R plot 1

- R 語言內建許多繪圖工具函數，這些函數可以顯示各種統計繪圖並且自建一些全新的圖形。可以先參考以下 R 語言的繪圖示範

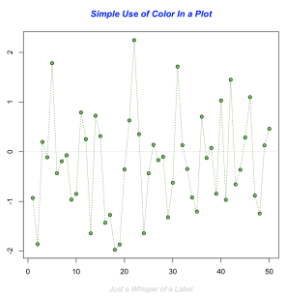
> demo(graphics)

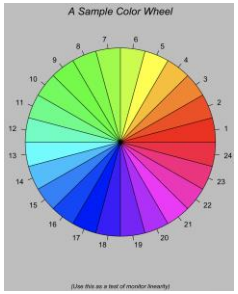
> demo(image)

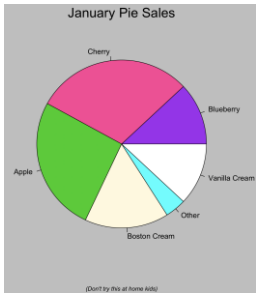
- R 語言的繪圖指令可以分成了三個基本的類型
  - 高階繪圖(High-level Plotting Functions):建立一個新的圖形，它可以包括座標軸及標題等
  - 低階繪圖(Low-level Plotting Functions):會在一個已經存在的圖形上加上其它的圖形元素，如額外的點及線等等
  - 互動式繪圖(Interactive Graphics Functions):允許互動式地用其他設備(如滑鼠)在一個已經存在的圖形上加上圖形資訊

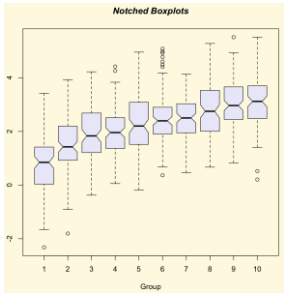
Demo(graphic)

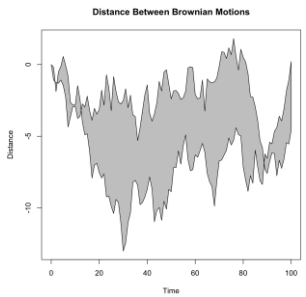
Here is some code which illustrates some of the differences between R and S graphics capabilities.
Note that colors are generally specified by a character string name (taken from the X11 rgb.txt file) and that line textures are given similarly.
The parameter "**bg**" sets the background parameter for the plot and there is also an "**fg**" parameter which sets the foreground color.

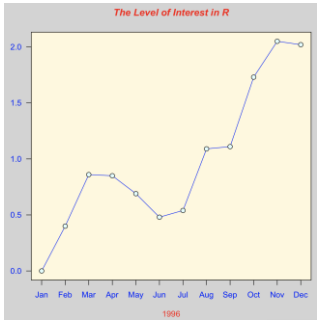


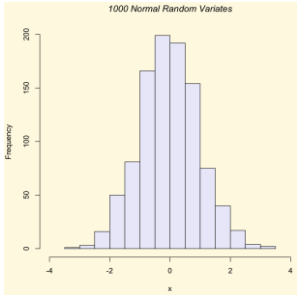


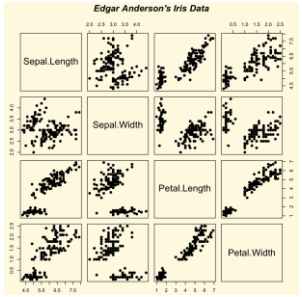


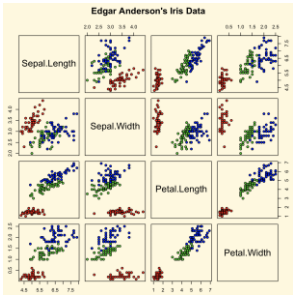


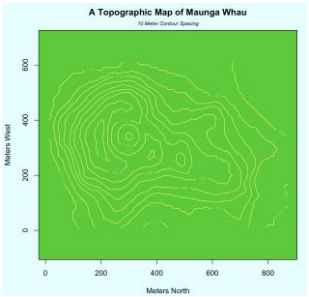


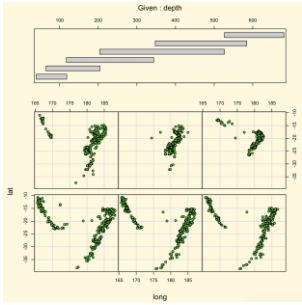






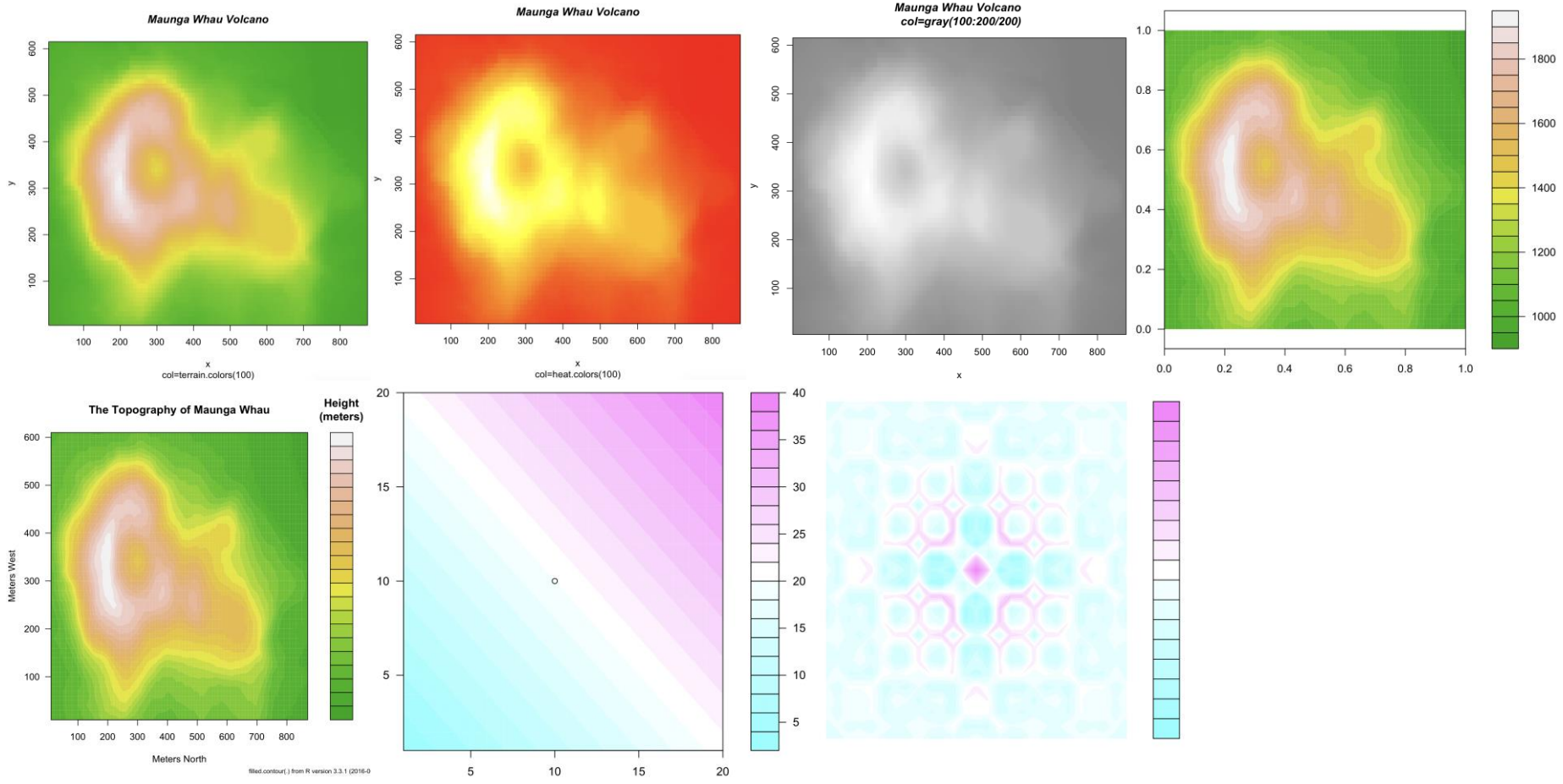






Demo(image)

Copyright (C) 1997-2009 The R Core Team



高階繪圖
(High-level Plotting Functions)

- 常用的高階繪圖函數為

plot(y)	以序號為橫坐標(x軸)、y為縱坐標(y軸)來繪圖
plot(x,y)	以x(x-軸)和y(y-軸)來繪圖
pie(y)	繪製餅形圖
boxplot(y)	繪製盒形圖
stem(y)	繪製莖葉圖(Stem-and-leaf Plot)
dotchart(y)	繪製點圖
hist(y)	繪製直方圖
barplot(y)	繪製條形圖
contour(x, y, z)	繪製等高線圖

- **plot() 函數引數的設定值如下:**

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

其中:

- x　表示為x座標位置
- y　表示為y座標位置
- type 表示為設定繪圖在(x,y)之顯示方式
- type="p" 表示為畫點
- type="l" 表示為畫線,
- type="b" 表示為畫點同時在點與點之間畫線連結
- type="s" 表示為階梯函數(Step Function)，為左連續函數
- type="S" 表示為階梯函數，為右連續函數
- type="o" 表示為畫線同時穿過畫點
- type="h" 表示為從點到x-橫軸畫垂直線
- type="n" 表示為不畫任何點與線，但容許畫坐標軸且建立坐標系統， 用於後面用低階圖形函數作圖.
- bty 表示為設定圖形座標軸外框(Box)的類型，其選項共有{ "o", "l", "7", "c", "u", "]" }
- pch 表示畫點時，設定pch=k，k是一個介於1至25之間的正整數，顯示一個相對應於k之特定符號，其預設值為1
- lty 表示畫線時，設定線條類型
- lty=1 表示為實線(Solid Line)
- lty=2 表示為虛線
- lwd 表示畫線時，設定線條寬度
- 　　lwd=1 表示為預設值
- 　　lwd=k 表示為線條寬度的倍數

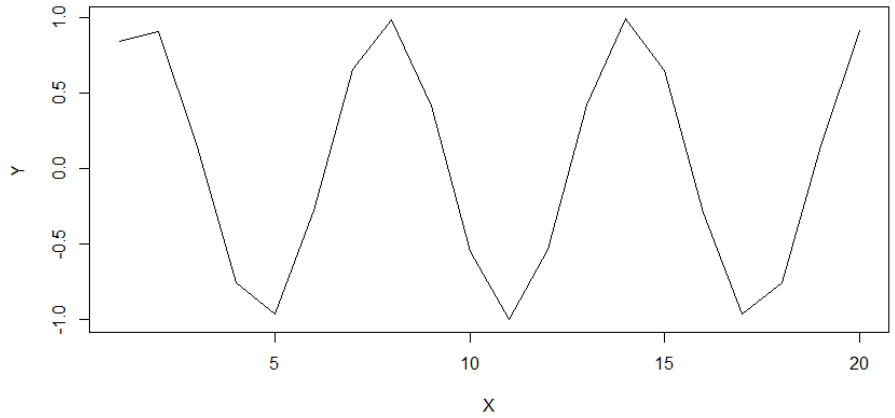
- col 表示為設定點、線的顏色。預設值為black，設定顏色值(Value) 是調色板的數值，或者顏色之文字名稱。
- bg 表示為設定圖形背景顏色，預設值為白色，例如 bg="white"
- xlim 表示為設定座標軸(x)範圍之上下界, 如xlim=c(1, 10)
- ylim 表示為設定座標軸(y)範圍之上下界, 如ylim=c(1, 10)
- log 表示為設定座標軸是否取對數值
- log="x" 表示為對x座標軸取對數值
- log="y" 表示為對y座標軸取對數值
- log="xy" 表示為對x與y座標軸都取對數值
- main 表示為設定圖形主標題(Main Title)，定義的文字放在圖形的上方，例如main="Title"
- sub 表示為設定圖形次標題(Subtitle)，定義的文字放在圖形的下方，例如sub="Subtitle"
- xlab 表示為設定x座標軸標記，例如xlab="X"
- ylab 表示為設定y座標軸標記，例如ylab="Y"
- ann 表示是否畫出自動設定之主標題及座標軸標記，ann=TRUE時會畫出自動設定之主標題及座標軸標記
- axes 表示是否畫出自動設定之座標軸與座標軸外框，axes=TRUE時會 畫出自動設定之座標軸與座標軸外框

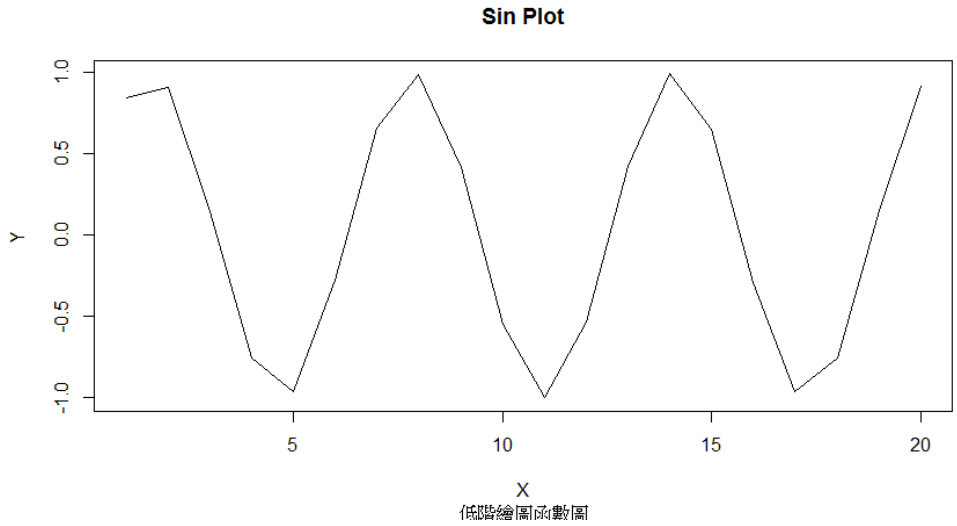
```
> y <- sin(1:20)
> plot(y, type="l",main="Sin Plot", xlab="X",ylab="Y")
```

低階繪圖
(Low-level Plotting Functions)

points(x,y)	在現有的圖形上加一個點
lines(x,y)	在現有的圖形上加一條線
text(x,y,labels=z.vec,...)	在(x,y)座標點標出由labels 所設定之對應z.vec之數值型或文字型向量
abline(a,b)	在現有的圖形上加畫一條截距為a和斜率為b直線
abline(h=y)	畫出在Y=y平行於x座標軸之水平線
abline(v=x)	畫出在X=x垂直於y座標軸之垂直線
polygon(x,y,...)	畫出以(x,y)座標點為頂點之多邊形(Polygon)，可以用col=引數指定一個特定色彩來填滿多邊形之內部
legend(x,y,leg.vec,...)	在現有的圖形的指定(x,y)座標位置繪製圖例(Legend)，圖例的說明文字由向量leg.vec 表示
title(main,sub)	main="My Main Title"設定圖形主標題，定義的文字放在圖形的上方, sub="My Subtitle"設定圖形次標題，定義的文字放在圖形的下方
mtext(text,side=3,line=1)	在現有的圖形之邊緣，加上文字

```
> y <- sin(1:20)
> plot(y, type="l", xlab="X",ylab="Y")
> title(main="Sin Plot",sub="低階繪圖函數圖")
```





互動式繪圖 (Interactive Graphics Functions)

- R 語言同時提供了允許使用者直接用滑鼠在一個圖形上取得(Extract)和增加(Add)資訊的函數，其中最為常用的是
  - **locator()函數**
  - **identify()函數**

- locator()函數的功能是供使用者用滑鼠左鍵點選當前圖形上的特定位置

  locator(n,type)

其中:

- n　表示為指定要點選幾個座標點，若不指定則預設n=512
- type 允許在被選擇的點上畫圖並且有高階繪圖函數一樣的效果；預設情況下不能畫圖。locator()函數使用列表物件包含元素x和y的方式傳回所選中點的位置資訊

```
> plot(2, 2)
> pts <- locator(n = 3)
> pts
```

- identify()函數的功能是容許使用者將定義的標籤(Label)，利用滑鼠左鍵放置在滑鼠遊標的點選處

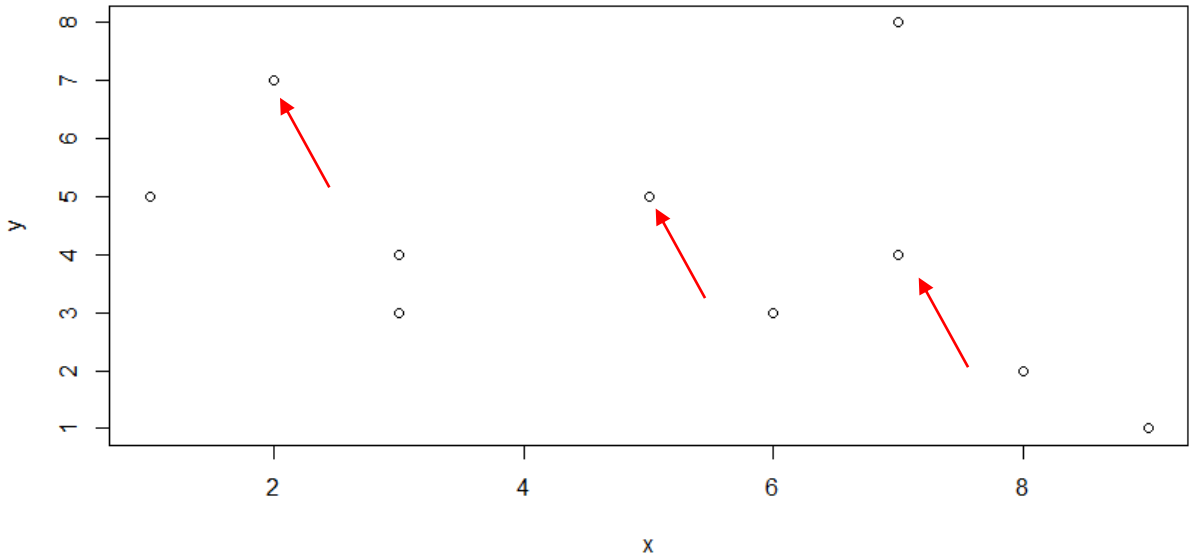
identify(x,y,labels)

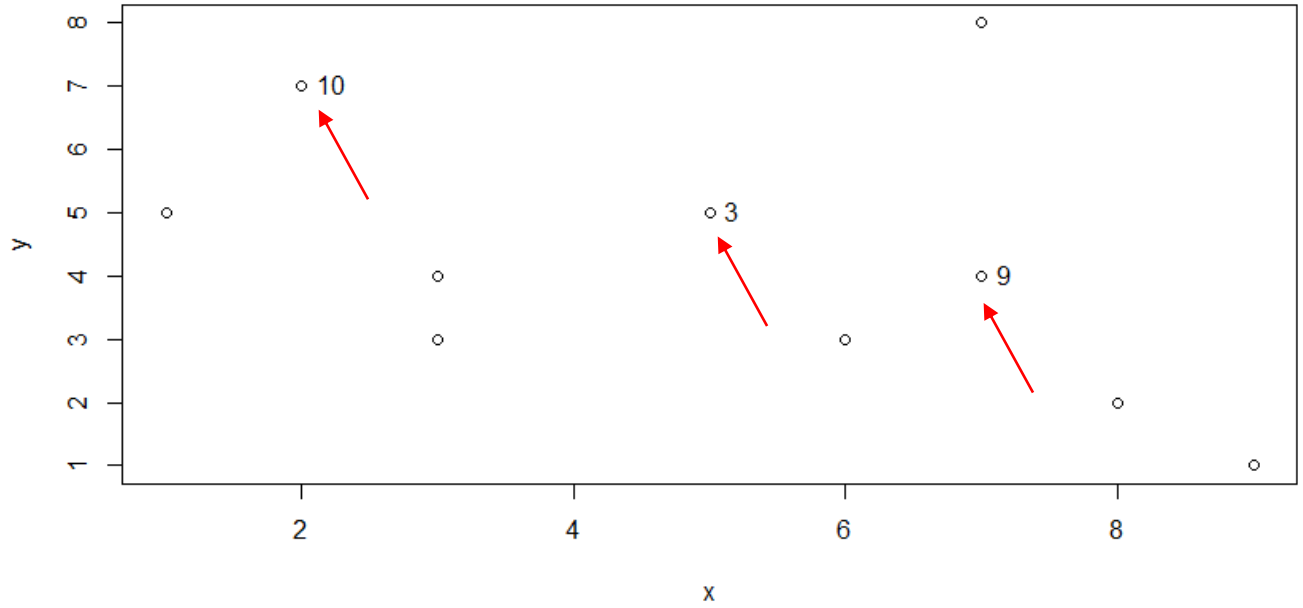
其中:

- x　　x座標位置
- y　　y座標位置
- labels　在labels未設定時，預設值為顯示點的序號，選擇完成後點選滑鼠右鍵結束選擇的動作;當使用引數labels="My Labels"可於滑鼠左鍵點選處顯示使用者定義的標籤"My Labels"，並點選滑鼠右鍵結束選擇的動作

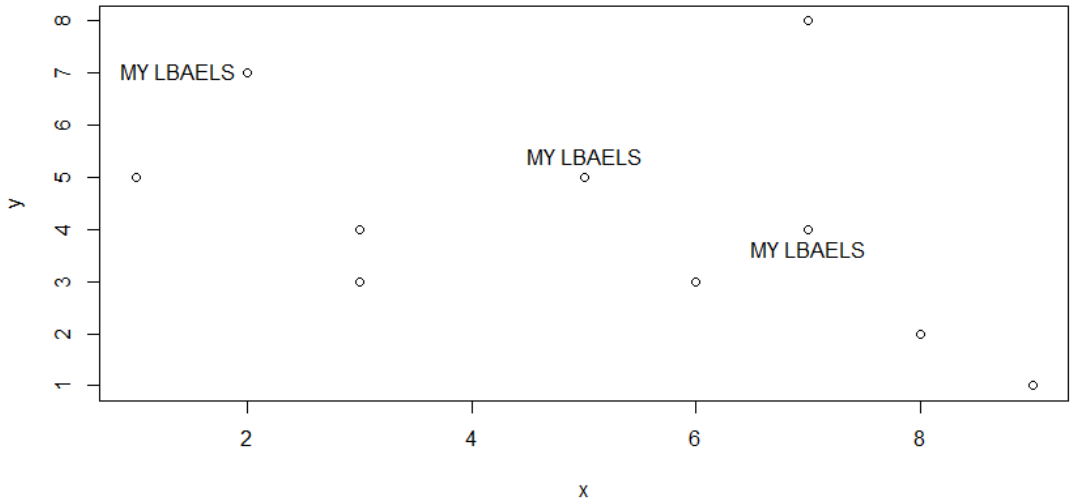
```
> x <- c(1, 3, 5, 7, 8, 9, 3, 6, 7, 2)
> y <- c(5, 3, 5, 8, 2, 1, 4, 3, 4, 7)
> plot(x, y)
> sel <- identify(x, y)
```

- identify()函數在選擇的過程中，會將選擇的資料點旁邊標上使用者定義的標籤，而這些標籤的位置的序號會在選擇完成後，傳回給sel變數
- 使用者就可以靠sel變數取得選擇的資料點
- (按 ESC 離開)


y
x
1
2
3
4
5
6
7
8
2
4
6
8


10
3
9
y
x
1
2
3
4
5
6
7
8
2
4
6
8

```
> x <- c(1, 3, 5, 7, 8, 9, 3, 6, 7, 2)
> y <- c(5, 3, 5, 8, 2, 1, 4, 3, 4, 7)
> plot(x, y)
> sel <- identify(x, y,"MY LBAELS")
```




```
> x.sel <- x[sel]
> y.sel <- y[sel]
```

```
> x.sel
> y.sel
```

Global Environment

Values	
sel	int [1:3] 3 9 10
x	num [1:10] 1 3 5 7 8 9 3 6 7 2
x.sel	num [1:3] 5 7 2
y	num [1:10] 5 3 5 8 2 1 4 3 4 7
y.sel	num [1:3] 5 4 7

圖形參數

- R 語言提供許多圖形參數(Graphics Parameters)可以控制圖形的顏色、文字對齊等
  - 直接呼叫 par() 函數可以得到目前所有參數的設定值

```
> par()
```

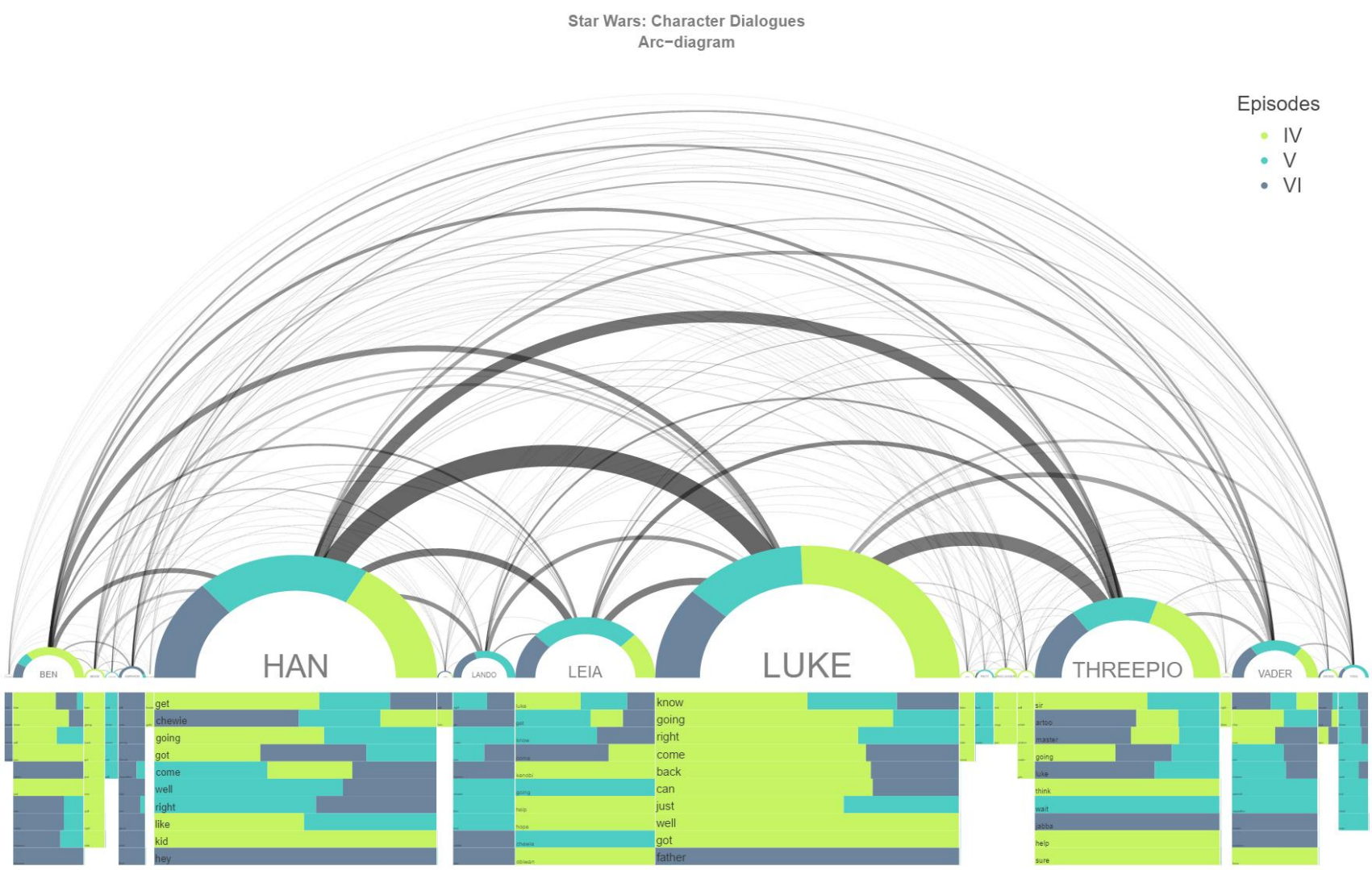
- 透過par()函數可更改及設定圖形參數

  par(par.name=par.value)

其中:

- par.name　表示為圖形參數名稱，它可以用於par()或某些(高階或低階)繪圖函數中當引數
- par.value　表示為設定給par.name圖形參數之設定值

```
> par(col=4, lty=4)
> plot(y, type="l", xlab="X",ylab="Y")
```

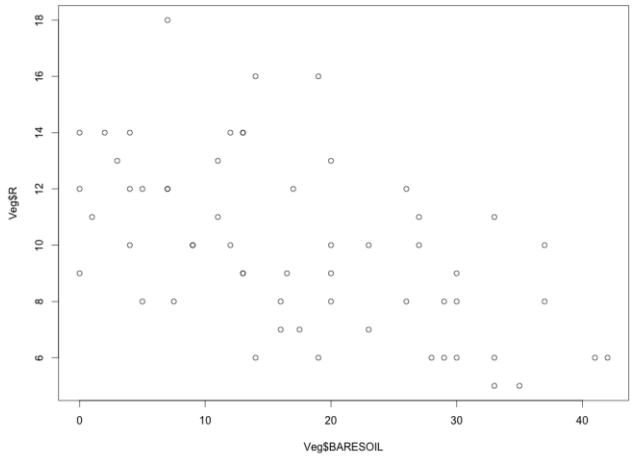


Description: Perform a statistical text analysis on the Star Wars scripts from Episodes IV, V, and VI, in order to obtain an arc-diagram representation (more or less equivalent to the one in Similar Diversity) with the most talkative characters of the Star Wars trilogy.

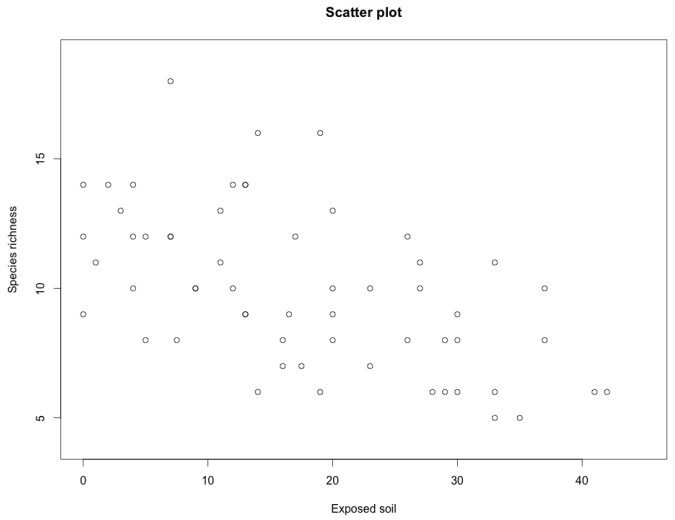
Plot example:

- #黃石國家公園（Yellowstone National Park）與 National Bison Range 所觀測到的草原生態資料，研究的目的在於觀察這裡的生物多樣性是否有隨著時間而改變。
- Veg <- read.table(file = " http://120.126.1.1/Vegetation2.txt", header = TRUE)

- #畫出物種豐富度（Veg$R）與外露泥土（Veg$BARESOIL）的關係分布圖，
- plot(Veg$BARESOIL, Veg$R)

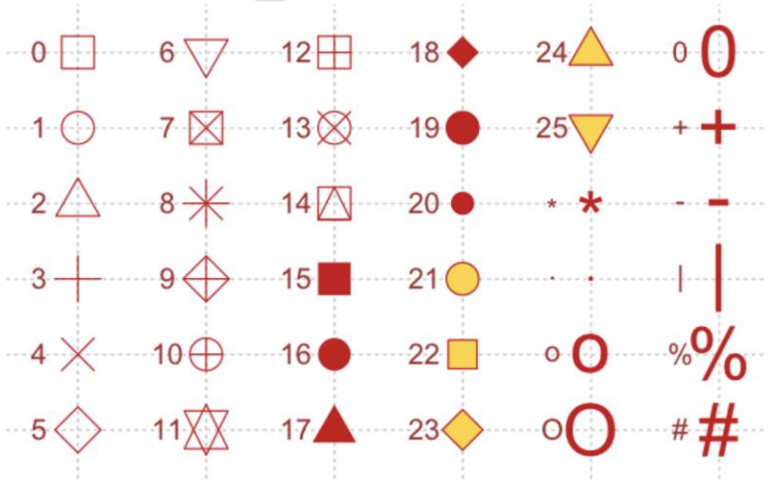


- plot (x = Veg$BARESOIL, y = Veg$R,
  xlab = *"Exposed soil"*,
  ylab = *"Species richness"*,
  main = *"Scatter plot"*,
  xlim = c(0, 45), ylim = c(4, 19))



資料點的符號

- plot 預設會使用圓圈來標示資料點，我們可以透過 **pch** 參數來使用不同的資料點符號，可用的符號如下：




```
plot(x = Veg$BARESOIL, y = Veg$R,
xlab = "Exposed soil",
ylab = "Species richness",
main = "Scatter plot",
xlim = c(0, 45), ylim = c(4, 19), pch = 16)
```

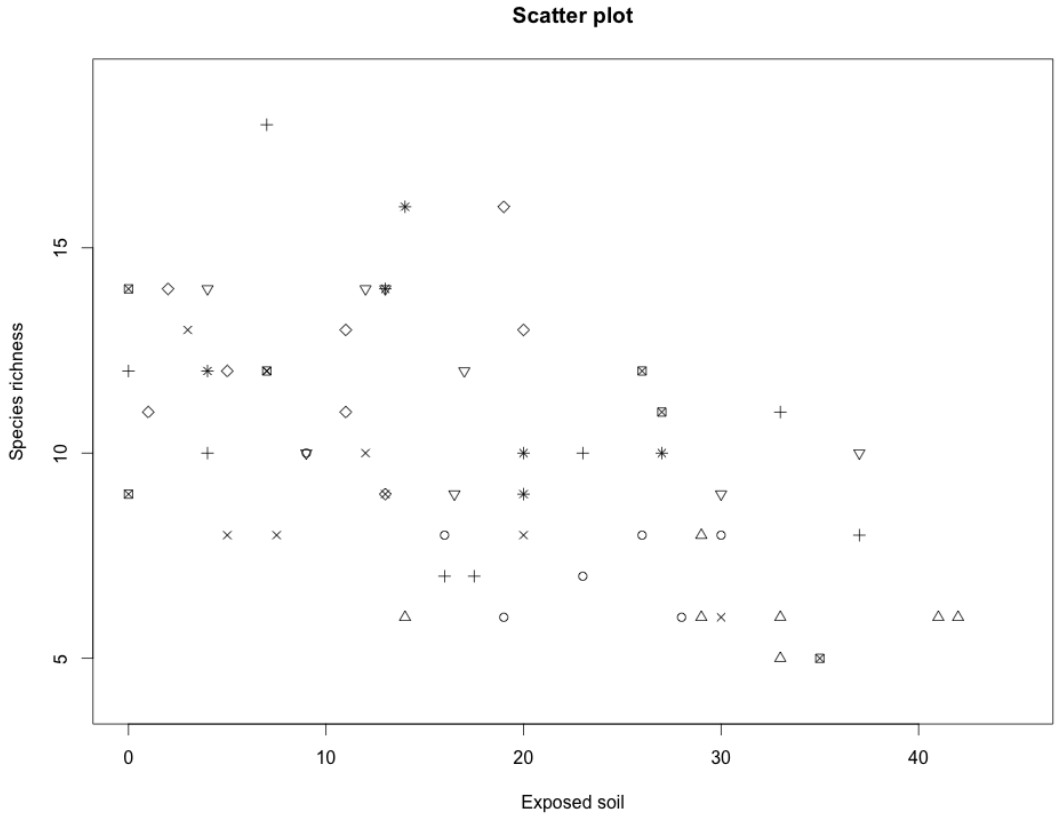
- plot(x = Veg$BARESOIL, y = Veg$R,

  xlab = *"Exposed soil"*,

  ylab = *"Species richness"*,

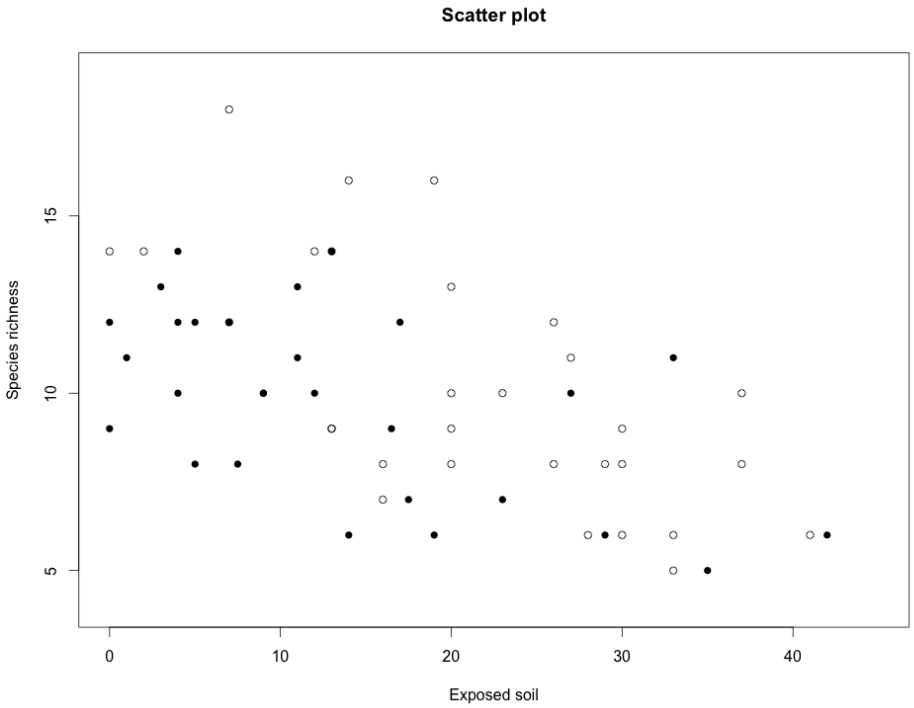
  main = *"Scatter plot"*,

  xlim = c(0, 45), ylim = c(4, 19), pch = Veg$Transect)



- 如果 Veg$Transect 的數值是從 0 開始編號的話（0、1、2、……），就不能直接將 Veg$Transect 指定給 pch，否則所有編號為 0 的資料會畫不出來。
- 如果 Veg$Transect 的資料長度跟 Veg$BARESOIL 與 Veg$R 不同的話，會造成畫出來的資料點符號錯誤。假設 Veg$Transect 的資料長度，R 會在資料長度不夠時，自動重複使用 Veg$Transect 的資料，而在我們這個例子中，所有變數的資料長度都相同，所以沒有這樣的問題。
- pch 只能接受整數的資料，如果指定為 factor 的話，會出現錯誤。

- 假設我們要將資料依據年份（Veg$Time）來區分，把 1974 年以前的資料跟 1974 年以後的資料分開，分別用編號 1 與 16 的符號來表示資料點
- Veg$Time2 <- Veg$Time
- Veg$Time2 [Veg$Time <= 1974] <- 1
- Veg$Time2 [Veg$Time > 1974] <- 16
- plot(x = Veg$BARESOIL, y = Veg$R,
  xlab = *"Exposed soil"*,
  ylab = *"Species richness"*,
  main = *"Scatter plot"*,
  xlim = c(0, 45), ylim = c(4, 19),
  pch = Veg$Time2)

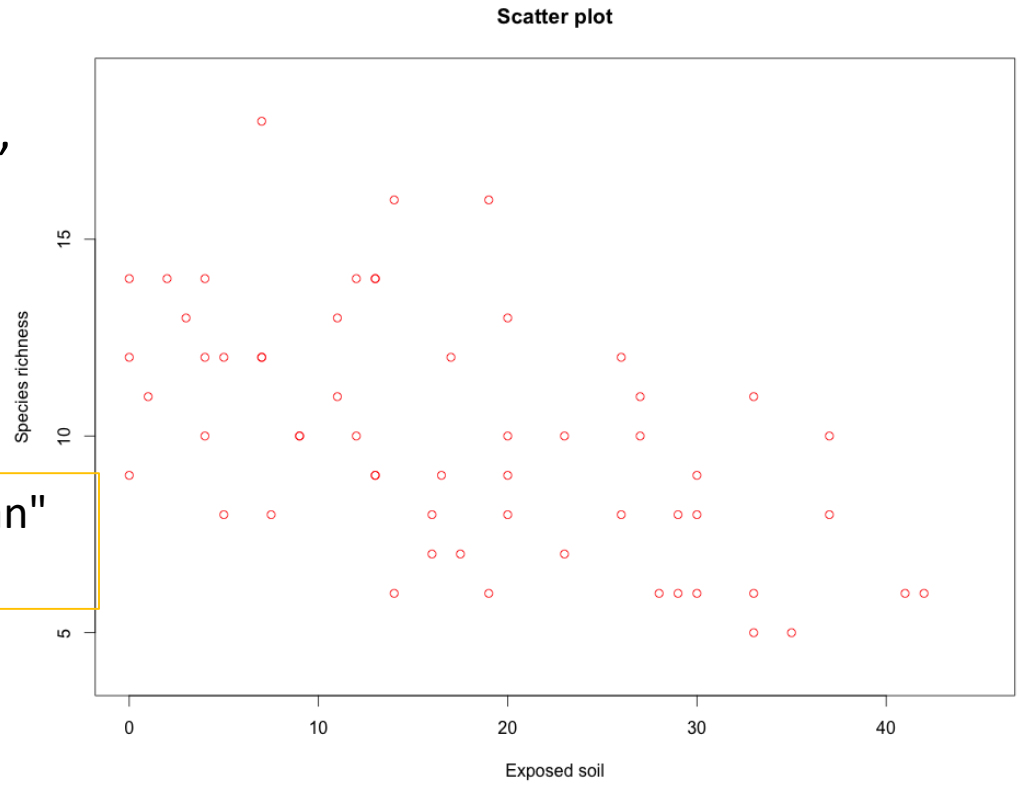


資料點顏色

- plot(x = Veg$BARESOIL, y = Veg$R,
  xlab = *"Exposed soil"*,
  ylab = *"Species richness"*,
  main = *"Scatter plot"*,
  xlim = c(0, 45), ylim = c(4, 19),
  col = 2)

palette()

```
[1] "black"   "red"     "green3"  "blue"    "cyan"
"magenta" "yellow"  "gray"
```



- Veg$Time2 <- Veg$Time
- Veg$Time2 [Veg$Time <= 1974] <- 15
- Veg$Time2 [Veg$Time > 1974] <- 16
- Veg$Col2 <- Veg$Time
- Veg$Col2 [Veg$Time <= 1974] <- 1
- Veg$Col2 [Veg$Time > 1974] <- 2
- plot(x = Veg$BARESOIL, y = Veg$R,

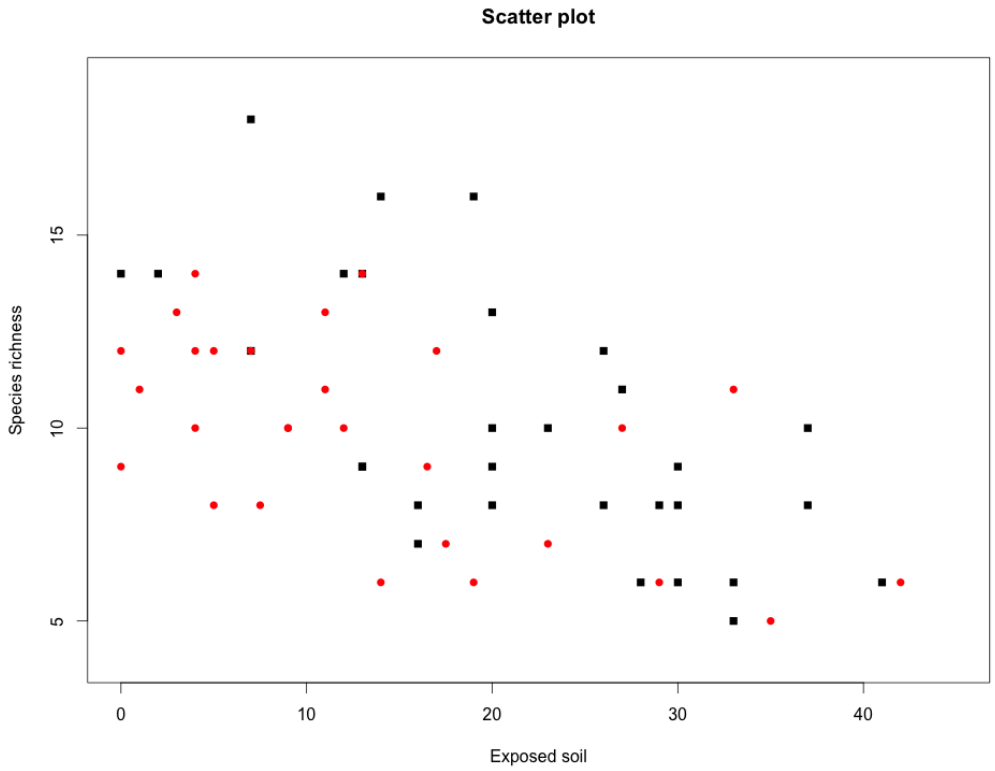
  xlab = *"Exposed soil"*,

  ylab = *"Species richness"*,

  main = *"Scatter plot"*,

  xlim = c(0, 45), ylim = c(4, 19),

  pch = Veg$Time2, col = Veg$Col2)



資料點大小

- 資料點的大小可以使用 cex 參數來指定，其預設值是 1
- plot(x = Veg$BARESOIL, y = Veg$R,

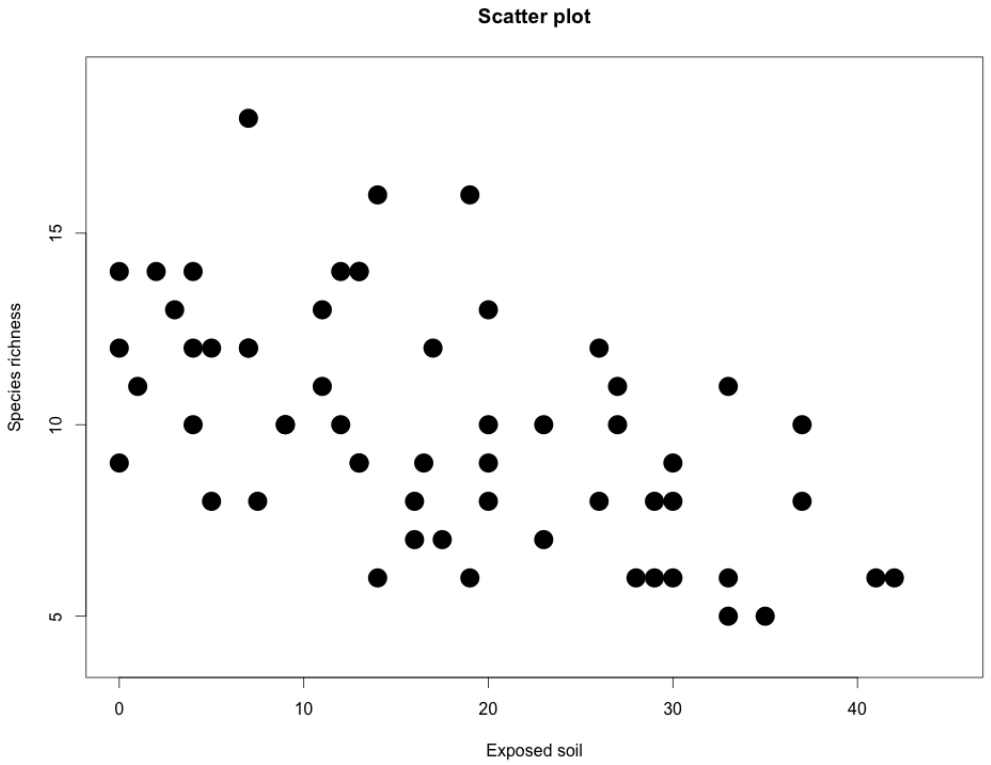
  xlab = *"Exposed soil"*,

  ylab = *"Species richness"*,

  main = *"Scatter plot"*,

  xlim = c(0, 45), ylim = c(4, 19),

  pch = 16, cex = 2.5)



- Veg$Cex2 <- Veg$Time
- Veg$Cex2[Veg$Time == 2002] <- 2
- Veg$Cex2[Veg$Time != 2002] <- 1
- plot(x = Veg$BARESOIL, y = Veg$R,

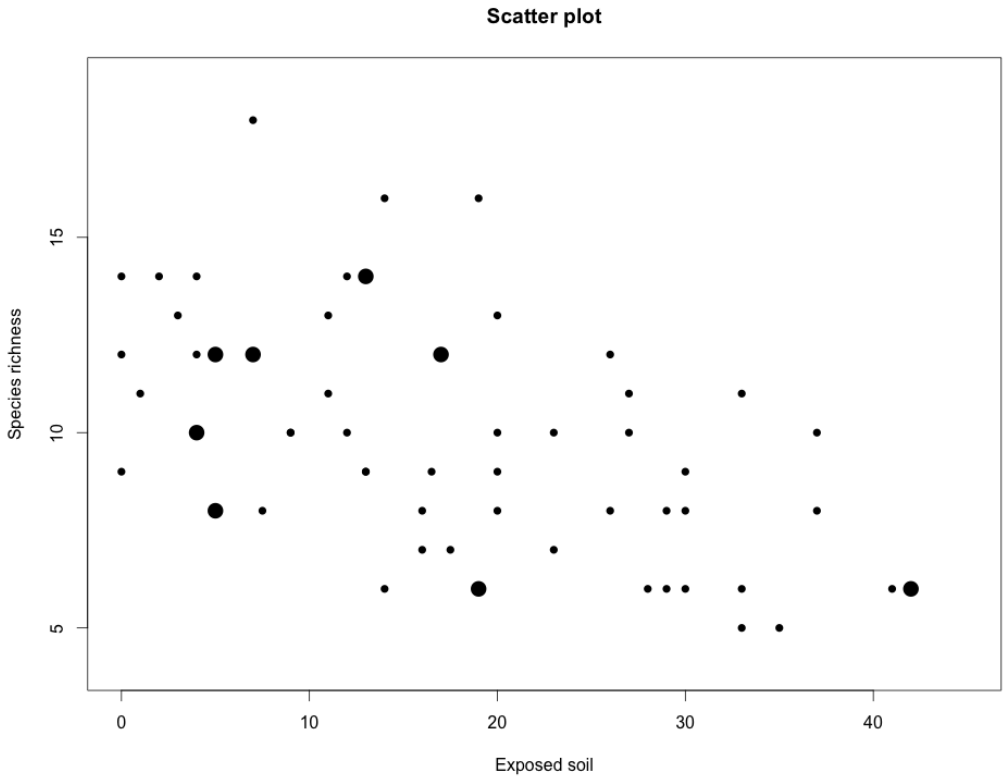
  xlab = *"Exposed soil"*,

  ylab = *"Species richness"*,

  main = *"Scatter plot"*,

  xlim = c(0, 45), ylim = c(4, 19),

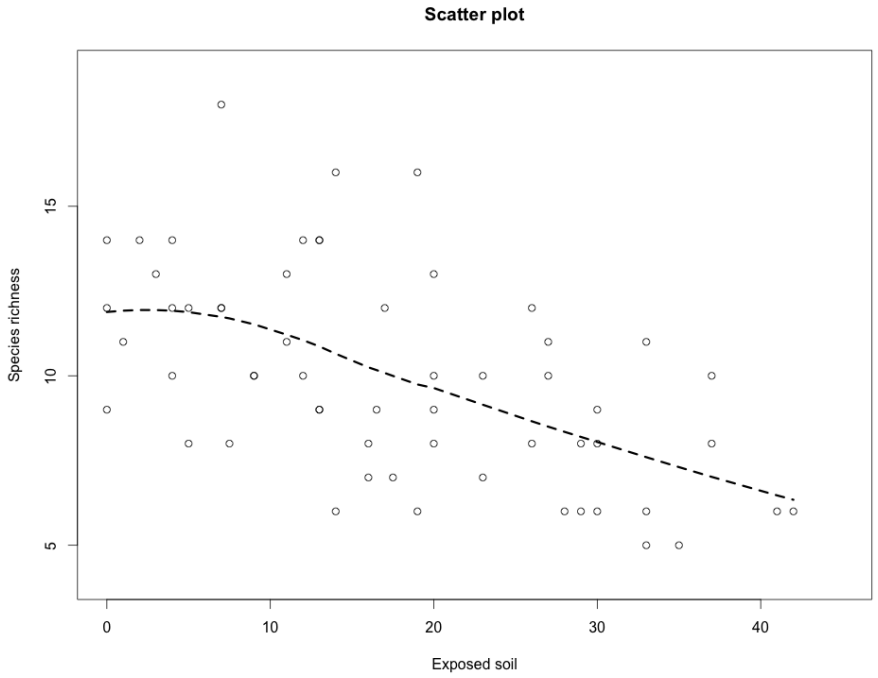
  pch = 16, cex = Veg$Cex2)



平滑曲線

- 由於整個資料沒有很明顯的趨勢，這時候可以利用 loess 配適一條平滑曲線，幫助我們更容易抓出整個資料大概的走勢：

```
plot(x = Veg$BARESOIL, y = Veg$R,
xlab = "Exposed soil",
ylab = "Species richness",
main = "Scatter plot",
xlim = c(0, 45), ylim = c(4, 19))
M.Loess <- loess(R ~ BARESOIL, data = Veg)
Fit <- fitted(M.Loess)
Ord1 <- order(Veg$BARESOIL)
lines(Veg$BARESOIL[Ord1], Fit[Ord1], lwd = 3,
lty = 2)
```

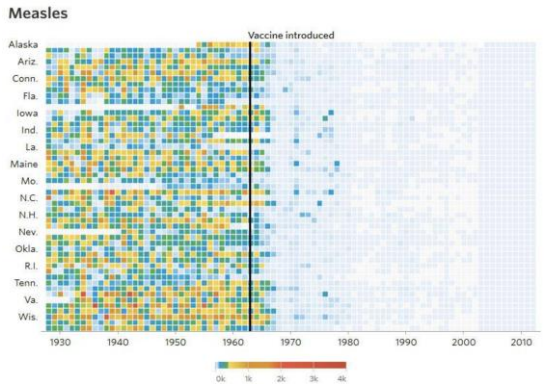


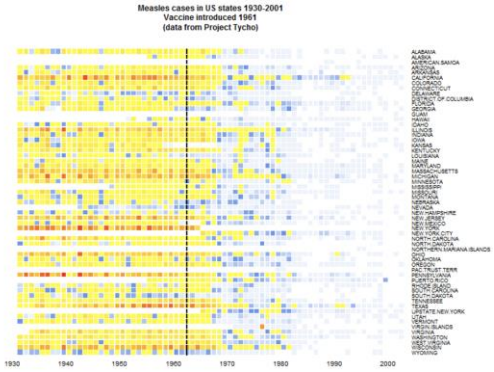
Plot session summary

函數	說明	範例
plot	畫出 x 與 y 的分佈圖（scatter plot）。	plot (x, y)
lines	在圖形中加入線條。	lines (x, y)
order	計算資料的排序向量。	order(x)
loess	LOESS 平滑曲線。	M <- loess (y ~ x)
fitted	計算配適值。	fitted (M)

ggplot2

ggplot2





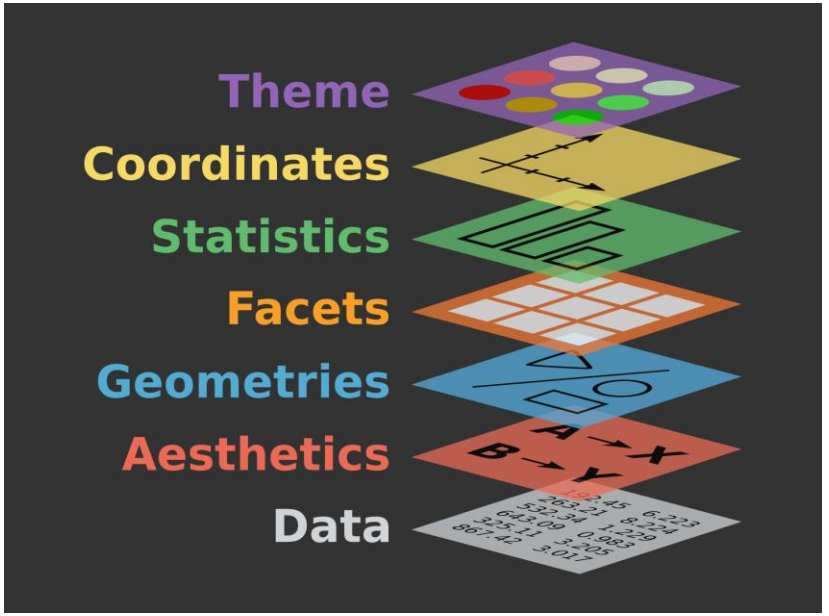
- install.packages("ggplot2")
- library(ggplot2)

- 繪圖文法（grammar of graphics）的概念最早是由 Wilkinson 所提出來的，其主要用來解釋統計圖形（statistical graphic）的概念，而後來 Wickham 則是以 Wilkinson 的繪圖文法理論為基礎，做了一些調整後應用於 R 語言中，發展出了 ggplot 這個繪圖系統。
- ggplot 系統將圖形視為一個從資料開始的一連串轉換，轉換的過程中間有很多道程序，包含幾何圖案、顏色與大小等美學屬性、統計量的計算與座標系統等，一個圖形就是原始資料結合這些轉換之後的結果。

https://benjaminlmoore.wordpress.com/2015/04/09/recreating-the-vaccination-heatmaps-in-r/

ggplot 繪圖文法

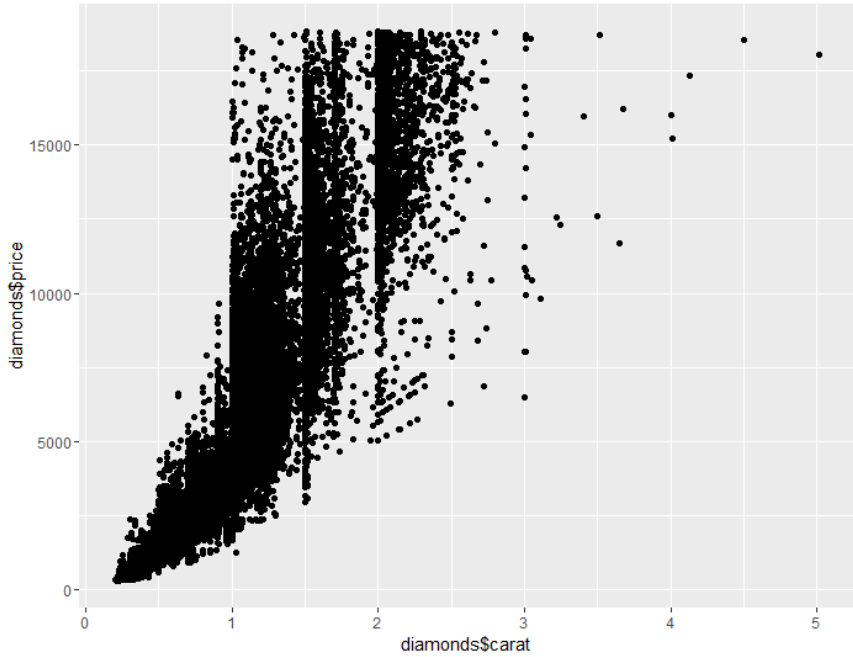
- **資料來源（data）**：指定原始資料來源的 data frame。
- **美學對應（aesthetic）**：指定原始資料與圖形之間的對應關係，例如哪一個變數要當作 x 座標變數，而哪一個要當作 y 座標變數，還有資料繪圖時的樣式等。
- **幾何圖案（geometry）**：要用什麼幾何圖形繪製資料，例如點、線條、多邊形等。
- **繪圖面（facet）**：指定如何將資料分散在多張子圖形中繪製，以利互相比較。
- **統計轉換（statistical transformation）**：指定如何以將資料轉換為各種統計量，例如將連續型資料轉為離散型的類別。
- **座標系統（coordinate system）**：指定繪圖時所使用的座標系統，除了常見的笛卡兒直角座標系統，也可以使用極坐標或地圖投影（map projection）。
- **主題（theme）**：控制資料以外的繪圖組件，例如座標軸、說明文字等。



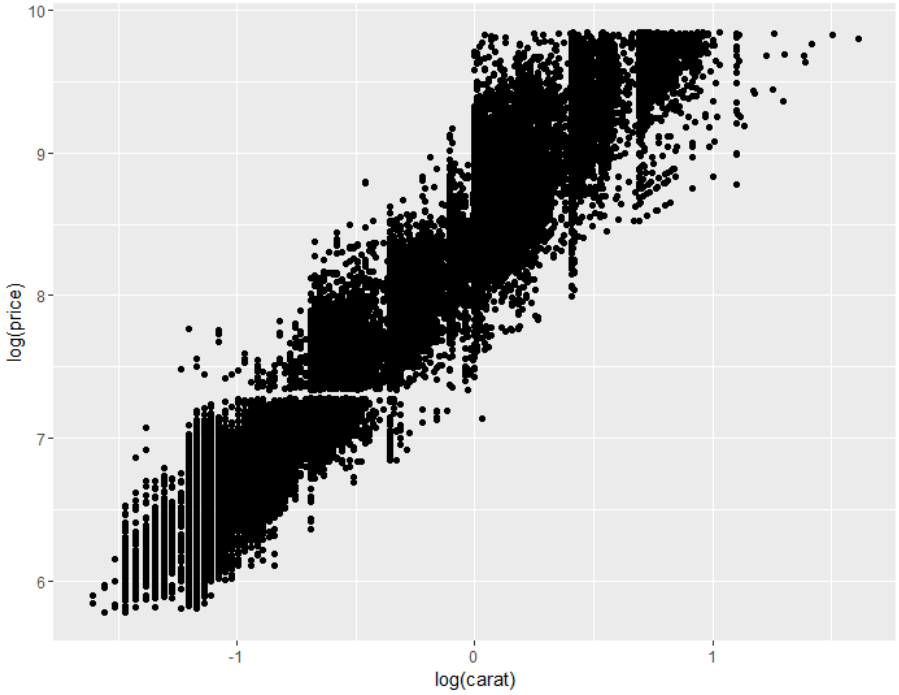
qplot

- qplot 函數是 ggplot 系統中最基本的一個繪圖函數，它的設計類似傳統 plot 函數，方便讓初次使用 ggplot 的使用者快速上手，繪製一些基本的圖形。

```
head(diamonds)
set.seed(5)   #設定亂數種子
diamonds.subset <- diamonds[sample(nrow(diamonds), 100), ]

qplot(diamonds$carat, diamonds$price)
qplot(carat, price, data = diamonds)
```

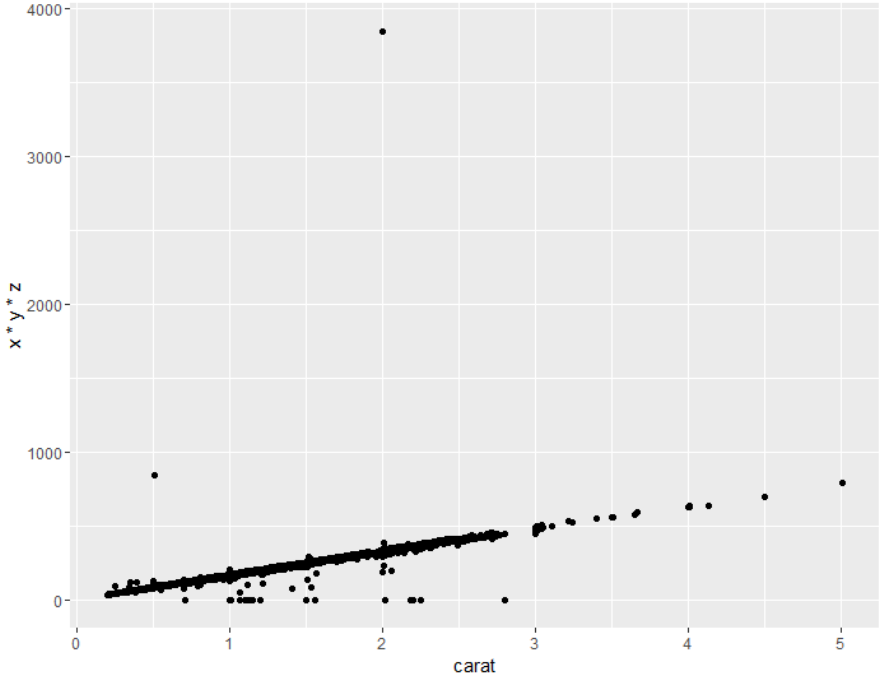


將變數透過對數（log）轉換一下：
qplot(log(carat), log(price), data = diamonds)



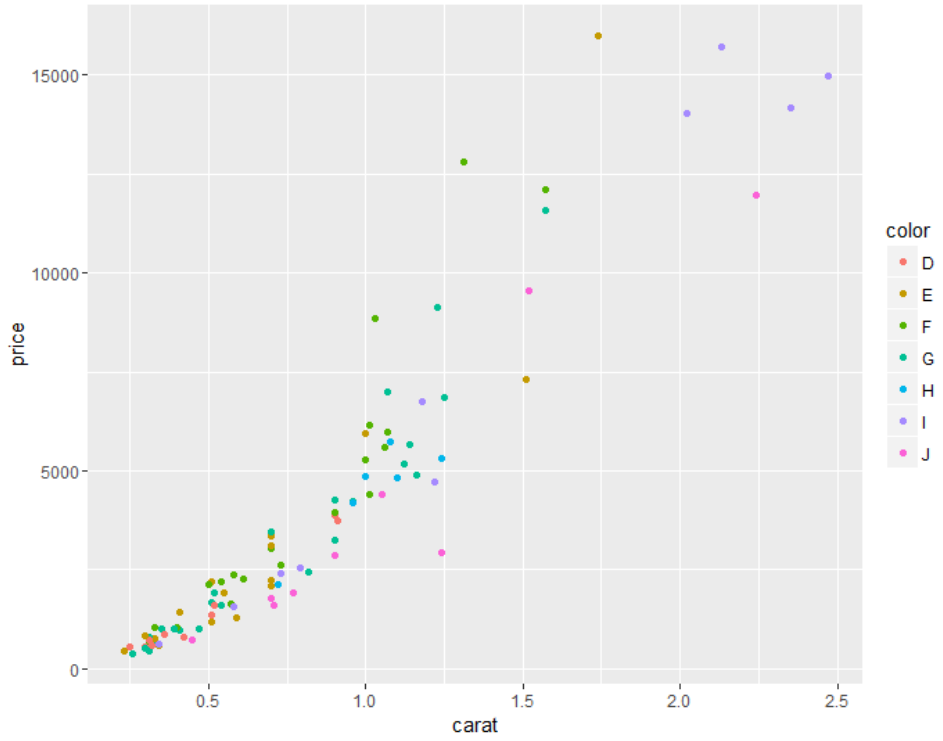
使用多個變數進行運算之後，作為 x 軸或 y 軸的座標值：

qplot(carat, x * y * z, data = diamonds)



圖形樣式

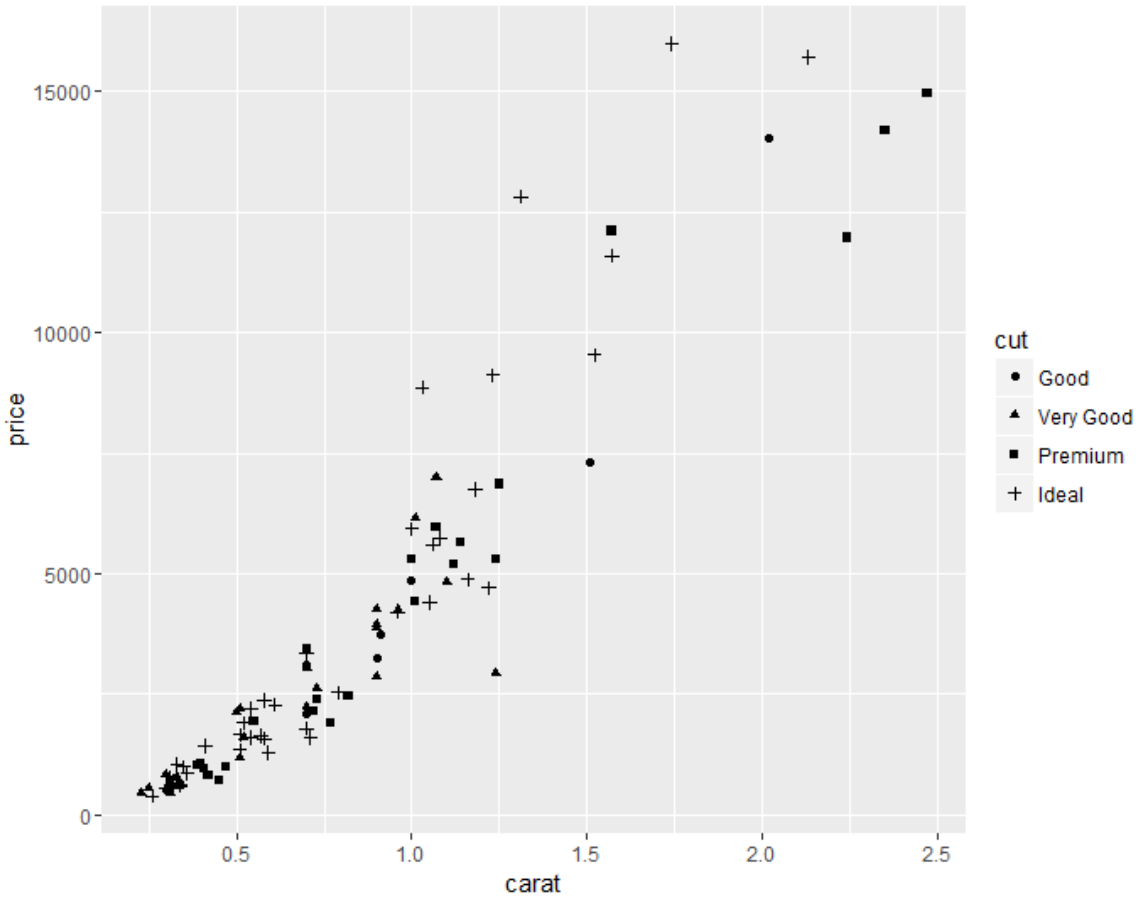
- qplot 可以自動處理這些繁瑣的動作，並且在圖形上加入圖示說明（legend）。
- 假設我們想要依據 diamonds 中的 color 欄位來替資料點著色，區別不同顏色的鑽石，可以使用 color 參數：
- qplot(carat, price, data = diamonds.subset, color = color)



資料點的形狀區分資料，可使用 shape：

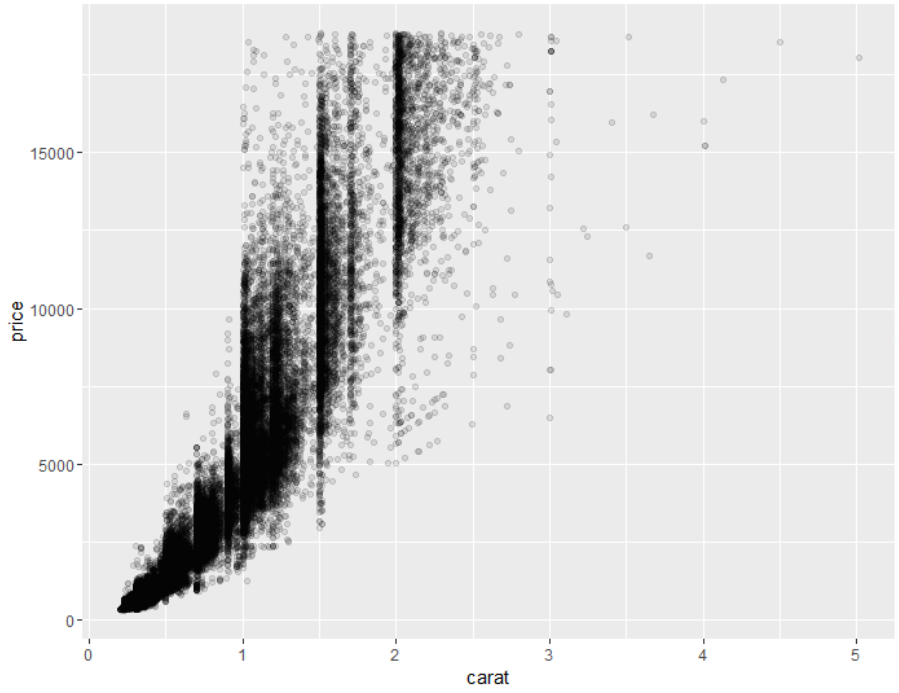
qplot(carat, price, data = diamonds.subset, shape = cut)

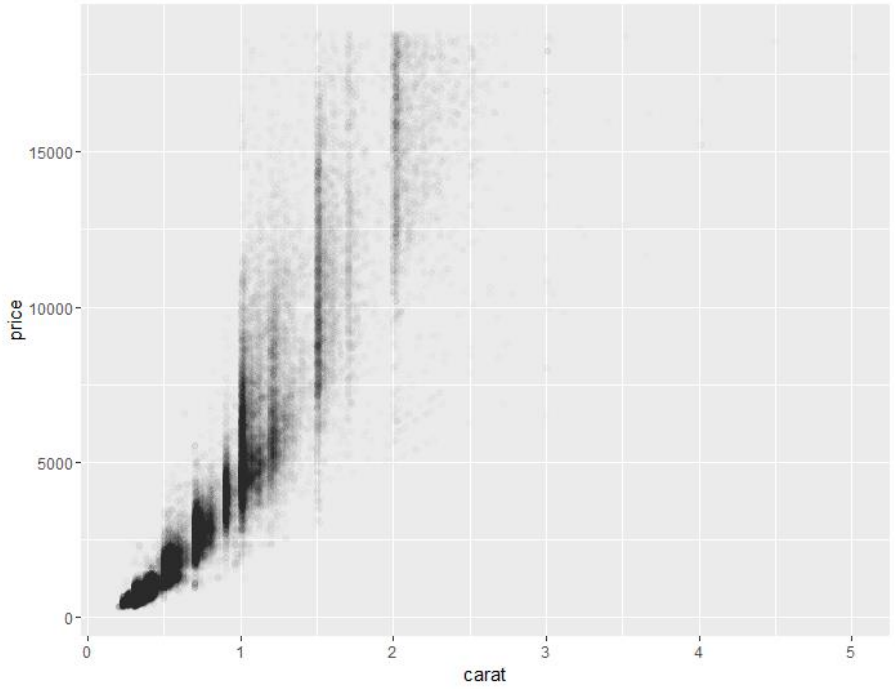
cut 欄位



資料點加上透明度的參數，這樣可以比較容易辨識實際的資料分佈情況：

```
qplot(carat, price, data = diamonds, alpha = I(1/10))
qplot(carat, price, data = diamonds, alpha = I(1/100))
```





隨著資料類型的不同，適合的資料呈現方式也不同，
例如**類別型**的資料就適合使用顏色、形狀來區隔，
而若是**連續型**的資料則可以使用資料點的大小來表示；
至於資料量的多寡也會有影響，資料量多的時候，除了使用透明度的技巧，也可以考慮以繪圖面（facet）繪出多張圖形的方式，避免過多的資料擠在同一張圖上難以區分。

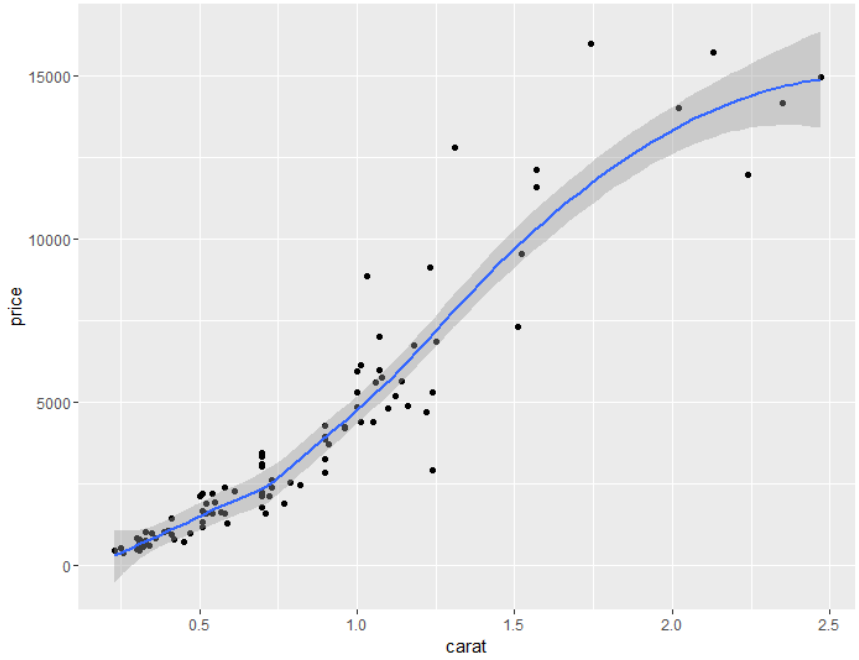
各式圖形

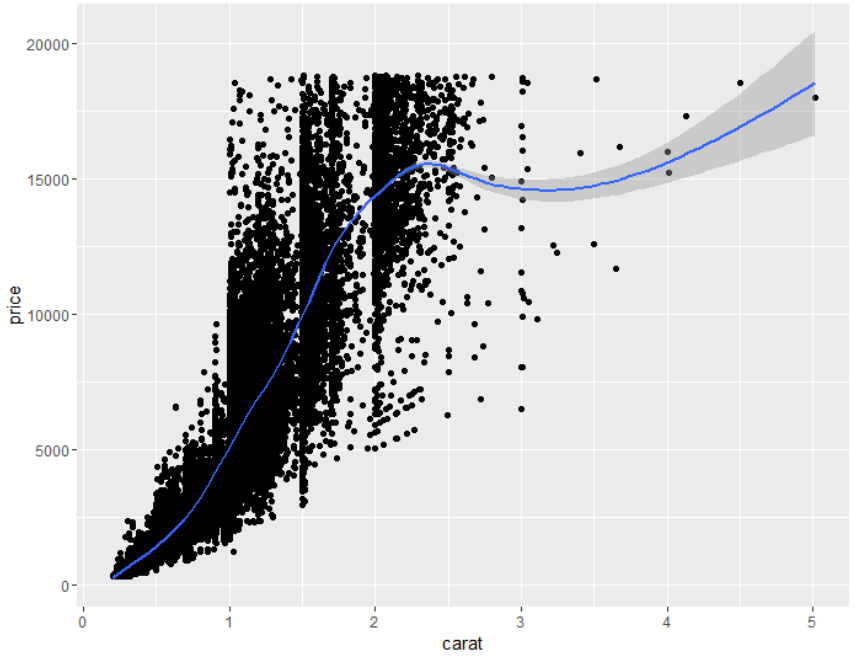
geom 參數可以設定用來呈現資料所使用的幾何圖形，我們可以藉著調整此參數來做出各種變化，有些 geom 參數還會伴隨一些統計量的計算（例如顯示直方圖的同時也需要計算每個 bin 的統計量）

- **呈現二維資料時比較常用的 geom 選項：**
  - "point"：使用點的方式呈現資料，這也是 qplot 在繪製二維資料時預設的方式。
  - "smooth"：以 smoother 配適資料，畫出平滑曲線與標準誤差（standard error）。
  - "boxplot"：以箱形圖呈現資料分佈情形。
  - "path"：以線段連接每一個資料點。
  - "line"：類似 "path"，但 "line" 只能產生由左至右的線段圖形。
- **一維的資料，常用的 "geom" 參數如下：**
  - "histogram"：直方圖，適用於連續型的資料，在資料是一維的情況下，預設會使用此方式。
  - "freqpoly"：類似直方圖，但以折線表示。
  - "density"：密度函數圖，適用於連續型的資料。
  - "bar"：長條圖，適用於離散型的類別資料。

在圖形上加入 Smoother

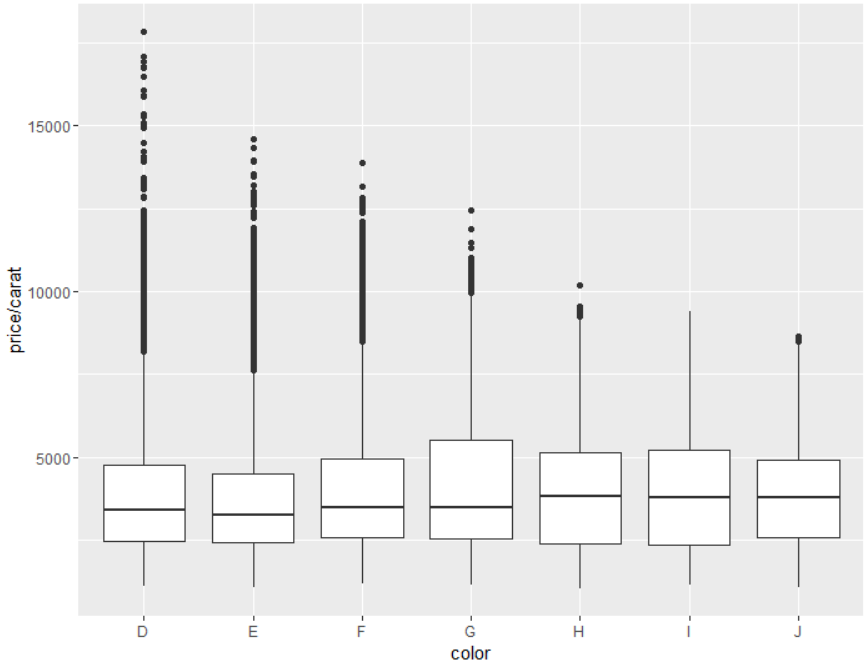
- qplot(carat, price, data = diamonds.subset, geom = c("point", "smooth"))
- qplot(carat, price, data = diamonds, geom = c("point", "smooth"))

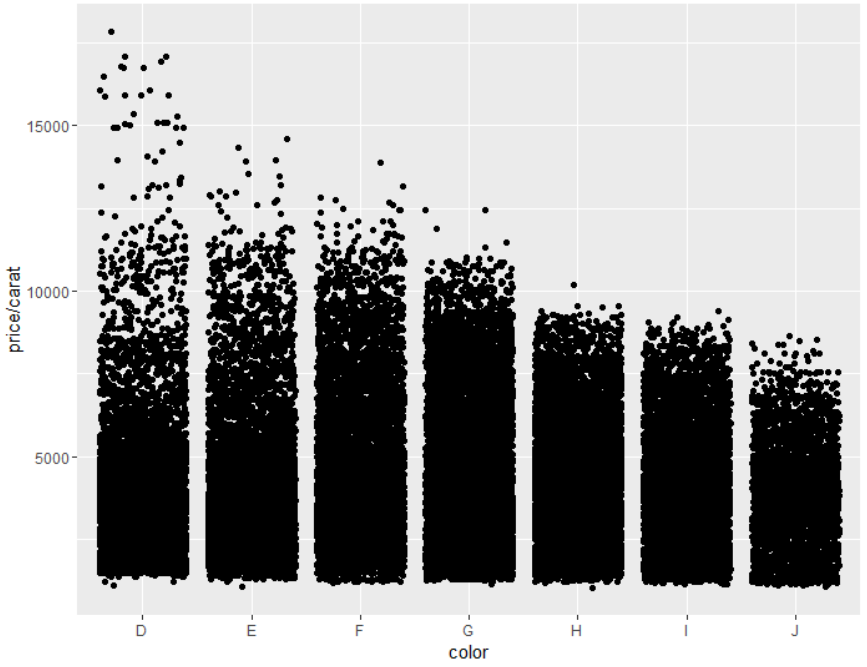




箱形圖與 Jitter 資料點

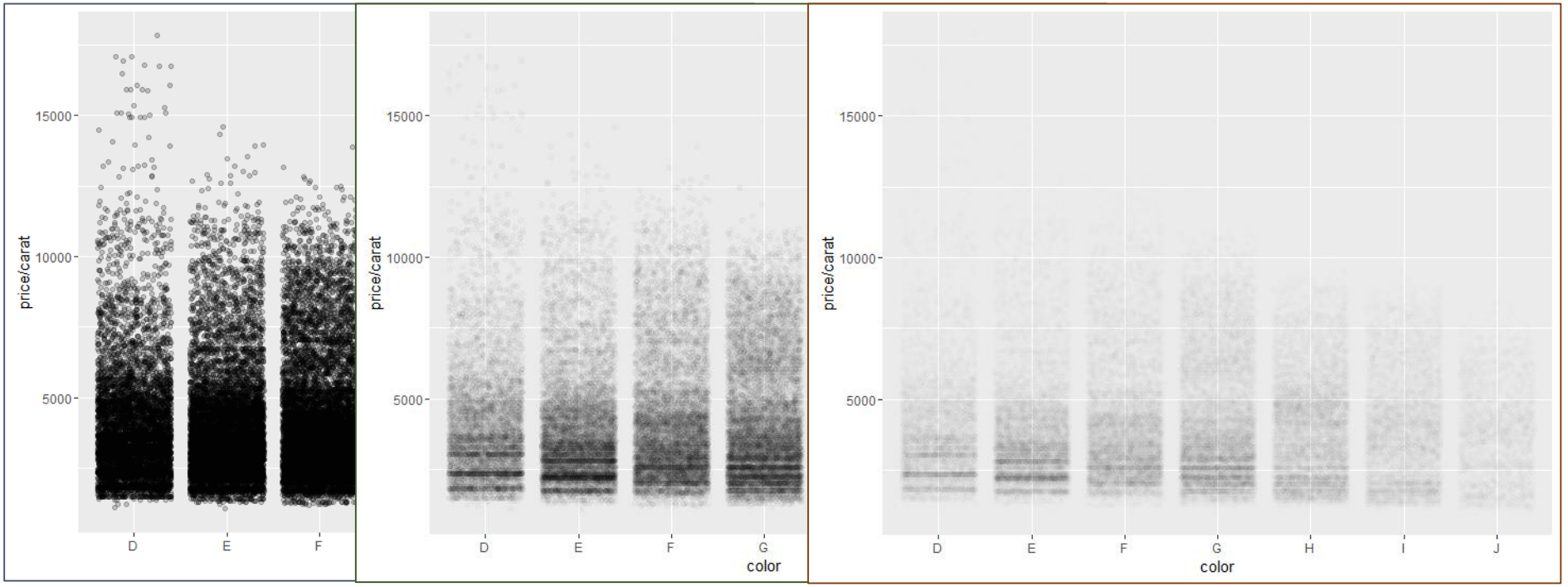
- 當一組資料同時含有一個類別型的變數以及一個或多個連續型的變數時，資料分析者通常都會想要比較各類別中各連續型變數的分佈狀況，而最常用的圖形就是箱形圖：
- qplot(color, price / carat, data = diamonds, geom = "boxplot")
- 若想要看到更細部的資訊，可以使用 jitter 的方式繪製資料點，它可以將每一個資料點都畫出來：
- qplot(color, price / carat, data = diamonds, geom = "jitter")





資料量太多的時候，可以使用透明度的技巧：

- qplot(color, price / carat, data = diamonds, geom = "jitter", alpha = I(1 / 5))
- qplot(color, price / carat, data = diamonds, geom = "jitter", alpha = I(1 / 50))
- qplot(color, price / carat, data = diamonds, geom = "jitter", alpha = I(1 / 200))



以 jitter 的方式繪圖可以顯示資料分佈的細部資訊，彌補箱形圖的不足。以這個例子而言，雖然不同透明度的 jitter 資料點可以顯示資料集中的區域，不過箱形圖可能還是比較容易辨識。

以 jitter 繪製資料點時，可以同時配合散佈圖中的各種參數來調整圖形，例如：size、color 與 shape，而箱形圖也可以使用 color、fill 與 size 來調整顏色與線條粗細。

EXERCISE

- # 以資料點的形狀區分 cut 變數
- qplot(color, price / carat, data = diamonds, geom = "jitter", alpha = I(1 / 5), shape = cut)

- # 以資料點的顏色區分 color 變數
- qplot(color, price / carat, data = diamonds, geom = "jitter", alpha = I(1 / 5), color = color)

- # 以資料點的顏色區分 cut 變數
- qplot(color, price / carat, data = diamonds, geom = "jitter", alpha = I(1 / 5), color = cut)

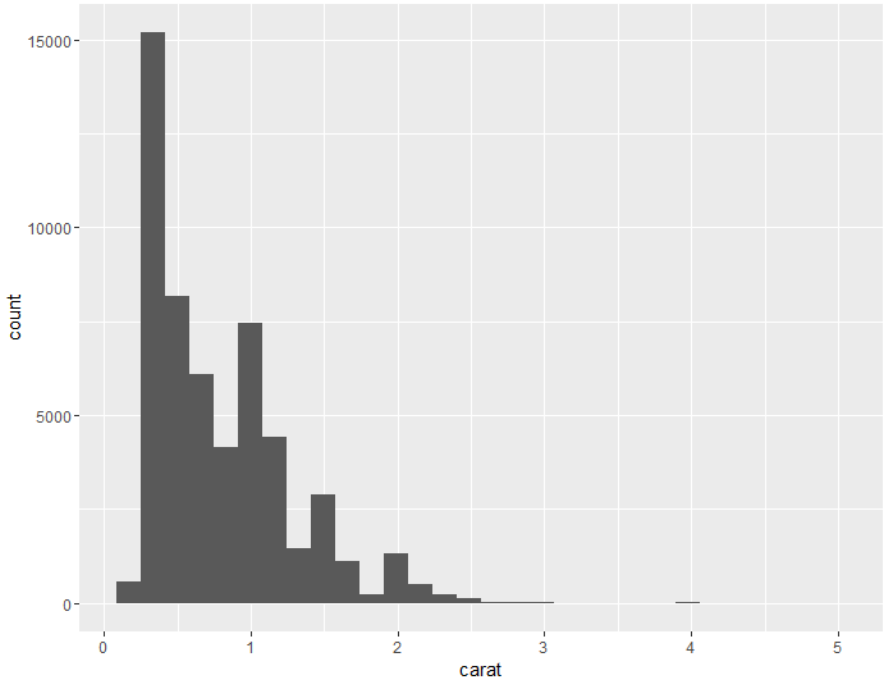
- # 以箱形圖的外框顏色區分 color 變數
- qplot(color, price / carat, data = diamonds, geom = "boxplot", color = color)

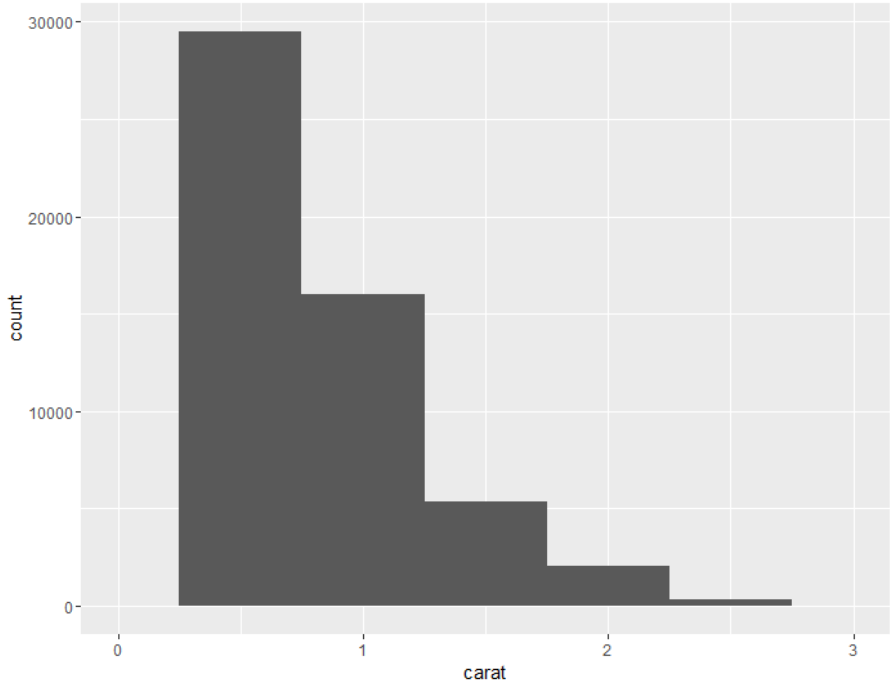
- # 以箱形圖的內部顏色區分 color 變數
- qplot(color, price / carat, data = diamonds, geom = "boxplot", fill = color)

- # 調整箱形圖的外框粗細
- qplot(color, price / carat, data = diamonds, geom = "boxplot", size = I(2))

直方圖與密度函數

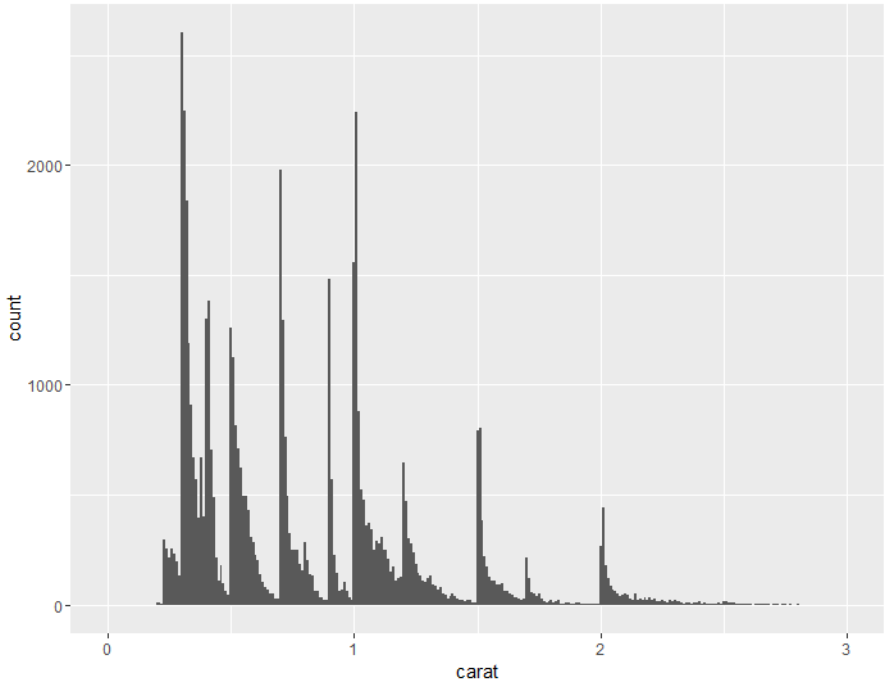
- qplot(carat, data = diamonds, geom = "histogram")
- qplot(carat, data = diamonds, geom = "histogram",binwidth = 0.5, xlim = c(0, 3))

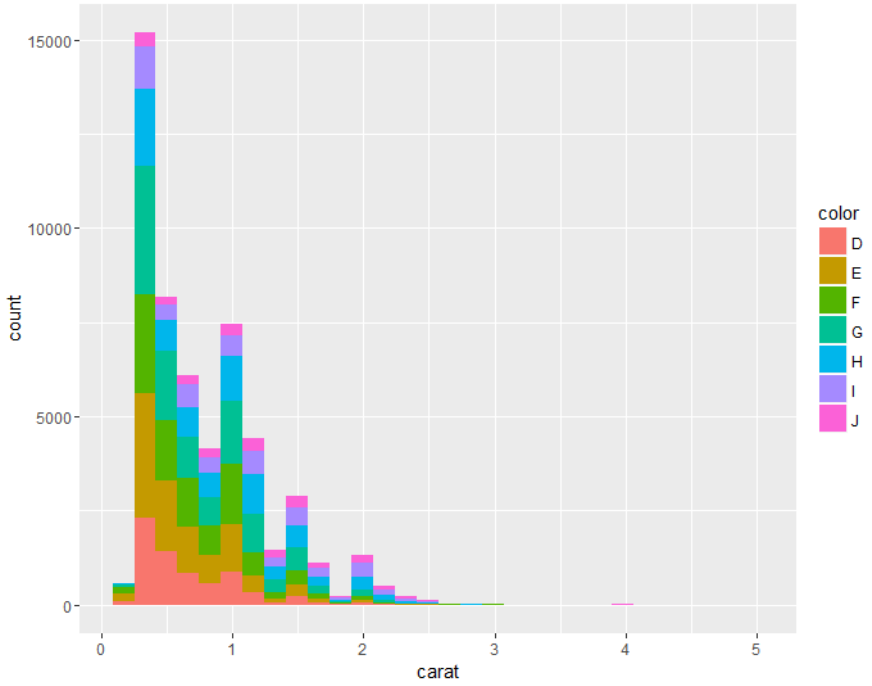




同時呈現多組資料、相互比較時，可以加上美學對應：

- qplot(carat, data = diamonds, geom = "histogram", binwidth = 0.01, xlim = c(0, 3))
- qplot(carat, data = diamonds, geom = "histogram", fill = color)

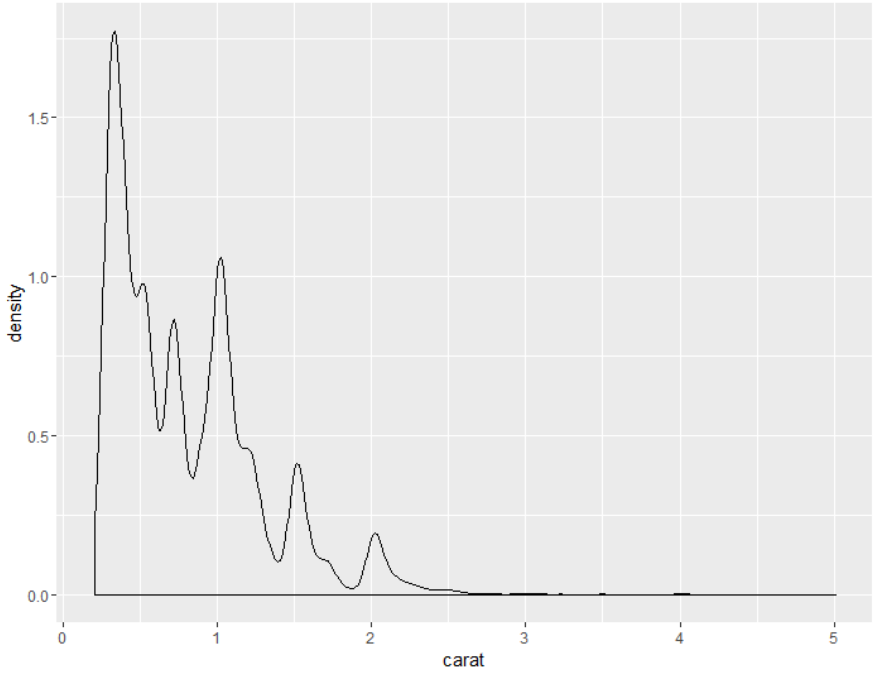


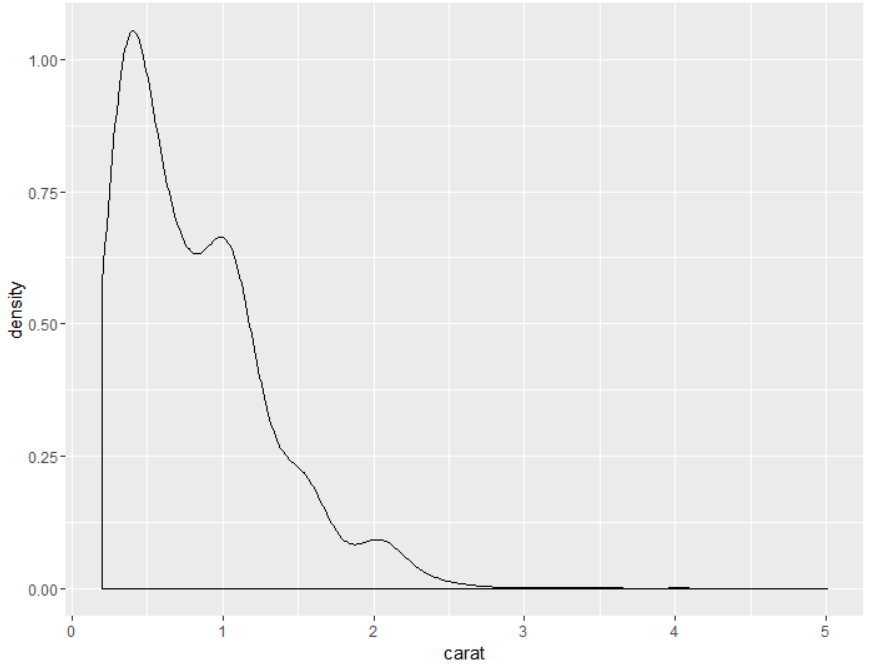


當美學對應(**aesthetics**)指定為一個類別型的變數時，會讓資料以此類別變數為依據區分為多個群組，所以 qplot 在這種狀況下就會用不同顏色畫出不同的鑽石顏色的資料，產生堆疊式的直方圖。

密度函數圖

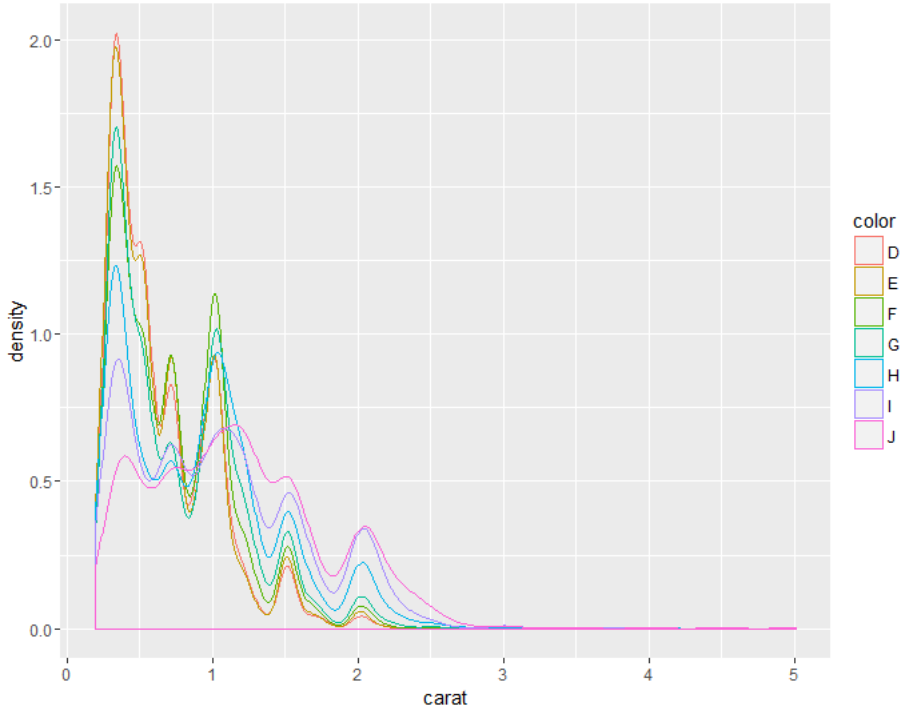
- qplot(carat, data = diamonds, geom = "density")
- 使用 adjust 參數調整密度函數圖的平滑程度，這個值越大則曲線越平滑，其效果類似直方圖的 bin 寬度：
- qplot(carat, data = diamonds, geom = "density", adjust = 3)





密度函數圖可以同時呈現多組資料的分佈：

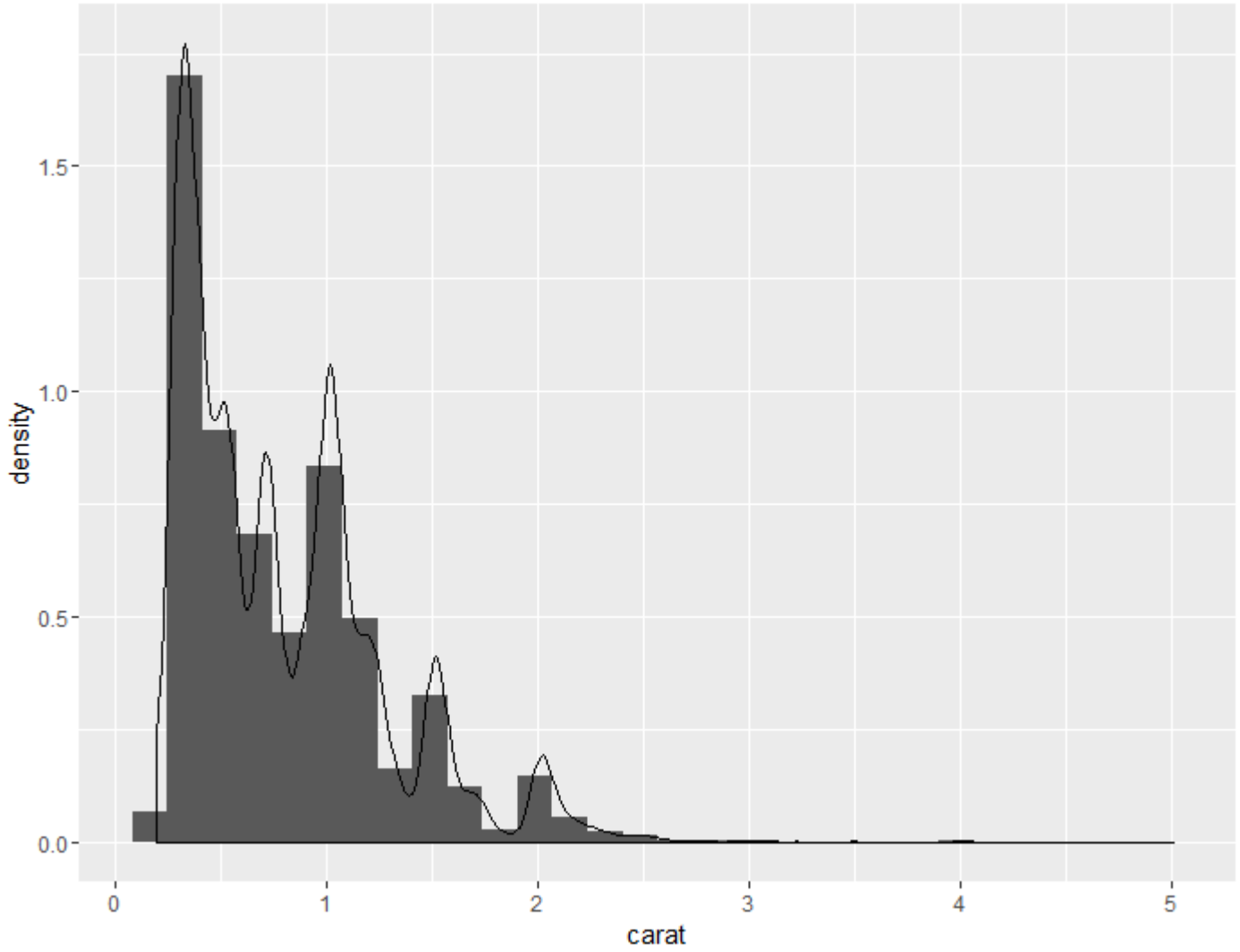
- qplot(carat, data = diamonds, geom = "density", color = color)



在比較多組資料時，密度函數圖可能比較容易閱讀，不過這種圖形是假設資料本身是屬於無界（unbounded）的連續型資料，若資料的類型不屬於這種就要注意。

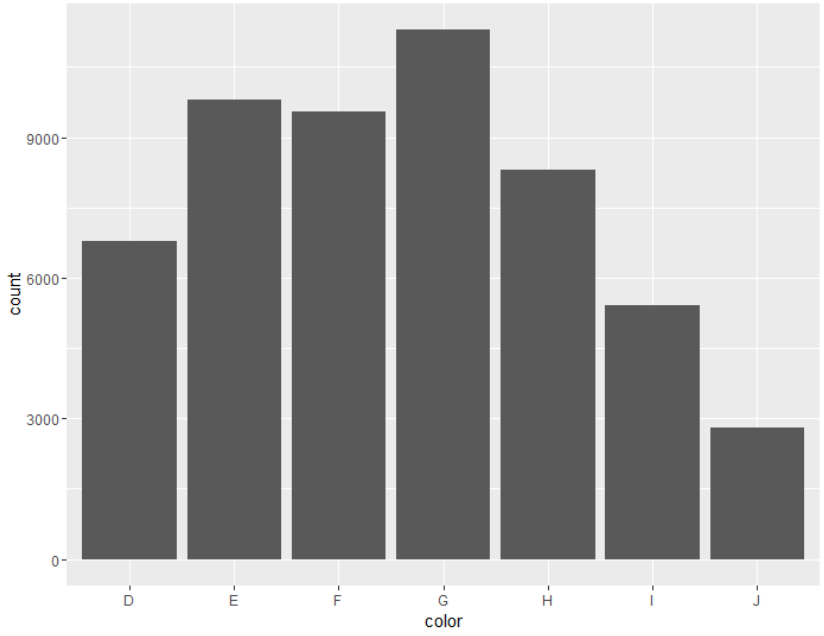
將直方圖密度函數圖畫在一起，可將 y 軸的單位指定為密度，再畫出兩種圖形：

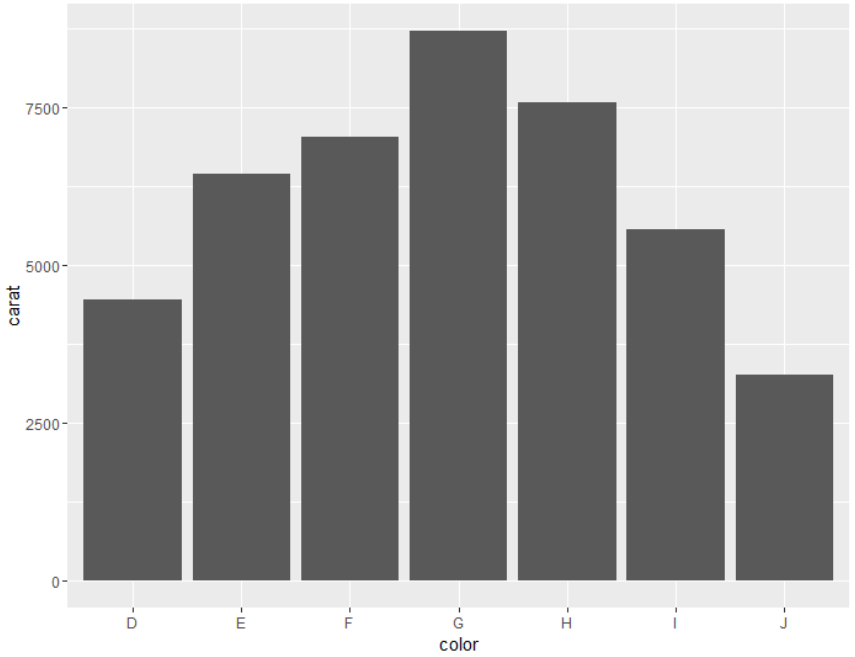
- qplot(carat, ..density.., data = diamonds, geom = c("histogram", "density"))



長條圖

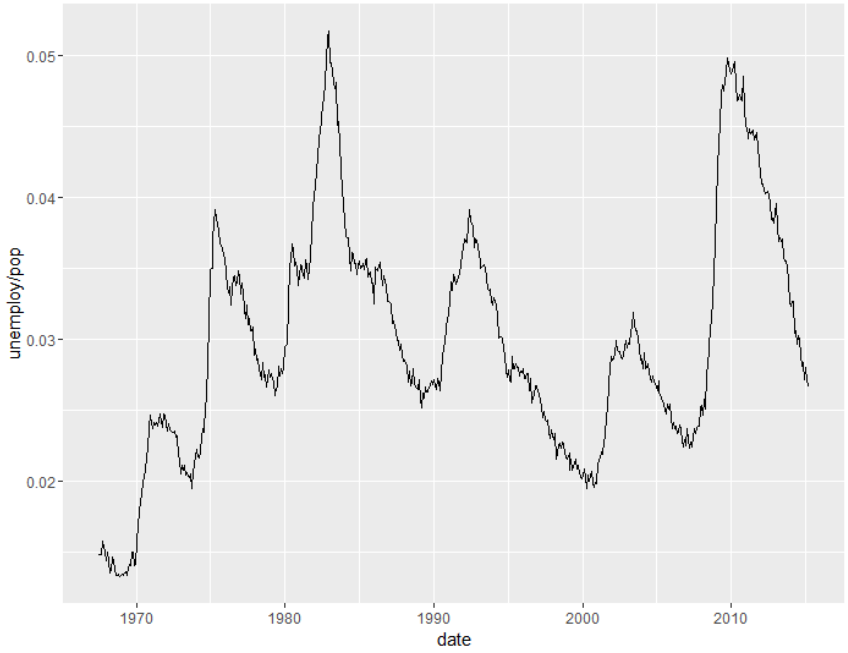
- 表示類別型的離散資料，在 ggplot 系統下使用 bar 這個 geom 繪製長條圖時，它會自動計算每個類別的數量，不需要像傳統的 barchart 一樣自行計算列聯表。
- 如果您已經把列聯表算出來了，或是想要自行指定列聯表的計算方式，可以使用 weight 這個參數指定數值的來源。
- qplot(color, data = diamonds, geom = "bar")
- qplot(color, data = diamonds, geom = "bar", weight = carat) + ylab("carat")

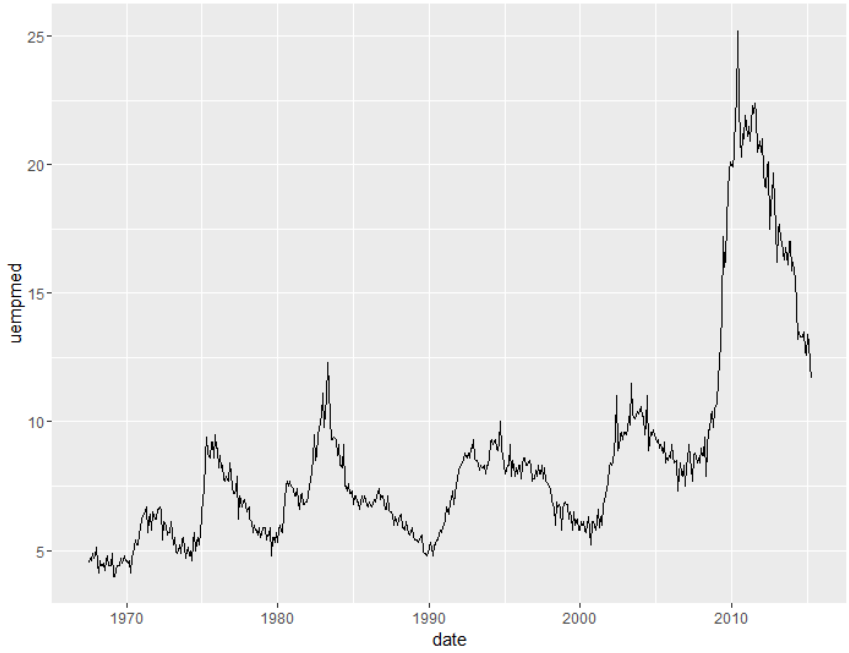




時間序列與路徑

- path 會依據資料在 data frame 中的順序，將每一個點以線段連接起來，
- 而 line 則是會將點的順序依據 x 軸座標來排序。通常 line 圖形的 x 軸是時間的資訊，用來呈現某個變數隨著時間的變化，而 path 則是用在比較兩個變數隨的時間變化的關係。
- qplot(date, unemploy / pop, data = economics, geom = "line")
- qplot(date, uempmed, data = economics, geom = "line")

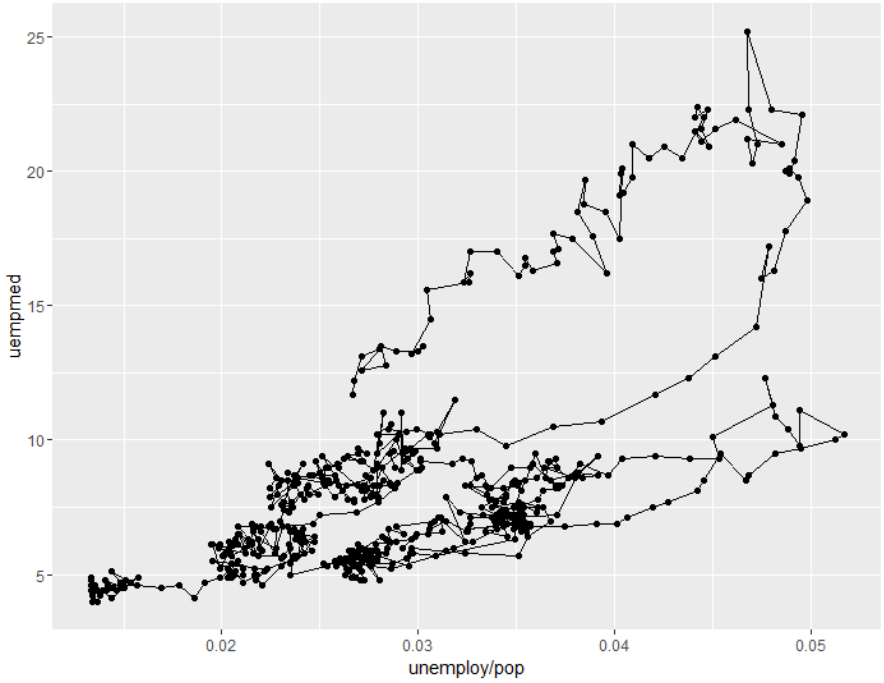


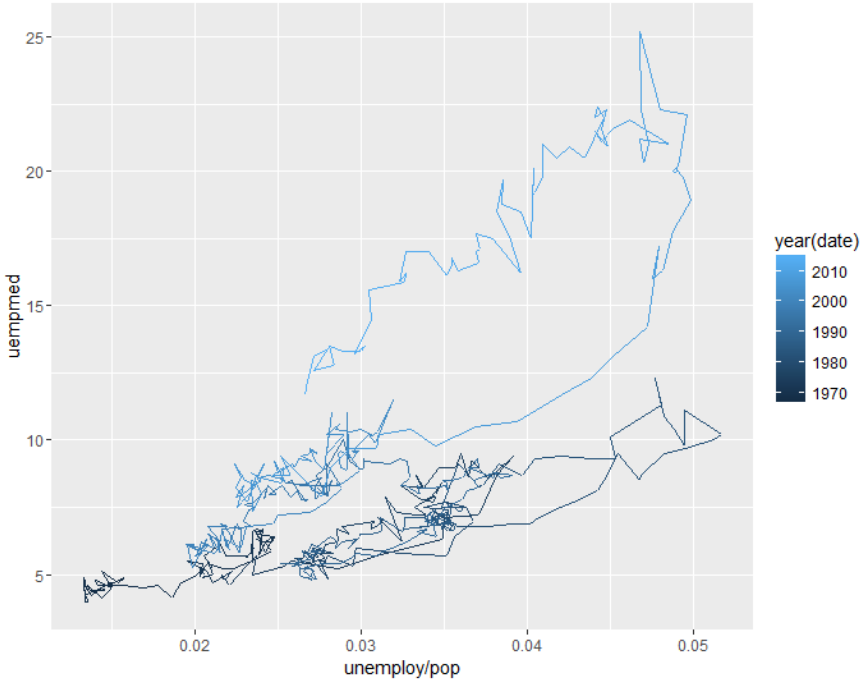


進一步細看失業率與失業時間之間的關係

- qplot(unemploy / pop, uempmed, data = economics, geom = c("point", "path"))
- 為了讓時間的資訊更容易辨識，可以再加上顏色來表示不同的年份：

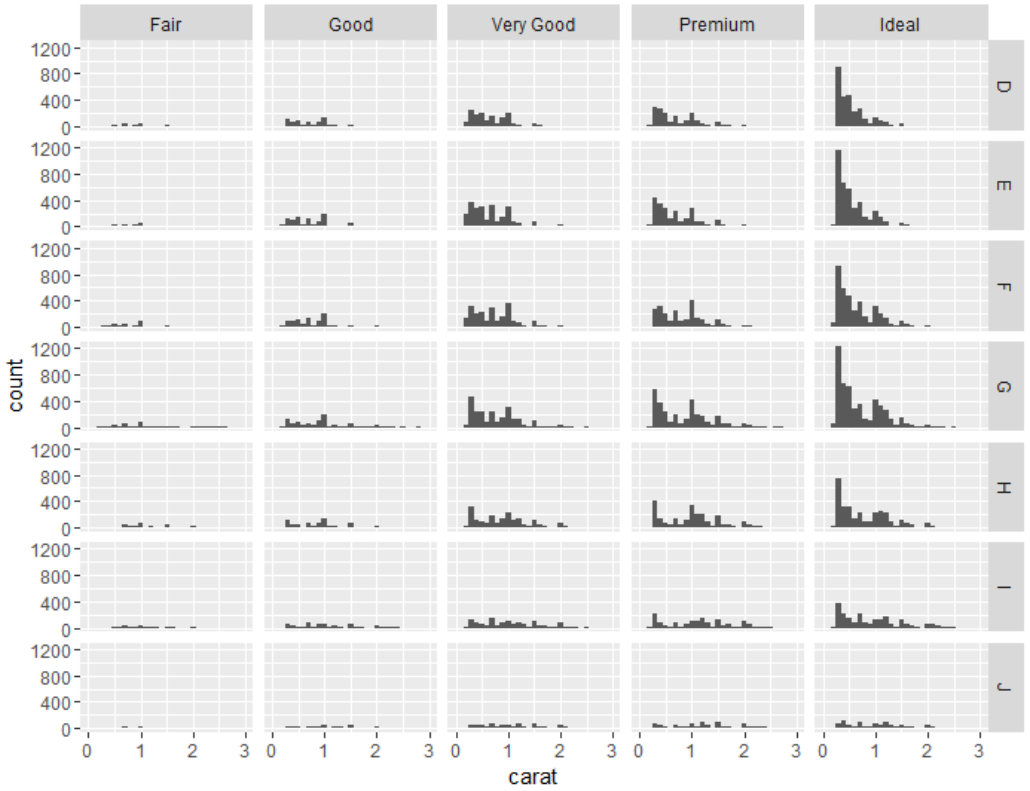
- year <- function(x) as.POSIXlt(x)$year + 1900
- qplot(unemploy / pop, uempmed, data = economics, geom = "path", colour = year(date))





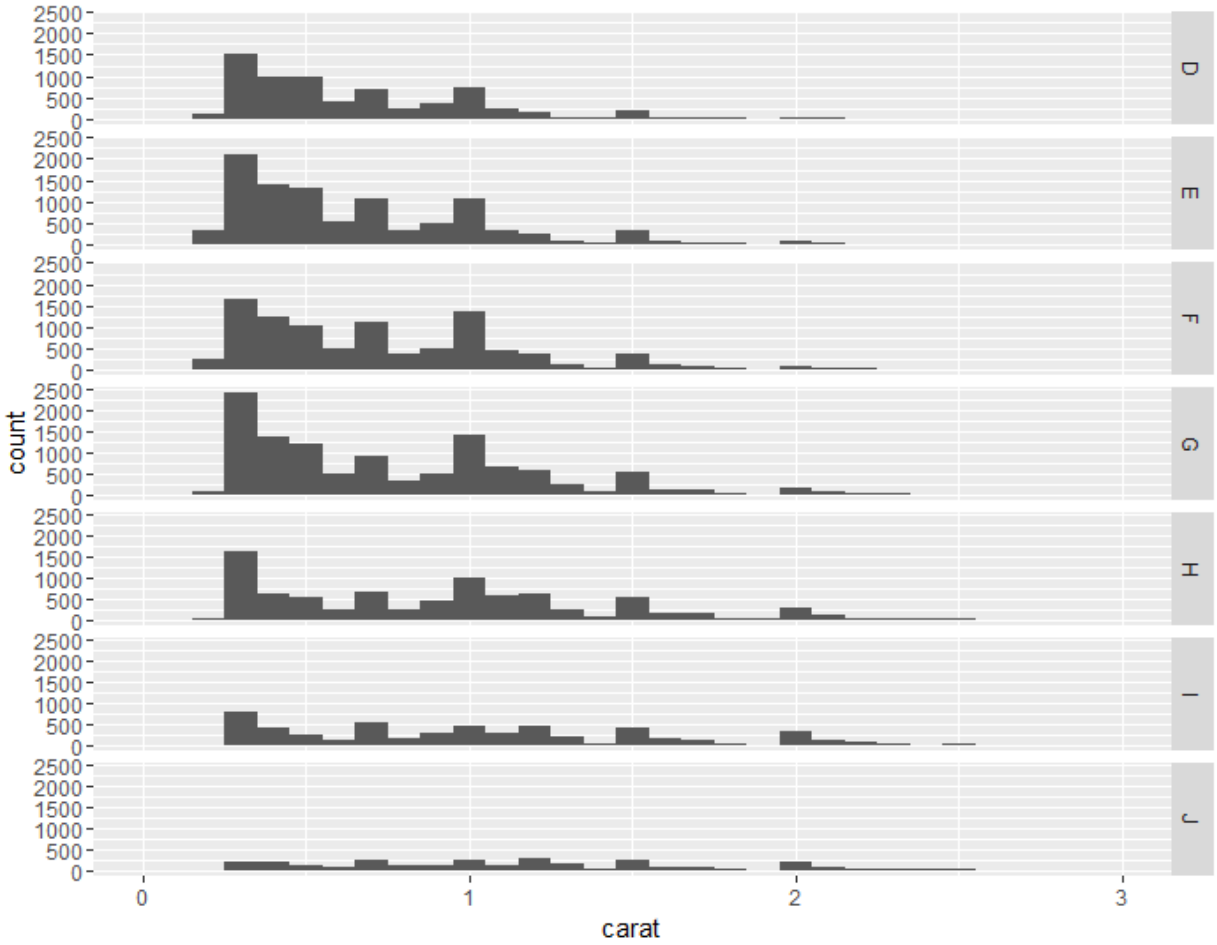
繪圖面（Facet）

- 提供另外一種資料區隔方式，它會先將資料分組後畫在表格式圖形中，而每一個子圖形的座標都相同，方便互相比較。
- qplot 的繪圖面功能是使用 facets 參數配合公式的方式指定，公式的左邊是列（row）、右邊是行（column），例如：
- qplot(carat, data = diamonds, facets = color ~ cut, geom = "histogram", binwidth = 0.1, xlim = c(0, 3))



若只需要對一個變數分組，則將公式另一側指定為一個句點（.），例如：

- qplot(carat, data = diamonds, facets = color ~ ., geom = "histogram", binwidth = 0.1, xlim = c(0, 3))



其他用法

- qplot 還有一些可以調整圖形的參數，這些用法與傳統的 plot 參數類似。

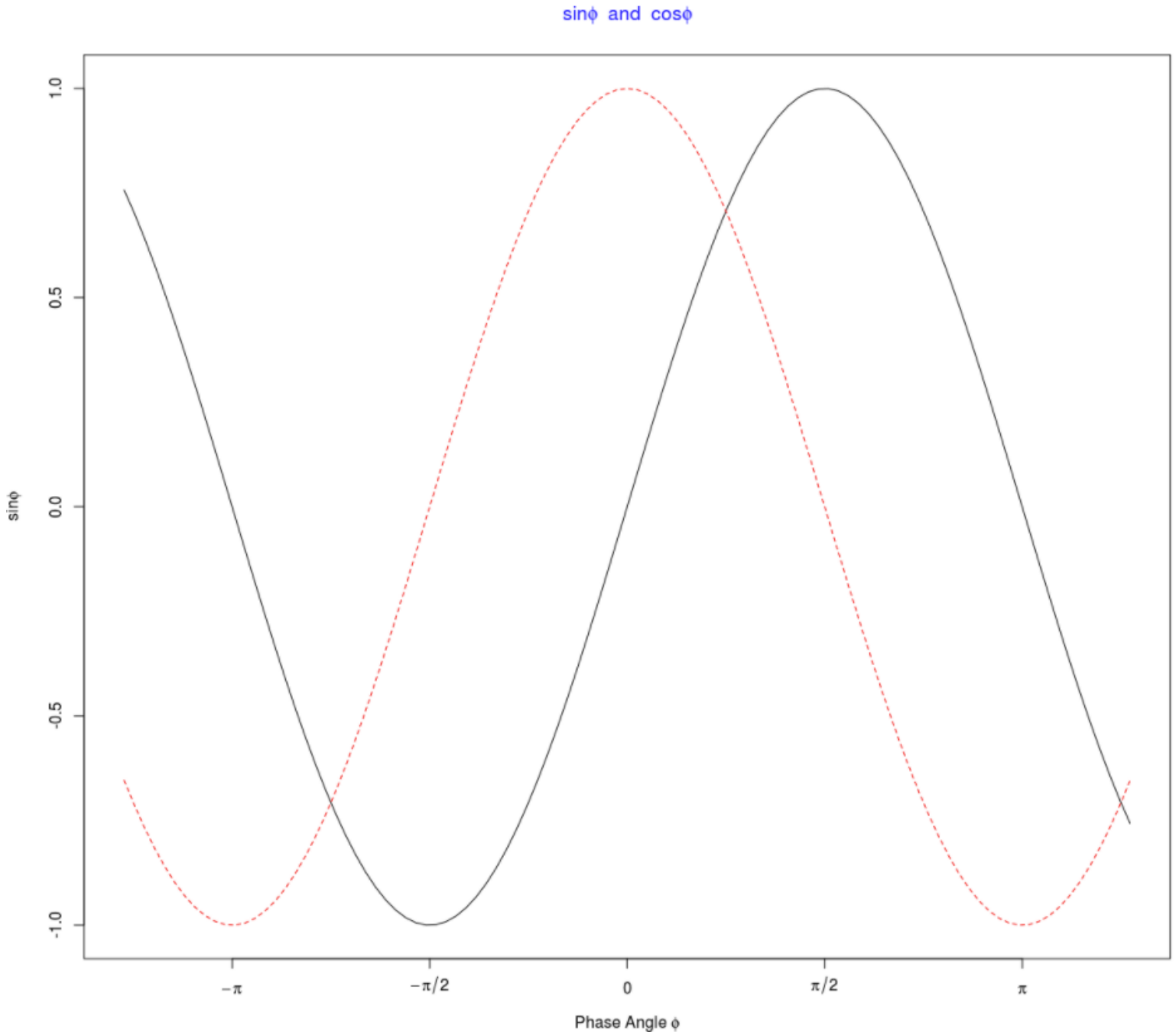
  - xlim、ylim：設定 x 軸與 y 軸的繪圖範圍。
  - log：指定需要對數轉換的座標軸，例如 log = "x" 就是將 x 軸經過對數轉換，而 log = "xy" 則是讓 x 與 y 軸都經過對數轉換。
  - main：指定圖形的標題，可指定為一般的字串或是以 expression 來表示的數學公式（詳細說明請參考 plotmath 的線上說明）。
  - xlab、ylab：指定 x 軸與 y 軸的名稱，其與 main 一樣可以指定為字串或數學公式。

Mathematical Annotation in R

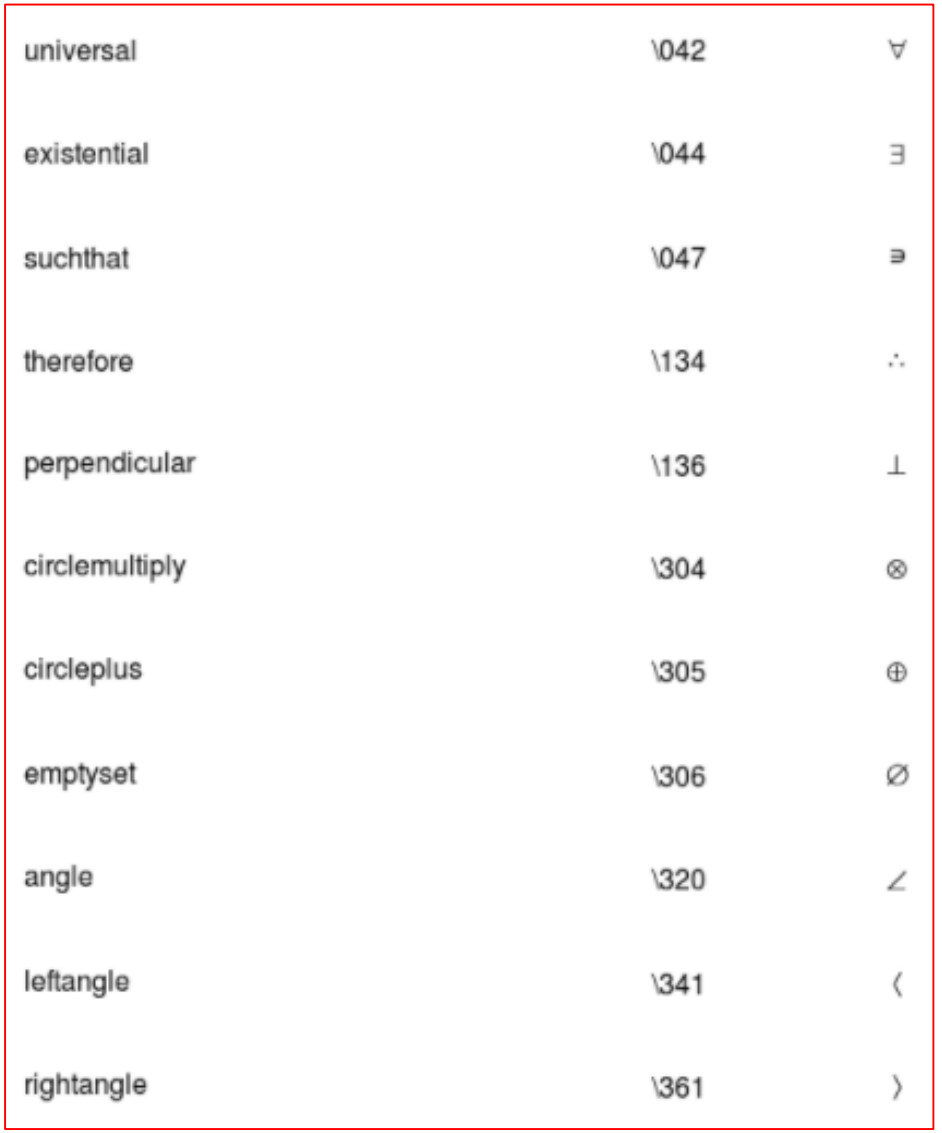
- plotmath

```
require(graphics)

x <- seq(-4, 4, len = 101)
y <- cbind(sin(x), cos(x))
matplot(x, y, type = "l", xaxt = "n",
        main = expression(paste(plain(sin) * phi, "  and  ",
                                plain(cos) * phi)),

        ylab = expression("sin" * phi, "cos" * phi), # only 1st is taken

        xlab = expression(paste("Phase Angle ", phi)),

        col.main = "blue")

axis(1, at = c(-pi, -pi/2, 0, pi/2, pi),
     labels = expression(-pi, -pi/2, 0, pi/2, pi))
```

https://stat.ethz.ch/R-manual/R-devel/library/grDevices/html/plotmath.html


sinϕ and cosϕ
1.0
0.5
0.0
-0.5
-1.0
sinϕ
−π
−π/2
0
π/2
π
Phase Angle ϕ

```
plot.new(); plot.window(c(0,4), c(15,1))
text(1, 1, "universal", adj = 0); text(2.5, 1,  "\\042")
text(3, 1, expression(symbol("\042")))
text(1, 2, "existential", adj = 0); text(2.5, 2,  "\\044")
text(3, 2, expression(symbol("\044")))
text(1, 3, "suchthat", adj = 0); text(2.5, 3,  "\\047")
text(3, 3, expression(symbol("\047")))
text(1, 4, "therefore", adj = 0); text(2.5, 4,  "\\134")
text(3, 4, expression(symbol("\134")))
text(1, 5, "perpendicular", adj = 0); text(2.5, 5,  "\\136")
text(3, 5, expression(symbol("\136")))
text(1, 6, "circlemultiply", adj = 0); text(2.5, 6,  "\\304")
text(3, 6, expression(symbol("\304")))
text(1, 7, "circleplus", adj = 0); text(2.5, 7,  "\\305")
text(3, 7, expression(symbol("\305")))
text(1, 8, "emptyset", adj = 0); text(2.5, 8,  "\\306")
text(3, 8, expression(symbol("\306")))
text(1, 9, "angle", adj = 0); text(2.5, 9,  "\\320")
text(3, 9, expression(symbol("\320")))
text(1, 10, "leftangle", adj = 0); text(2.5, 10,  "\\341")
text(3, 10, expression(symbol("\341")))
text(1, 11, "rightangle", adj = 0); text(2.5, 11,  "\\361")
text(3, 11, expression(symbol("\361")))
```



- qplot(carat, price, data = diamonds.subset, xlab = "Price ($)", ylab = "Weight (carats)", main = "Price-weight relationship")
- qplot(carat, price/carat, data = diamonds.subset, ylab = expression(frac(price,carat)), xlab = "Weight (carats)", main="Small diamonds", xlim = c(.2,1))
- qplot(carat, price, data = diamonds.subset, log = "xy")