylim = c( 0, 16),
+ type = "n", xlab = "", ylab = "",
+ main = "Rectangles")
> rect( 0.5, 0, 12.5, 12.5)
> n = 0:6
> rect( 1 + n, 1 + n, 2 + 2 * n, 2 + 2 * n, col = rainbow(4),
+ border = n+1, lwd = 4)
>
```

symbols 函數給定 x 與 y， circles 控制圓的大小， bg 則是控制顏色， xlim 與 ylim 控制圖的兩軸。

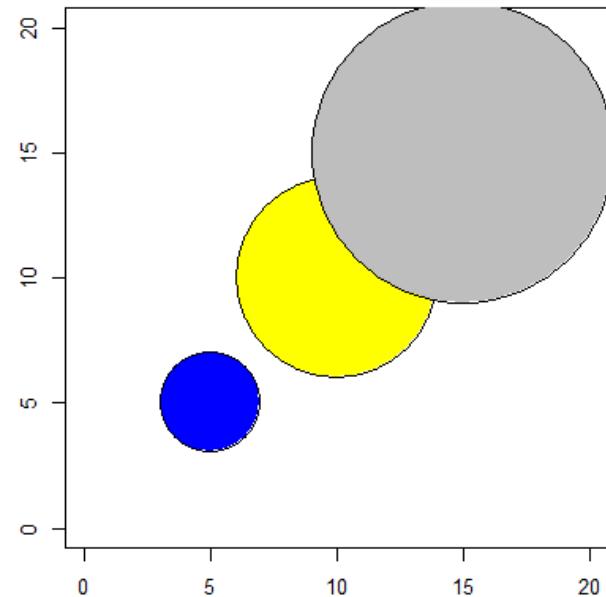
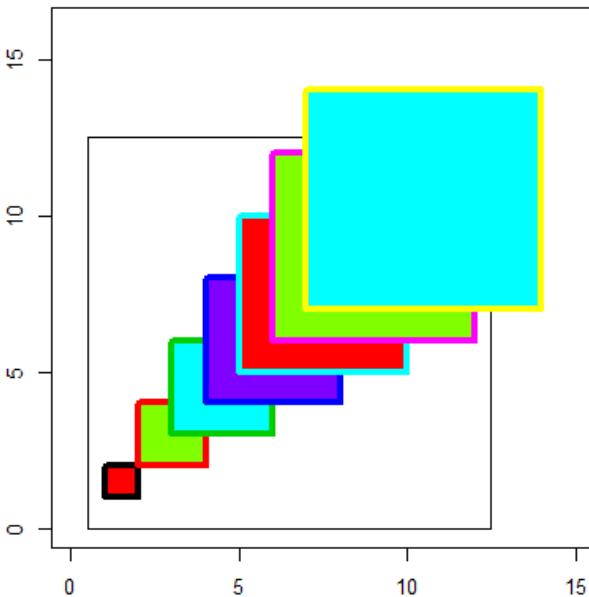
```
> symbols( x = c(5, 10, 15), y = c(5, 10, 15),
+ circles = c( 1, 2, 3),
+ xlim = c(0, 20), ylim = c(0, 20),
+ bg = c("blue", "yellow", "gray"),
+ xlab = "", ylab = "")
>
> par(bcp.par)
```

R 方形與圓形 繪圖 - plot

110



Rectangles



R 方形與圓形繪圖 - plot

```

windows()
bcp.par = par(mai=c(0.7, 0.7, 0.7, 0.7))
(bcp.mat = matrix(c( 1, 2, 1, 3), 2, 2))
layout(bcp.mat)
plot( 0, xlim = c( 0,20), ylim = c( 0, 18),
      type = "n", xlab = "", ylab = "",
      main = "OAO")
rect( 0, 0, 20, 18, border = 0, lwd = 4 , col = 3)
rect( 18 * 3 / 8,
      18 * 2/8,
      18 * 6/8,
      18 * 4/8, border = 0, lwd = 4 , col = 1)
rect( 18 * 1 / 8,
      18 * 4/8,
      18 * 3/8,
      18 * 6/8, border = 0, lwd = 4 , col = 1)
rect( 18 * 6 / 8,
      18 * 4/8,
      18 * 8/8,
      18 * 6/8, border = 0, lwd = 4 , col = 1)
rect( 18 * 2 / 8,
      18 * 0.8/8,
      18 * 3/8,
      18 * 3/8, border = 0, lwd = 4 , col = 1)
rect( 18 * 6 / 8,
      18 * 0.8/8,
      18 * 7/8,
      18 * 3/8, border = 0, lwd = 4 , col = 1)

plot( 0, xlim = c( 0,15), ylim = c( 0, 16),
      type = "n", xlab = "", ylab = "",
      main = "Rectangles")
rect( 0.5, 0, 12.5, 12.5)
n = 0:6
rect( 1 + n, 1 + n, 2 + 2 * n, 2 + 2 * n, col = rainbow(4),
      border = n+1, lwd = 4)

symbols( x = c(5, 10, 15), y = c(5, 10, 15),
         circles = c( 1, 2, 3),
         xlim = c(0, 20), ylim = c(0, 20),
         bg = c("blue", "yellow", "gray"),
         xlab = "", ylab = "")

par(bcp.par)

```