# 第 13 天

## 探索資料分析（2）

除了內建的 **Base Plotting System** 可以繪製圖形，有很高比例的 R 語言使用者更依賴使用 ggplot2 這個繪圖套件，它簡潔、彈性高以及美觀的輸出，是吸引這些使用者的原因。gg 意指 **grammer of graphics**，核心理念是利用正規而有結構的文法來探索資料，它的作者是 [Hadley Wickham](http://hadley.nz/) 與 Winston Chang。

### 安裝 ggplot2 套件

我們先來複習一下怎麼進行安裝套件，透過命令列（Console）安裝的方法是輸入：

> install.packages("ggplot2")

透過 RStudio 介面安裝的步驟是在右下角的 **packages** 頁籤點選 **install**：

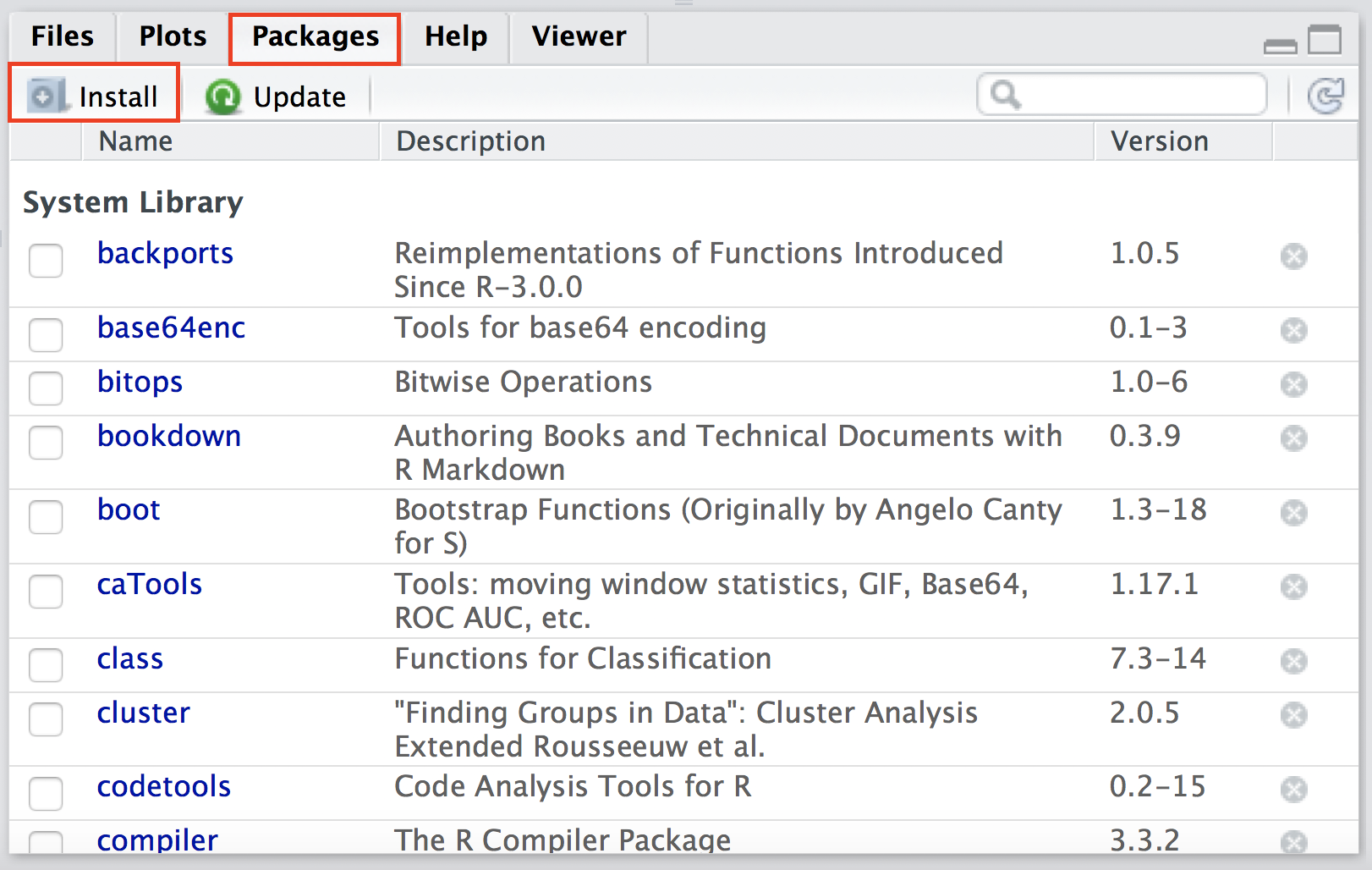


圖 13-1 點選 install

接著是輸入安裝套件的名稱：ggplot2。

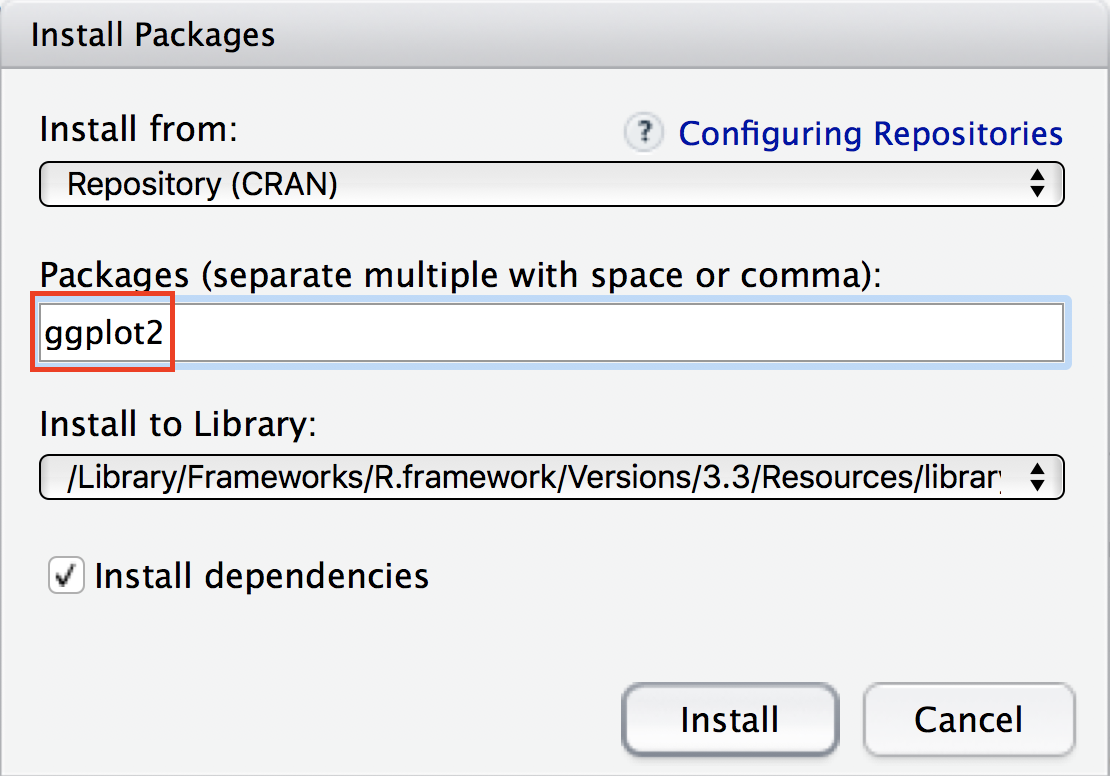


圖 13-2 輸入 ggplot2

### 載入 ggplot2 套件

接著我們複習怎麼載入套件，透過命令列（Console）載入的方法是輸入：

library(ggplot2)

透過 RStudio 介面載入的方法是在右下角的 **packages** 頁籤下找到 ggplot2 然後將前面的核取方框打勾即可。

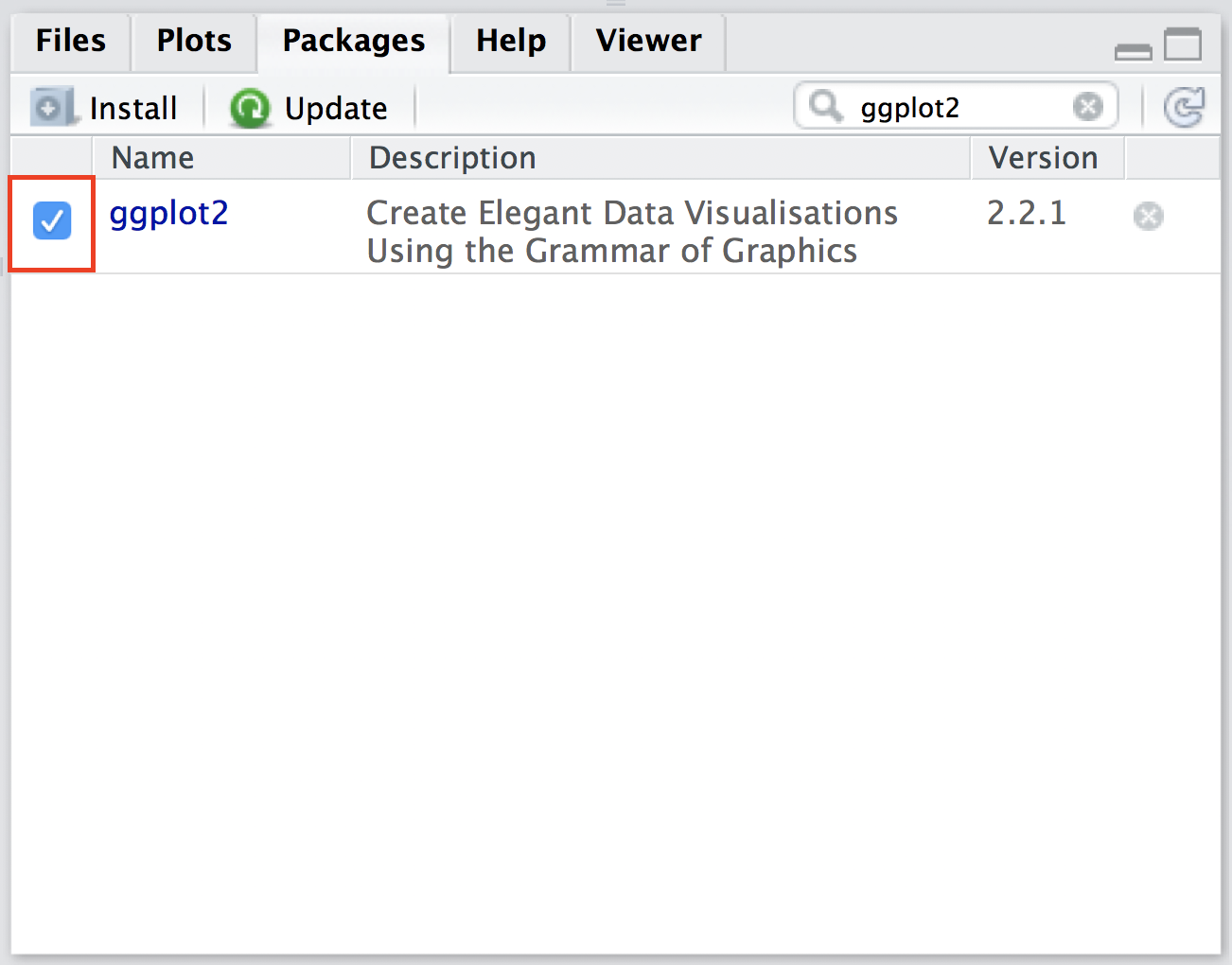


圖 13-3 載入 ggplot2

### 一個簡單的 ggplot2 散佈圖

成功安裝且載入 ggplot2 套件以後，我們用一個簡單的散佈圖藉此暸解它的語法邏輯：

> library(ggplot2)  
  
> ggplot(cars, aes(x = speed, y = dist))

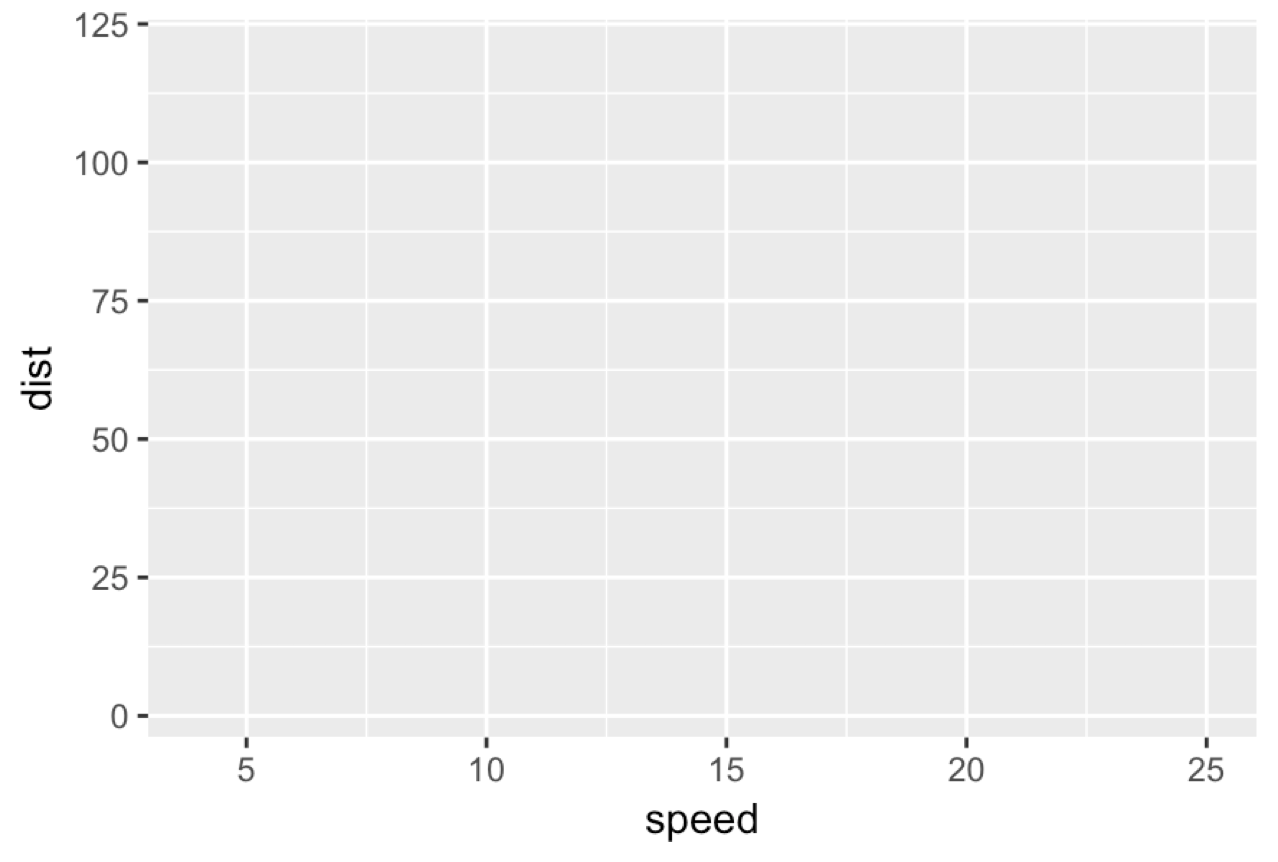


圖 13-4 簡單的散佈圖

這段程式會建立出一個 ggplot 的物件並且綁定 cars 的兩個變數到這個物件之上，這裡我們可以觀察到 ggplot2 繪圖來源是資料框，素材是資料框中的變數，aes() 則負責將素材綁定至 X 軸與 Y 軸，aes 是 **aesthetic mappings** 的縮寫。

接下來我們還要告訴 ggplot 物件要用什麼樣的形式呈現這個繪圖，這時會利用 geom\_() 指定繪圖的形式，geom 是 **geometric objects** 的縮寫，底線後面則會加上樣式，以我們現在繪製簡單的散佈圖需求而言，使用 geom\_point() 就可以加上點的樣式：

> library(ggplot2)  
  
> ggplot(cars, aes(x = speed, y = dist)) + geom\_point()

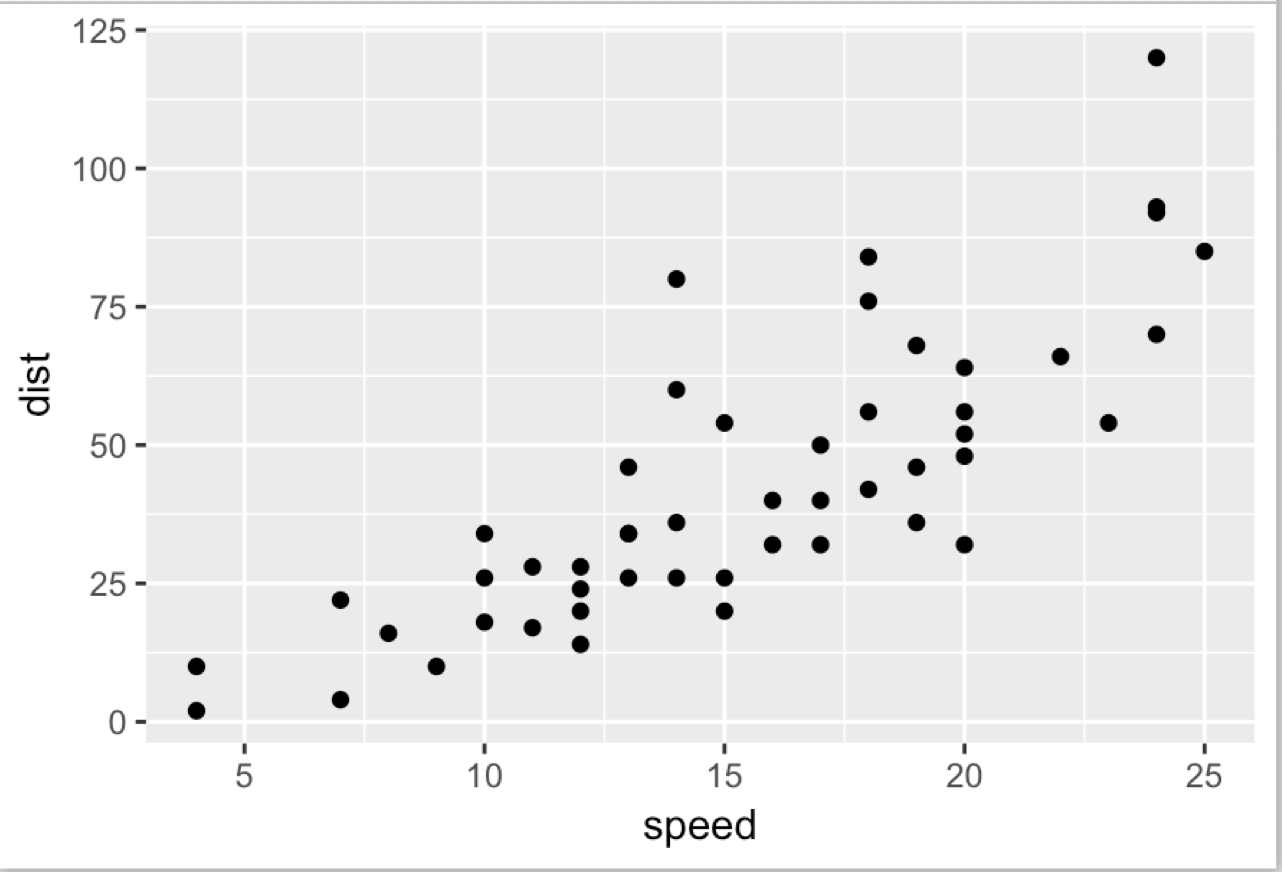


圖 13-5 簡單的散佈圖（2）

### ggplot2 繪圖

在開始檢視不同樣式的資料探索圖形之前，我們用下表對這些 geom\_() 有一個概觀：

|  |  |
| --- | --- |
| geom\_ | 圖形 |
| geom\_histogram() | 直方圖 |
| geom\_boxplot() | 盒鬚圖 |
| geom\_line() | 線圖 |
| geom\_point() | 散佈圖 |
| geom\_bar() | 長條圖 |
| stat\_function(fun, geom = "line") | 曲線圖 |

#### 探索數值分佈

由於 ggplot2 的繪圖來源是一個資料框，讓我們將 rnorm(1000) 儲存為一個 hist\_df 資料框中的 norm\_nums 變數，再使用 geom\_histogram() 繪製直方圖來探索數值的分佈：

> library(ggplot2)  
>   
> set.seed(123)  
> norm\_nums <- rnorm(1000)  
> hist\_df <- data.frame(norm\_nums = norm\_nums)  
> ggplot(hist\_df, aes(x = norm\_nums)) + geom\_histogram()  
`stat\_bin()` using `bins = 30`. Pick better value with  
`binwidth`.

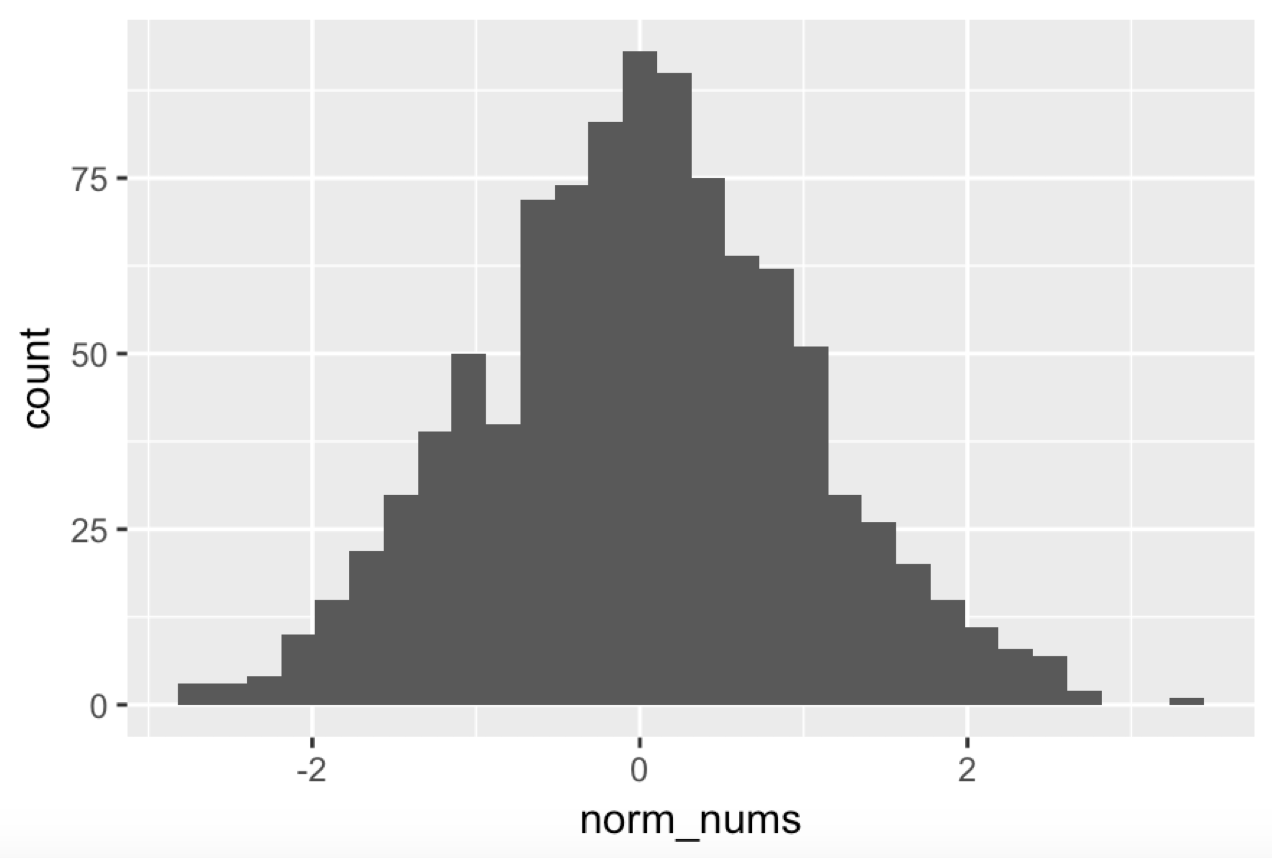


圖 13-6 ggplot2 直方圖

由於直方圖的**分箱數（bins）**使用了 30 個分箱這個預設值，我們可以透過調整 binwidth 參數來更改，適當增加 binwidth 會減少分箱數，反之減少 binwidth 會增加分箱數，至於增加與減少的幅度，視資料的範圍斟酌：

> library(ggplot2)  
>   
> set.seed(123)  
> norm\_nums <- rnorm(1000)  
> hist\_df <- data.frame(norm\_nums = norm\_nums)  
> ggplot(hist\_df, aes(x = norm\_nums)) + geom\_histogram(binwidth = 0.1) # 比預設的分箱數多  
> ggplot(hist\_df, aes(x = norm\_nums)) + geom\_histogram(binwidth = 0.5) # 比預設的分箱數少

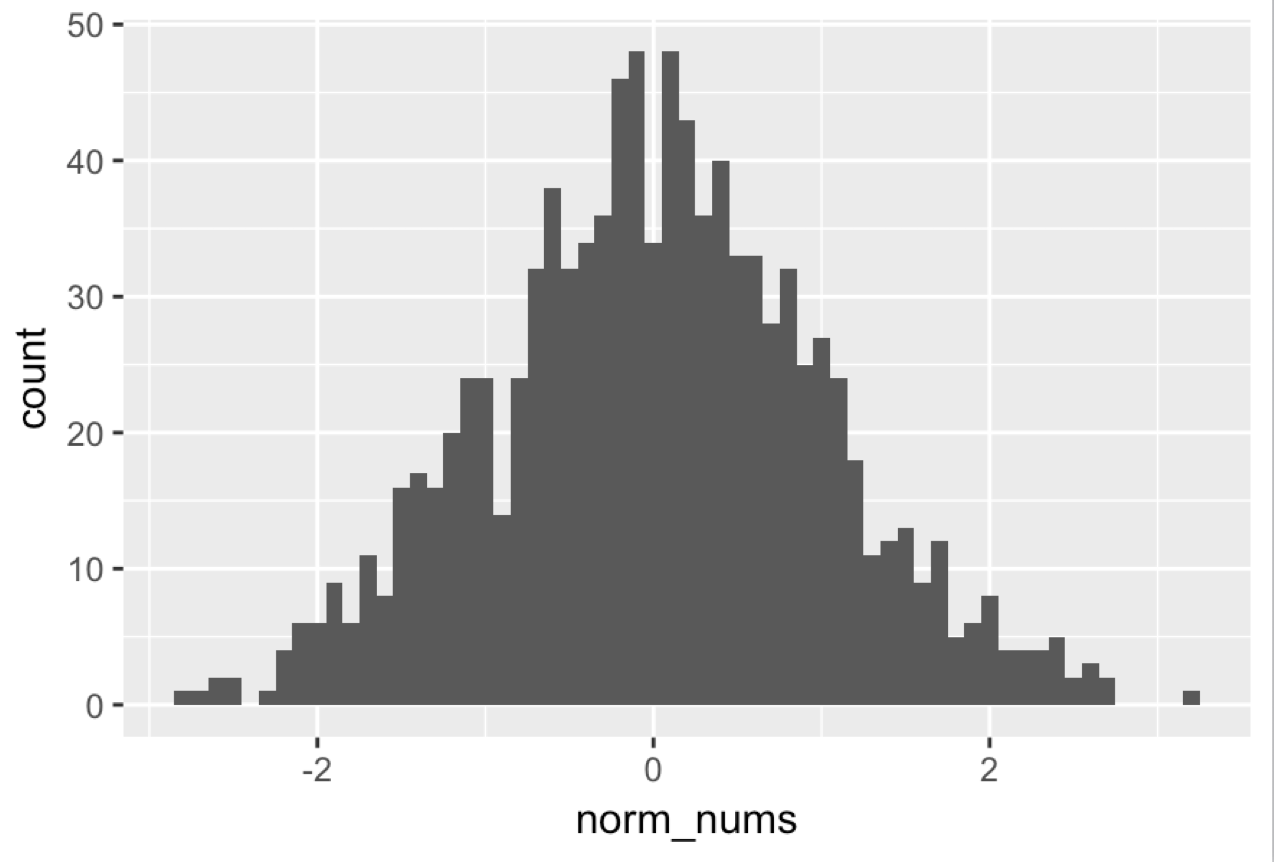


圖 13-7 ggplot2 直方圖（2）

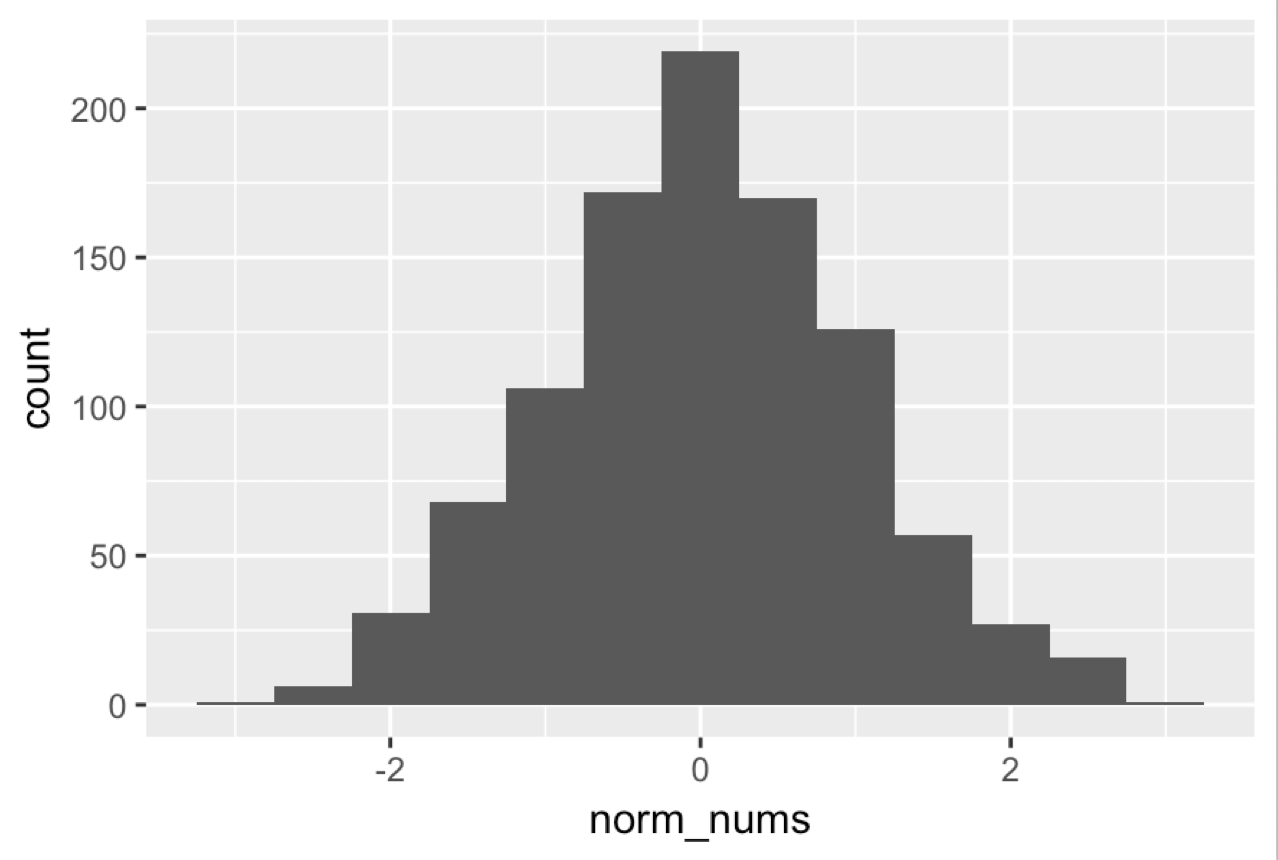


圖 13-8 ggplot2 直方圖（3）

#### 探索不同類別與數值分佈的關係

使用 geom\_boxplot() 繪製盒鬚圖來探索不同類別與數值分佈的關係，我們探索內建資料 iris 中不同鳶尾花品種（Species 變數）的萼片長度（Sepal.Length 變數）分佈差異：

> library(ggplot2)  
  
> ggplot(iris, aes(x = Species, y = Sepal.Length)) + geom\_boxplot()

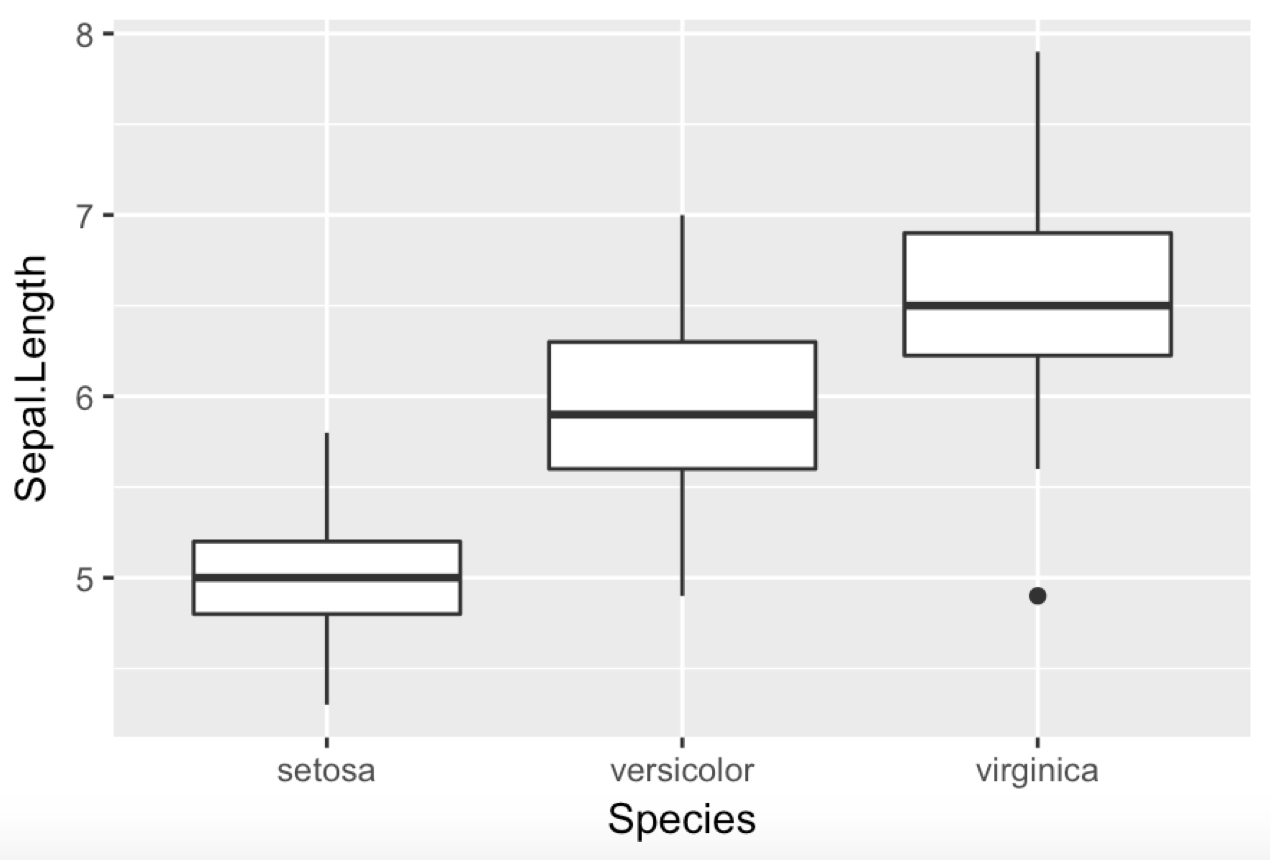


圖 13-9 ggplot2 盒鬚圖

#### 探索數值與日期（時間）的關係

使用 geom\_line() 繪製線圖來探索數值與日期（時間）的關係：

> library(ggplot2)  
>  
> x <- seq(from = as.Date("2017-01-01"), to = as.Date("2017-01-31"), by = 1)  
> set.seed(123) # 確保我們得到一樣的 y  
> y <- sample(1:100, size = 31, replace = TRUE)  
> line\_df <- data.frame(x = x, y = y)  
> ggplot(line\_df, aes(x = x, y = y)) + geom\_line()

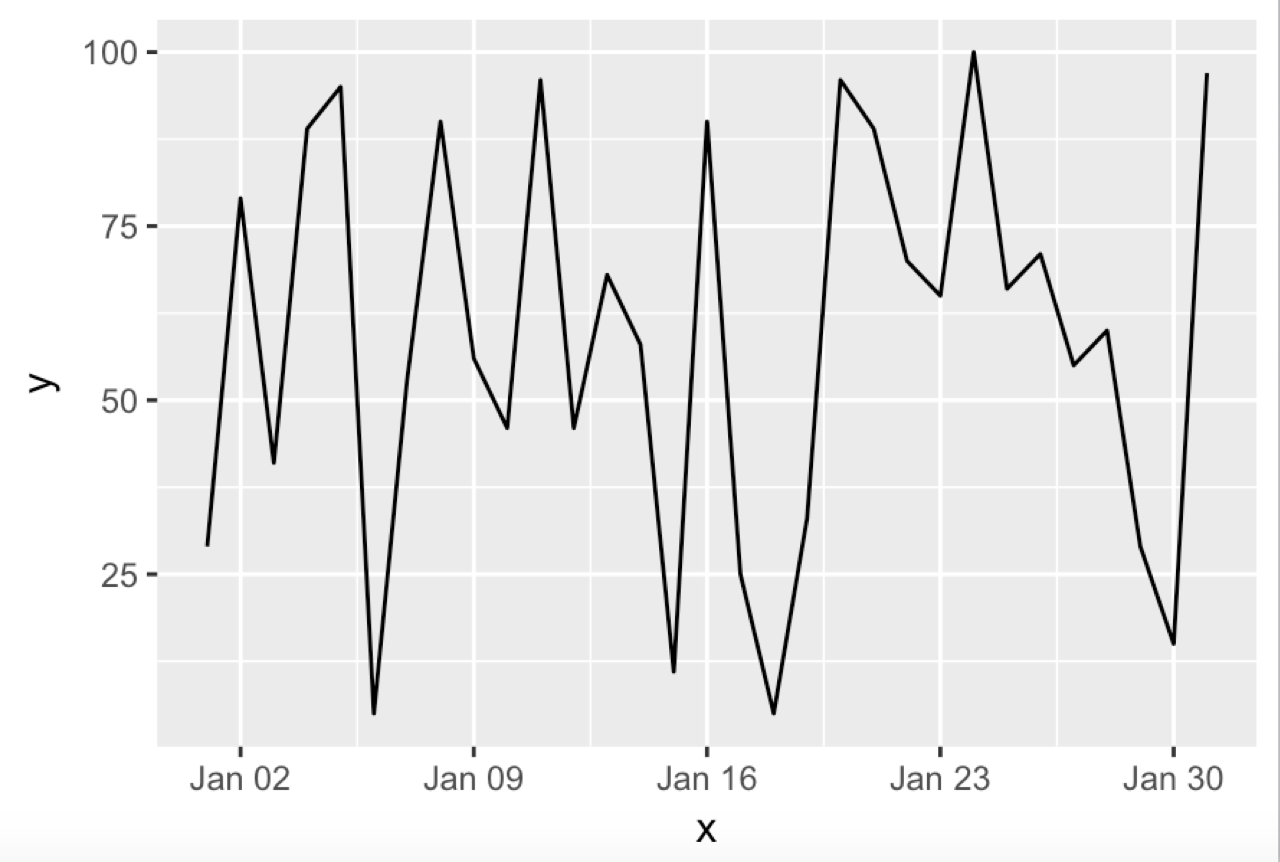


圖 13-10 ggplot2 線圖

我們可以透過增加 scale\_x\_date(date\_labels) 調整日期顯示的格式，預設是使用 %b %d 這個顯示格式（月份縮寫與日，中間以空格隔開），關於這個縮寫代表的意義可以參考[第 04 天：變數類型的判斷與轉換](chapter4.md)。在下面的程式我們改為使用 %m.%d 月份與日都以數字表示，中間以 . 隔開：

> library(ggplot2)  
>  
> x <- seq(from = as.Date("2017-01-01"), to = as.Date("2017-01-31"), by = 1)  
> set.seed(123) # 確保我們得到一樣的 y  
> y <- sample(1:100, size = 31, replace = TRUE)  
> line\_df <- data.frame(x = x, y = y)  
> ggplot(line\_df, aes(x = x, y = y)) + geom\_line() + scale\_x\_date(date\_labels = "%m.%d")

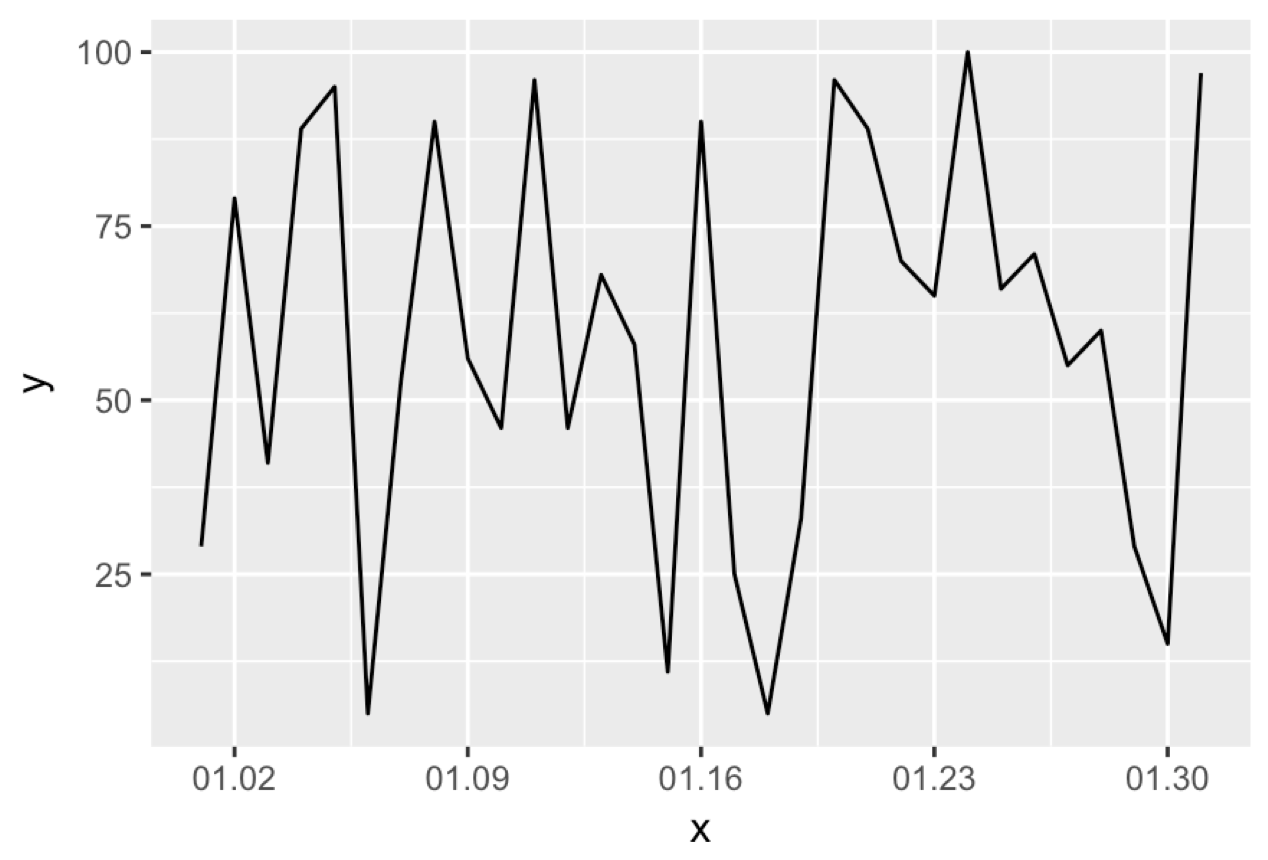


圖 13-11 ggplot2 線圖（2）

#### 探索兩個數值的關係

使用 geom\_point() 繪製散佈圖來探索兩個數值的關係：

> library(ggplot2)  
  
> ggplot(cars, aes(x = speed, y = dist)) + geom\_point()

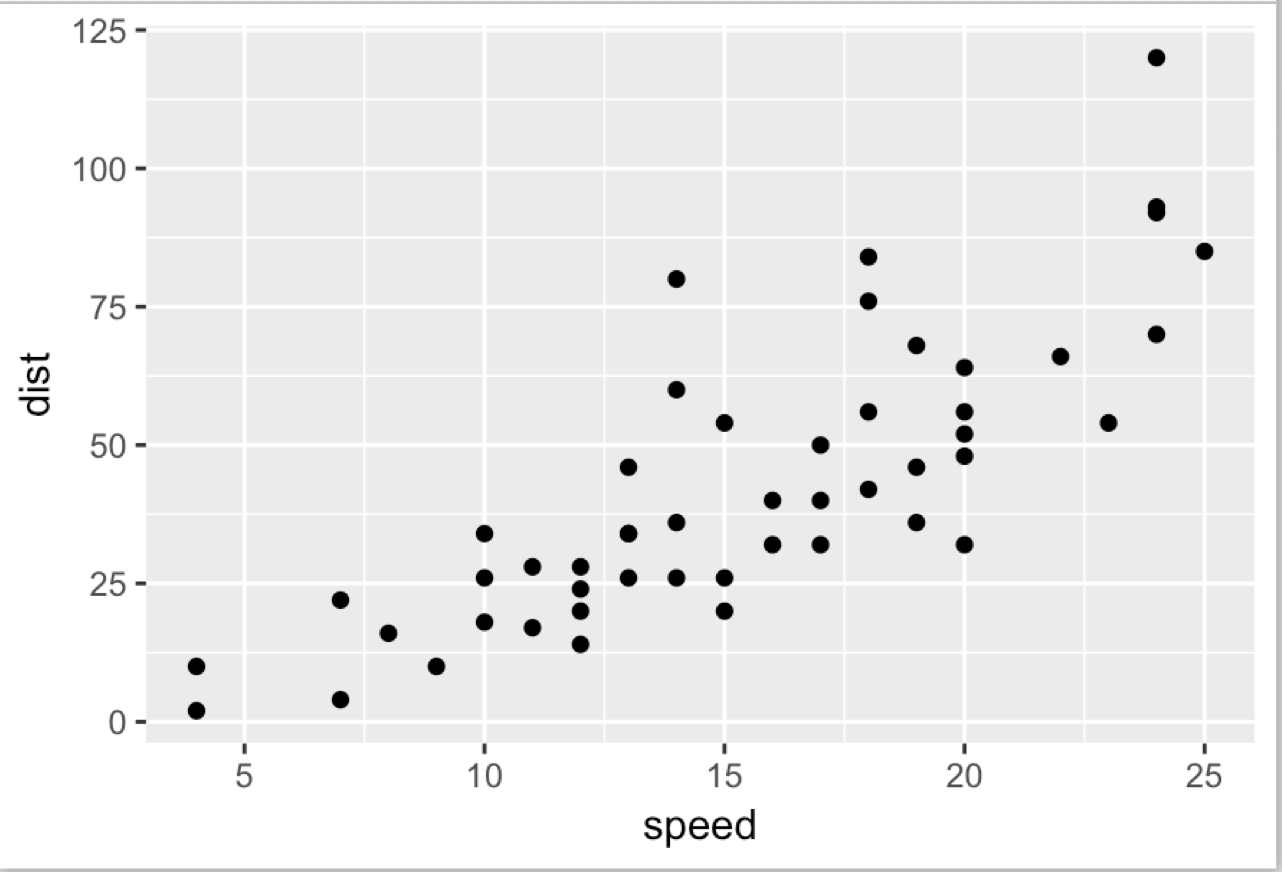


圖 13-12 ggplot2 散佈圖

#### 探索類別

使用 geom\_bar() 繪製長條圖來探索類別的分佈，例如 ice\_cream\_df 資料框紀錄了 100 個人最喜歡的冰淇淋口味：

> library(ggplot2)  
  
> ice\_cream\_flavor <- rep(NA, times = 100)  
> for (i in 1:100){  
+ ice\_cream\_flavor[i] <- sample(c("vanilla", "chocolate", "matcha", "other"), size = 1)  
+ }  
> ice\_cream\_df <- data.frame(ice\_cream\_flavor = ice\_cream\_flavor)  
> ggplot(ice\_cream\_df, aes(x = ice\_cream\_flavor)) + geom\_bar()

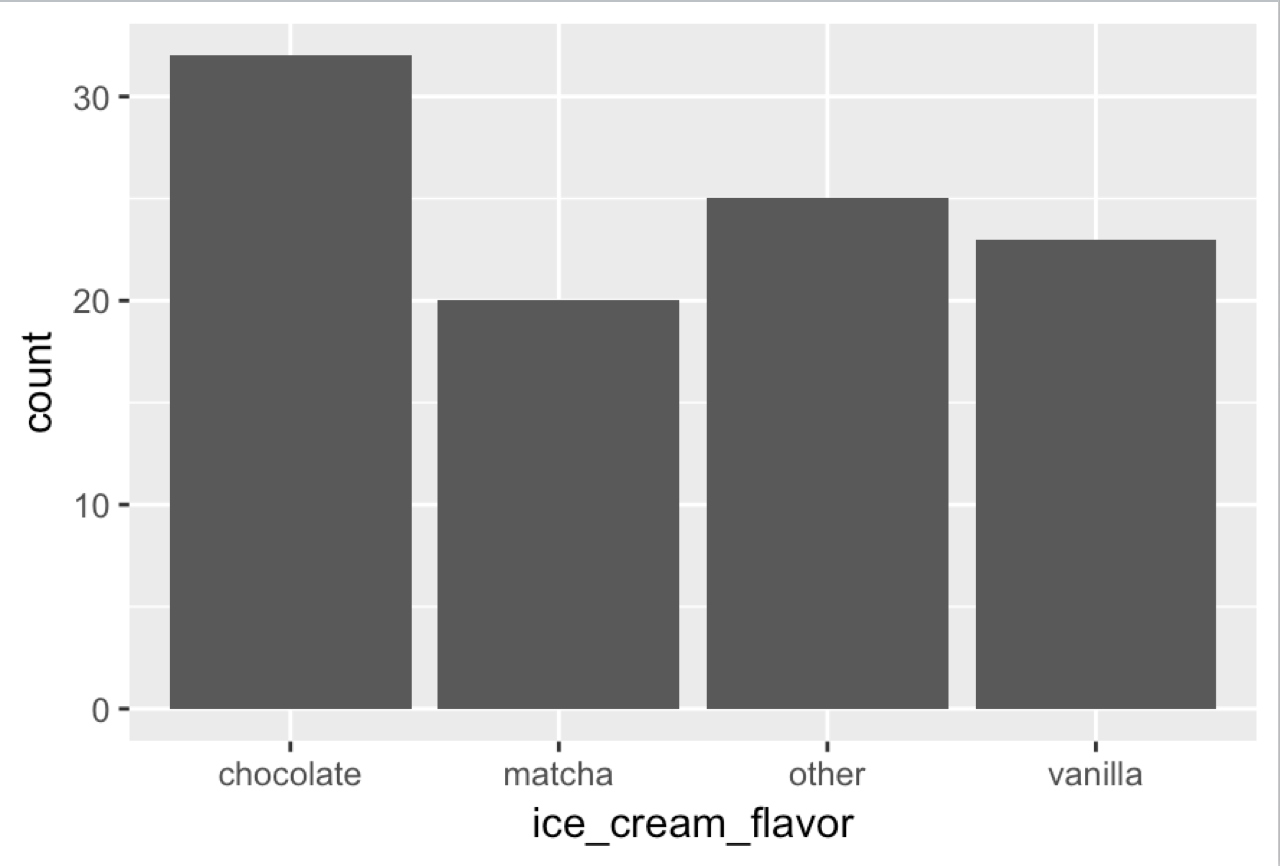


圖 13-13 ggplot2 長條圖

與 **Base Plotting System** 最大的不同在於，我們不需要再使用 table() 函數先統計各種口味的人數才能繪圖，ggplot2 針對類別變數自動會分別加總。

假如我們的資料本身是已經統計過的資訊，比如長得像這樣：

> ice\_cream\_flavor <- rep(NA, times = 100)  
> for (i in 1:100){  
+ ice\_cream\_flavor[i] <- sample(c("vanilla", "chocolate", "matcha", "other"), size = 1)  
+ }  
> flavor <- names(table(ice\_cream\_flavor))  
> votes <- as.vector(unname(table(ice\_cream\_flavor)))  
> ice\_cream\_df <- data.frame(flavor = flavor, votes = votes)  
> ice\_cream\_df  
 flavor votes  
1 chocolate 32  
2 matcha 20  
3 other 25  
4 vanilla 23

這時繪製長條圖就必須要指定一個參數 stat = "identity"：

> library(ggplot2)  
>  
> ice\_cream\_flavor <- rep(NA, times = 100)  
> for (i in 1:100){  
+ ice\_cream\_flavor[i] <- sample(c("vanilla", "chocolate", "matcha", "other"), size = 1)  
+ }  
> flavor <- names(table(ice\_cream\_flavor))  
> votes <- as.vector(unname(table(ice\_cream\_flavor)))  
> ice\_cream\_df <- data.frame(flavor = flavor, votes = votes)  
> ggplot(ice\_cream\_df, aes(x = flavor, y = votes)) + geom\_bar(stat = "identity")

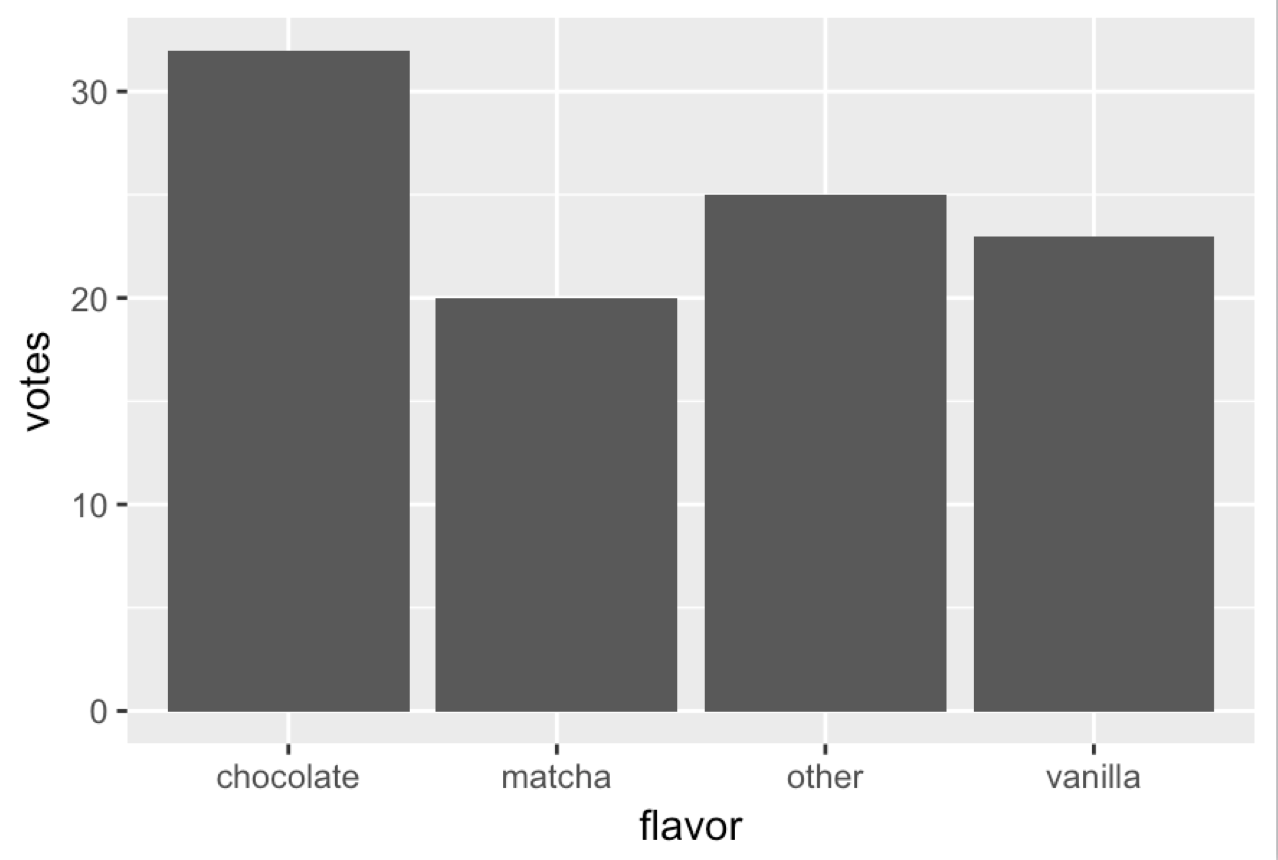


圖 13-14 ggplot2 長條圖（2）

不然會出現這樣的錯誤訊息：

> library(ggplot2)  
>  
> ice\_cream\_flavor <- rep(NA, times = 100)  
> for (i in 1:100){  
+ ice\_cream\_flavor[i] <- sample(c("vanilla", "chocolate", "matcha", "other"), size = 1)  
+ }  
> flavor <- names(table(ice\_cream\_flavor))  
> votes <- as.vector(unname(table(ice\_cream\_flavor)))  
> ice\_cream\_df <- data.frame(flavor = flavor, votes = votes)  
> ggplot(ice\_cream\_df, aes(x = flavor, y = votes)) + geom\_bar()  
Error: stat\_count() must not be used with a y aesthetic.

這是因為 ggplot2 的 geom\_bar() 預設會協助我們計算類別變數中的相異個數：stat = "count"，也就是 table() 函數做的事情，假如長條的高度已經在資料框中被計算好，那麼就用 stat = "identity" 就可以了，這是很多初接觸 ggplot2 的使用者會感到困惑的地方，值得我們花一些篇幅解釋。

#### 繪畫函數

使用 stat\_function(fun, geom = "line") 將函數繪製出來，我們需要在資料框中設定輸入的值域，舉例來說我們可以將 sin() 函數在

與

之間的輸出描繪出來：

> library(ggplot2)  
  
> sin\_df <- data.frame(x = c(-pi, pi))  
> sin\_df  
 x  
1 -3.141593  
2 3.141593  
> ggplot(sin\_df, aes(x = x)) + stat\_function(fun = sin, geom = "line")

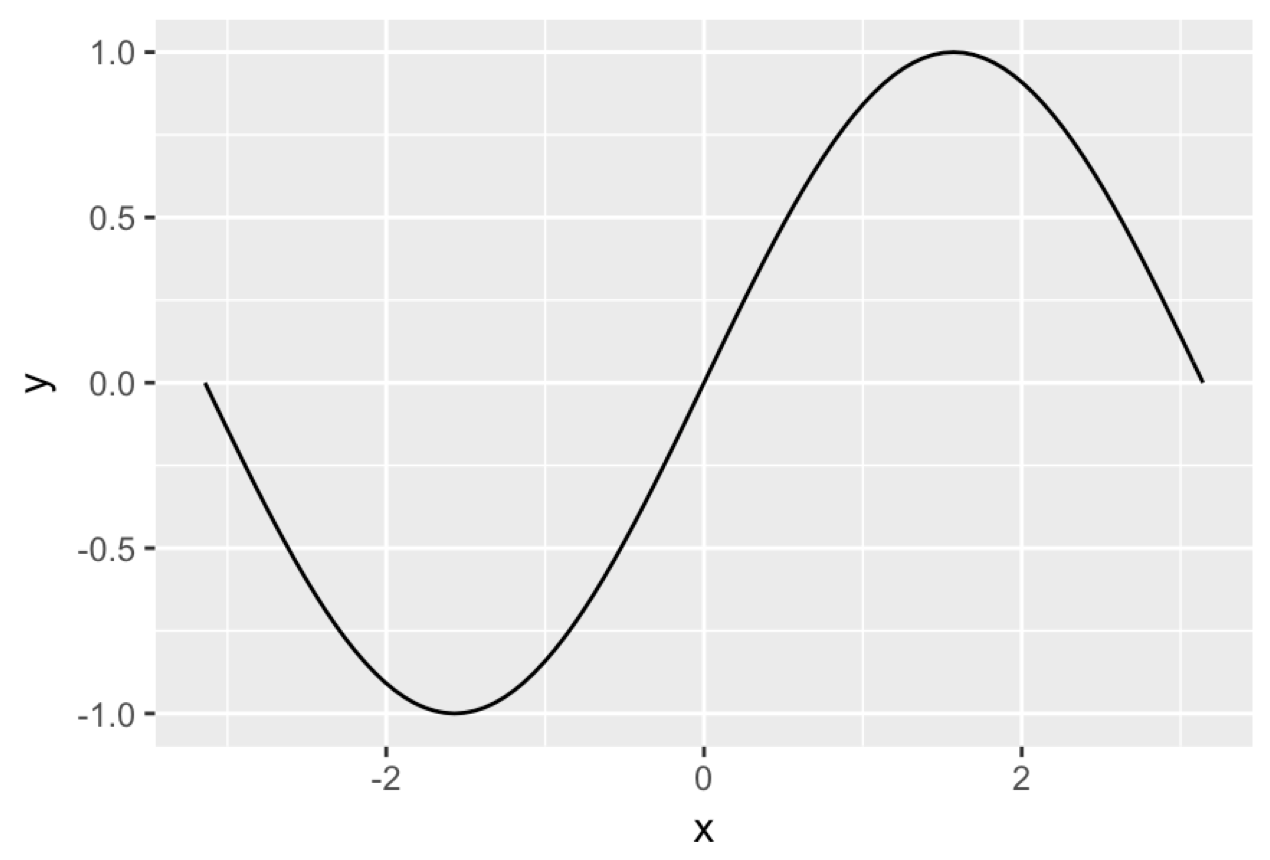


圖 13-15 ggplot2 繪畫函數

如果是一個自訂函數 my\_sqr()，很簡單的

，我們可以將它在 -3 與 3 之間的輸出描繪出來：

> library(ggplot2)  
>   
> my\_sqr <- function(x){  
+ return(x^2)  
+ }  
> my\_sqr\_df <- data.frame(x = c(-3, 3))  
> ggplot(my\_sqr\_df, aes(x = x)) + stat\_function(fun = my\_sqr, geom = "line")



圖 13-16 ggplot2 繪畫函數（2）

#### 常用的自訂元素

##### 自訂標題、X 軸標籤與 Y 軸標籤

加入 ggtitle()、xlab() 與 ylab() 分別指定：

> library(ggplot2)  
  
> ggplot(cars, aes(x = speed, y = dist)) +  
+ geom\_point() +   
+ ggtitle("Car speed vs. braking distance") +  
+ xlab("Car speed(mph)") +   
+ ylab("Braking distance(ft)")

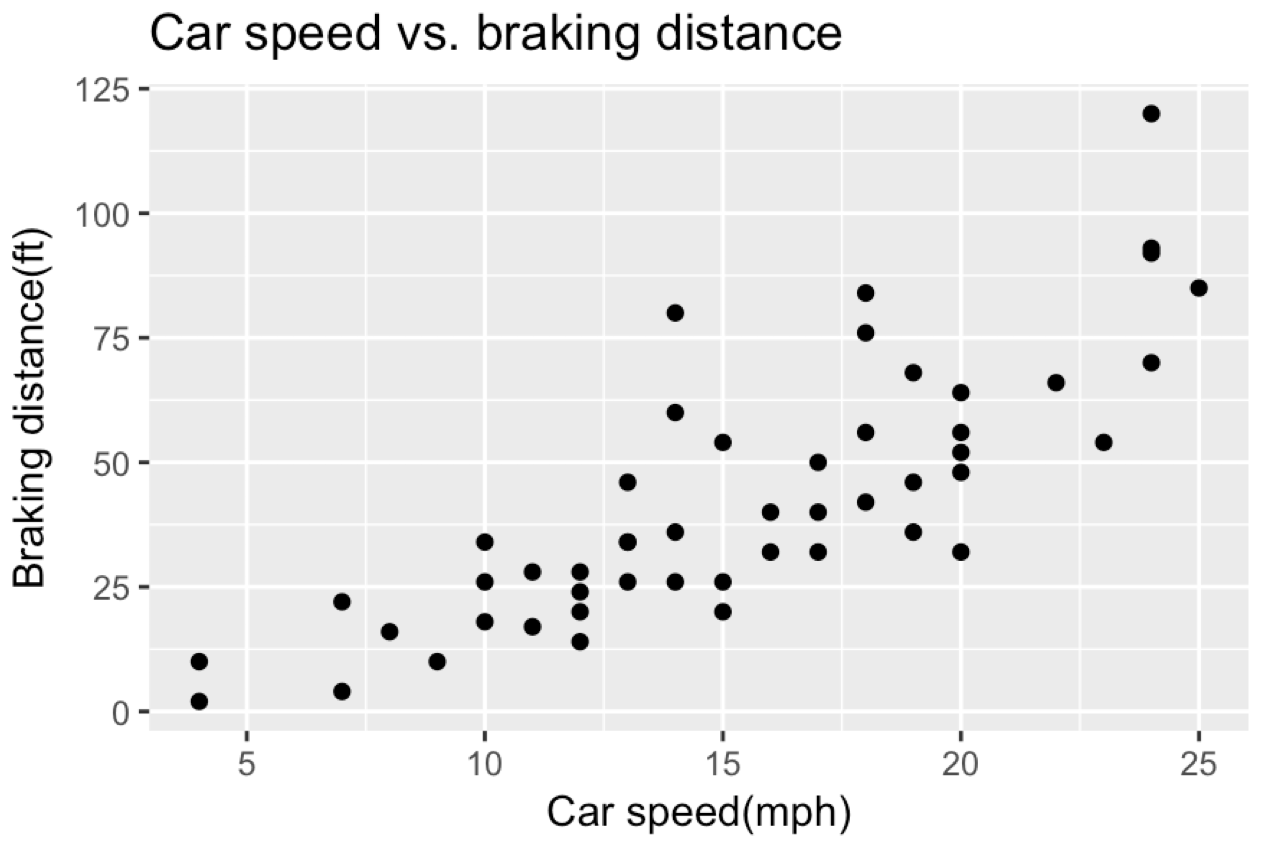


圖 13-17 ggplot2 自訂標題與座標軸標籤

##### 隱藏格線

ggplot2 繪製的圖形預設都有格線，我們可以加入 theme() 進行非常細緻的格線隱藏設定：

|  |  |
| --- | --- |
| 目的 | 設定 |
| 隱藏主要格線 | panel.grid.major = element\_blank() |
| 隱藏次要格線 | panel.grid.minor = element\_blank() |
| 隱藏 X 軸主要格線 | panel.grid.major.x = element\_blank() |
| 隱藏 Y 軸主要格線 | panel.grid.major.y = element\_blank() |
| 隱藏 X 軸次要格線 | panel.grid.minor.x = element\_blank() |
| 隱藏 Y 軸次要格線 | panel.grid.minor.y = element\_blank() |

隱藏主要與次要格線：

> library(ggplot2)  
  
> ggplot(cars, aes(x = speed, y = dist)) +  
+ geom\_point() +  
+ theme(panel.grid.major = element\_blank(),   
+ panel.grid.minor = element\_blank())

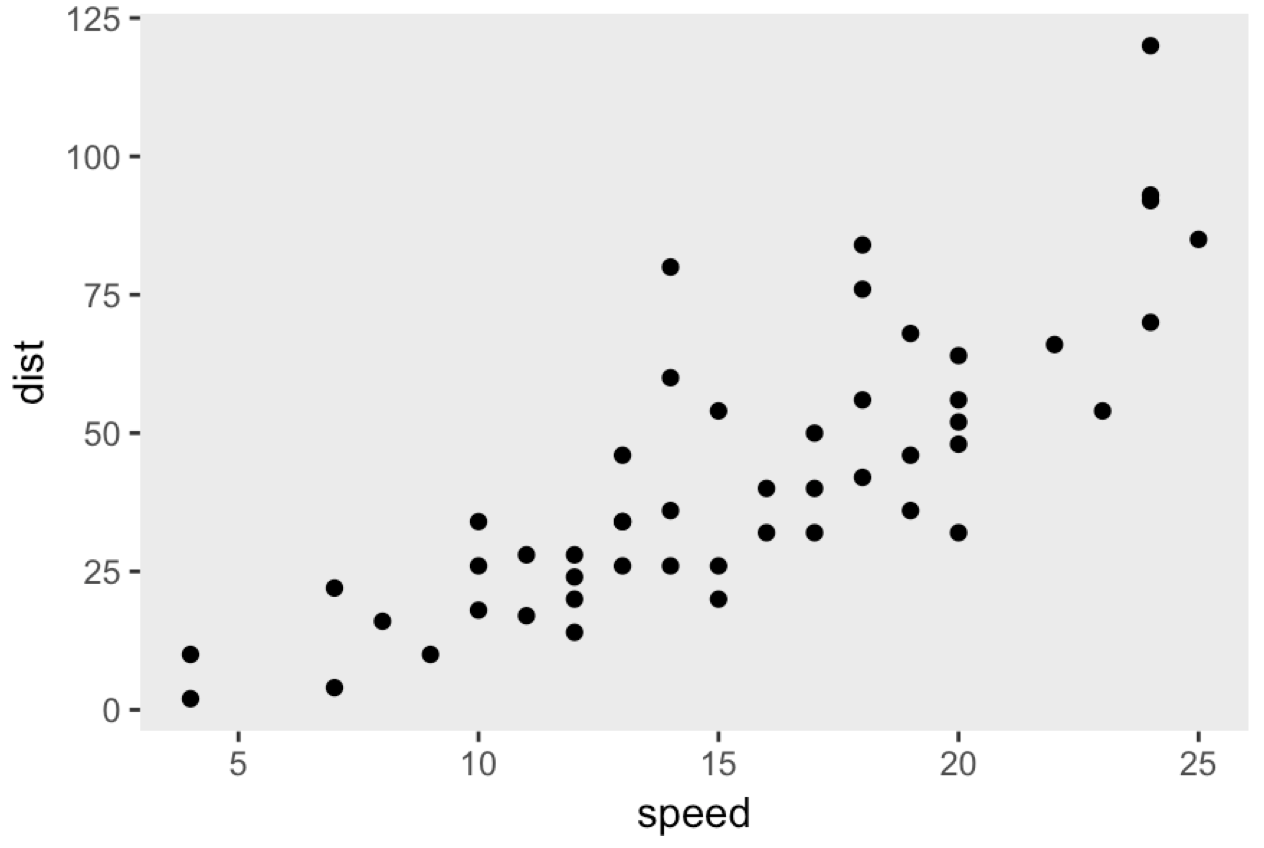


圖 13-18 ggplot2 隱藏格線

##### 調整圖形為水平方向

加入 coord\_flip()，這是 **coordinate flip** 的縮寫。

> library(ggplot2)  
  
> ice\_cream\_flavor <- rep(NA, times = 100)  
> for (i in 1:100){  
+ ice\_cream\_flavor[i] <- sample(c("vanilla", "chocolate", "matcha", "other"), size = 1)  
+ }  
> ice\_cream\_df <- data.frame(ice\_cream\_flavor = ice\_cream\_flavor)  
> ggplot(ice\_cream\_df, aes(x = ice\_cream\_flavor)) +  
+ geom\_bar() +  
+ coord\_flip()

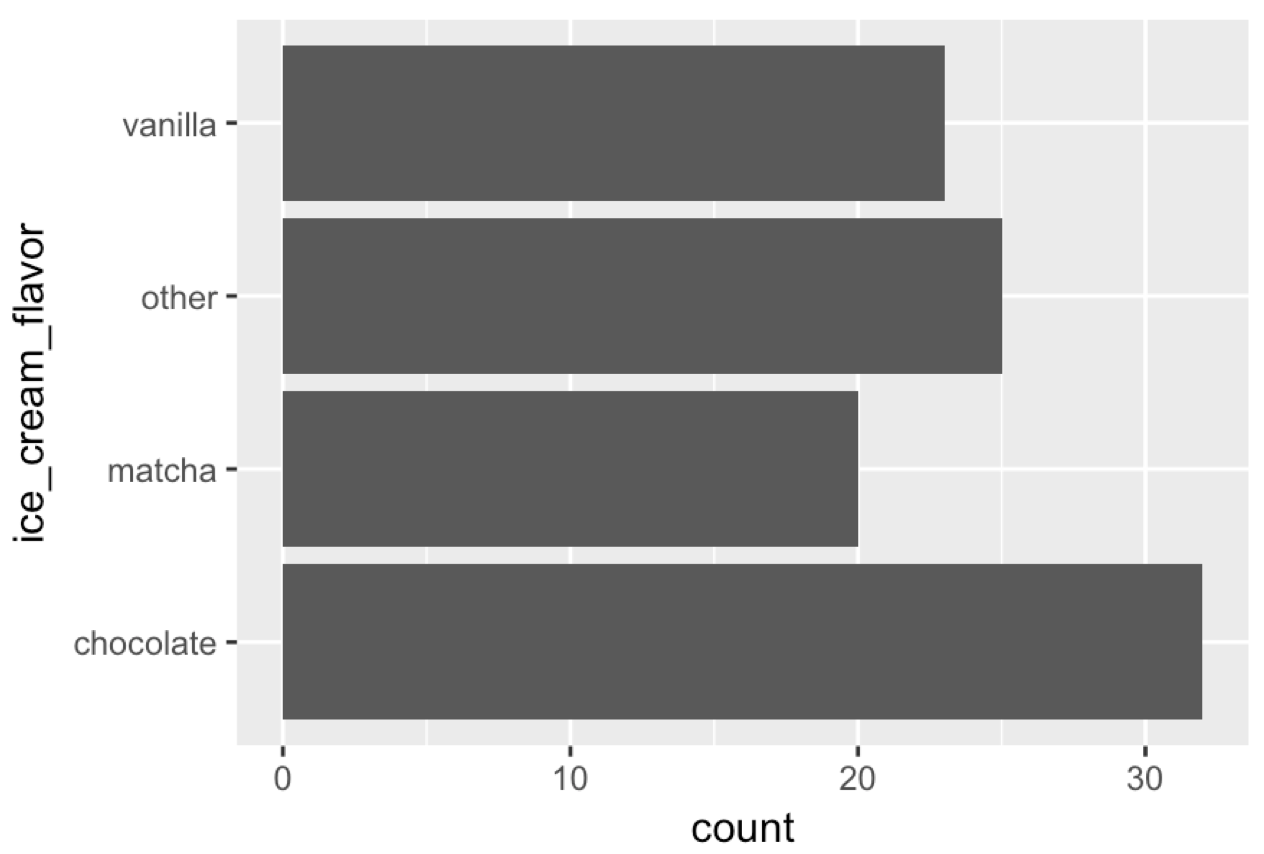


圖 13-19 ggplot2 水平的長條圖

##### 在直方圖上加上密度曲線

在 geom\_histogram() 設定中加入 aes(y = ..density..) 以及加入 geom\_density()。

> library(ggplot2)  
>   
> set.seed(123)  
> norm\_nums <- rnorm(1000)  
> hist\_df <- data.frame(norm\_nums = norm\_nums)  
> ggplot(hist\_df, aes(x = norm\_nums)) +  
 geom\_histogram(binwidth = 0.5, aes(y = ..density..), alpha= 0.5) +  
 geom\_density()

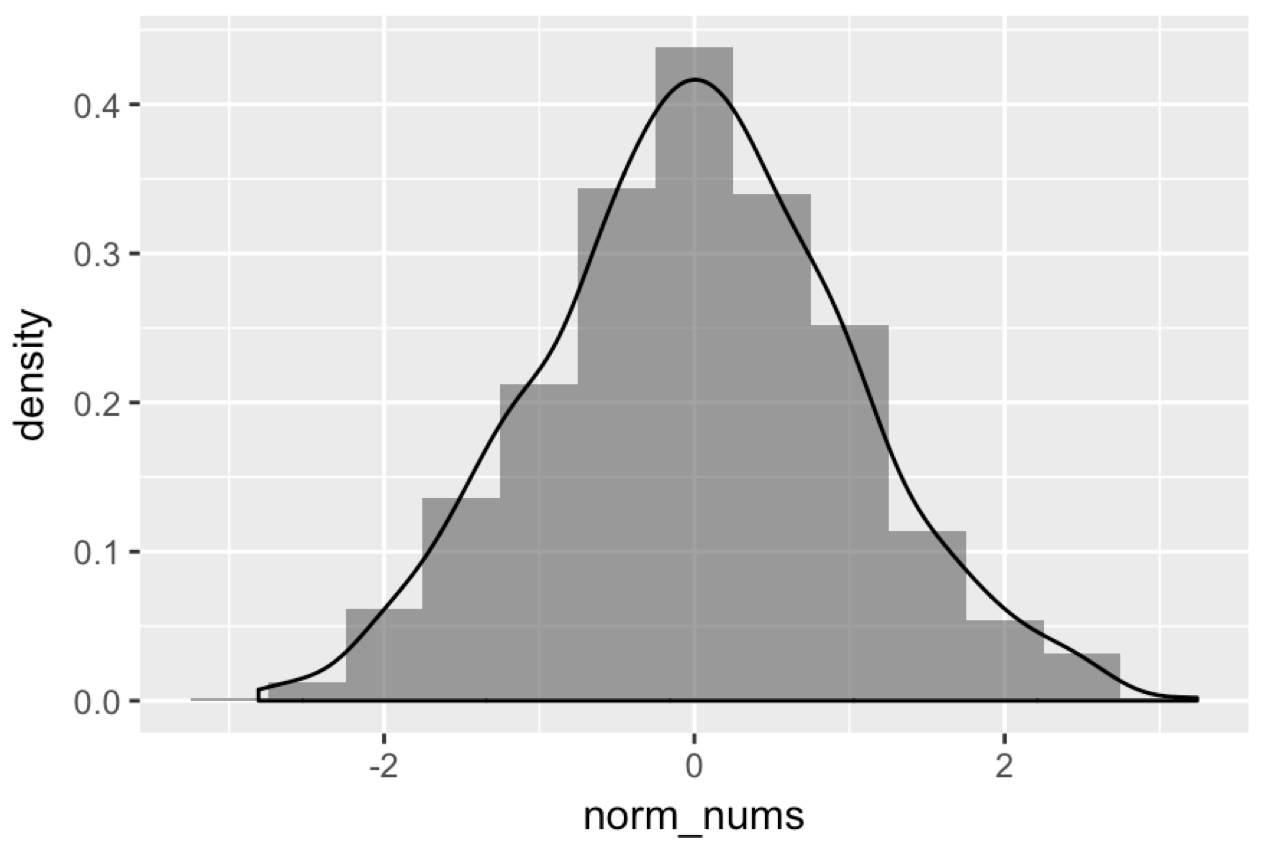


圖 13-20 ggplot2 直方圖加上密度曲線

##### 調整資料點的形狀與顏色

在 geom\_point() 中加入 shape 與 colour 的設定。

> library(ggplot2)  
  
> ggplot(cars, aes(x = speed, y = dist)) +  
+ geom\_point(shape = 2, colour = "red") # 資料點改成紅色的空心三角形

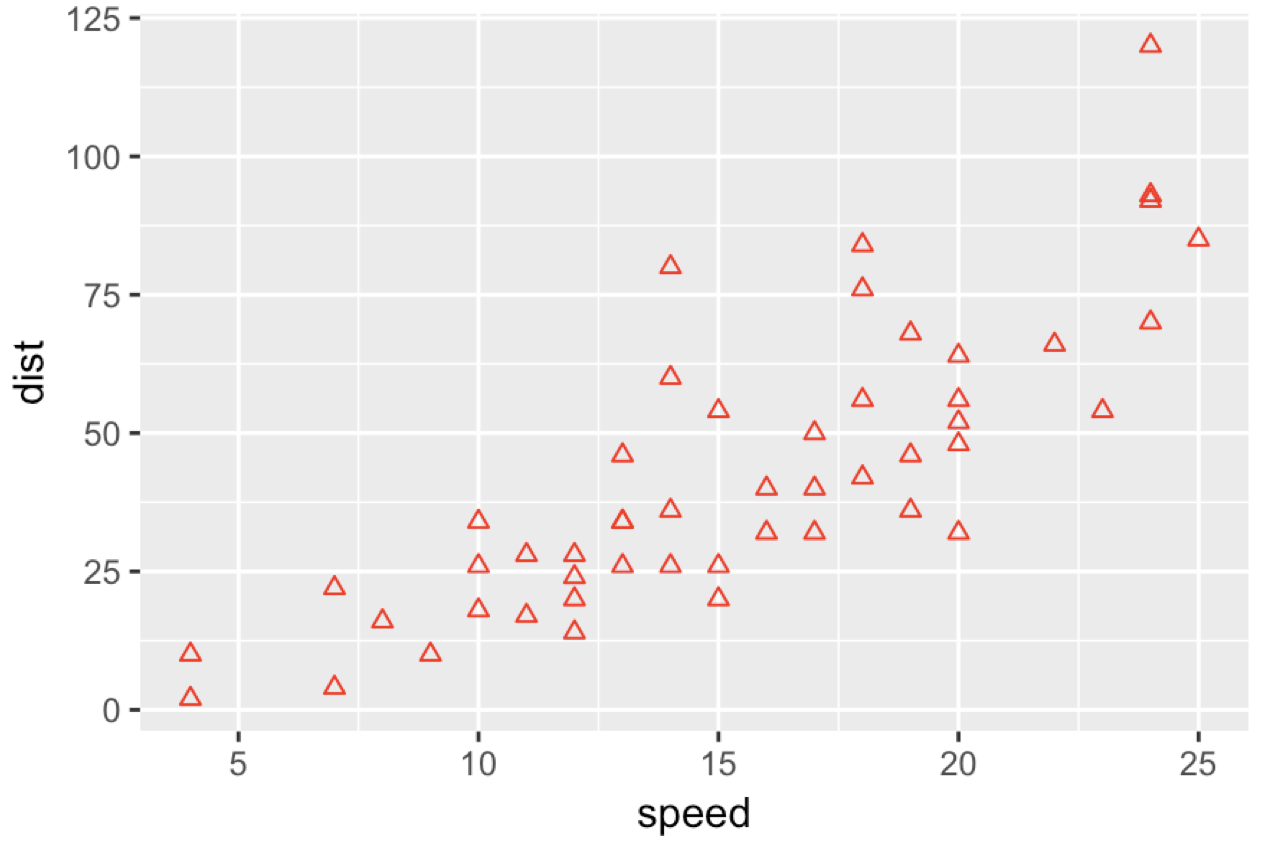


圖 13-21 ggplot2 調整資料點

應用不同的**類別**區分資料點的形狀與顏色在 ggplot2 是很容易的：

> library(ggplot2)  
  
> ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) +  
+ geom\_point(aes(shape = Species, colour = Species))

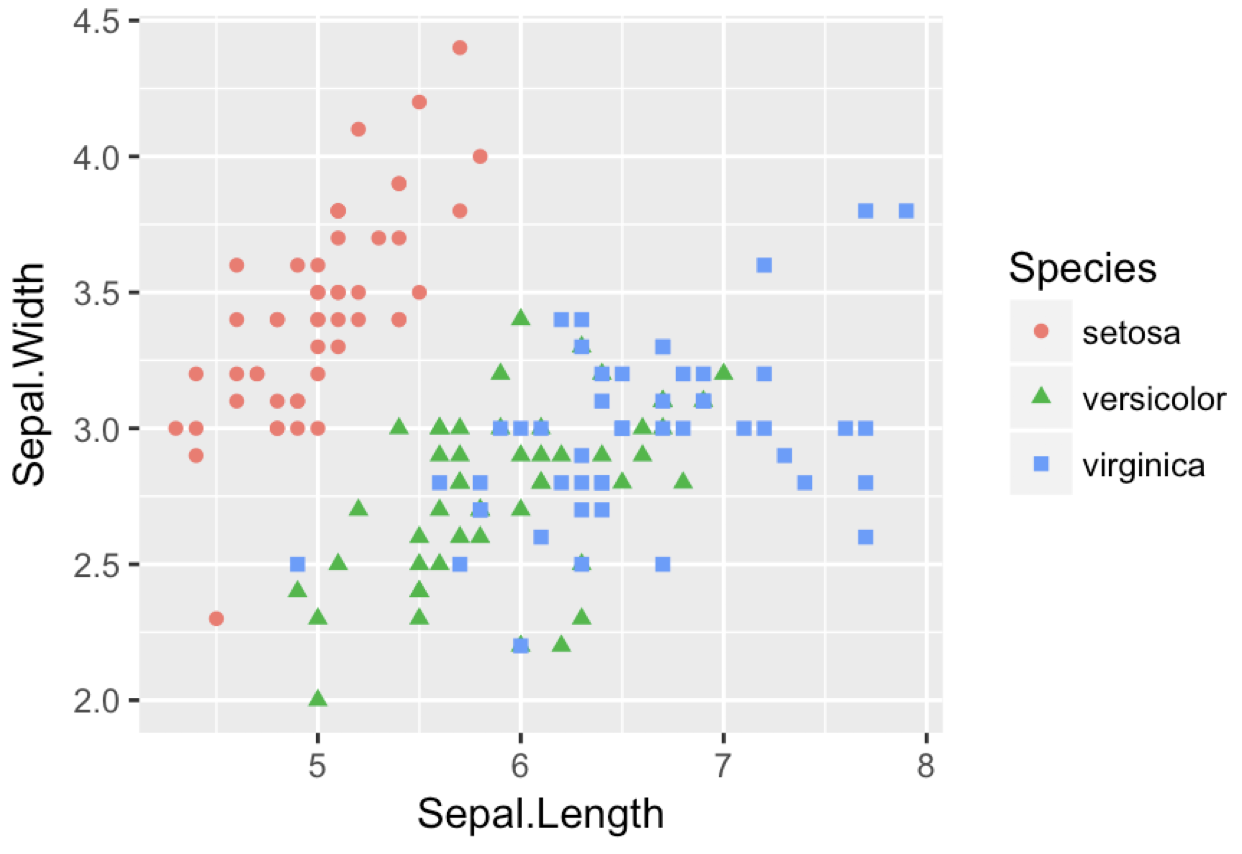


圖 13-22 ggplot2 調整資料點

##### 繪畫多個圖形

ggplot2 套件本身並沒有針對繪畫多個圖形這部分設計函數，但我們可以利用另外一個套件 gridExtra 所提供的 grid.arrange() 函數來達到 par(mfrow = c(m, n)) 的效果：

library(ggplot2)  
library(gridExtra)  
  
g1 <- ggplot(iris, aes(x = Species, y = Sepal.Length)) +  
 geom\_boxplot()  
g2 <- ggplot(iris, aes(x = Species, y = Sepal.Width)) +  
 geom\_boxplot()  
g3 <- ggplot(iris, aes(x = Species, y = Petal.Length)) +  
 geom\_boxplot()  
g4 <- ggplot(iris, aes(x = Species, y = Petal.Width)) +  
 geom\_boxplot()  
grid.arrange(g1, g2, g3, g4, nrow = 2, ncol = 2)

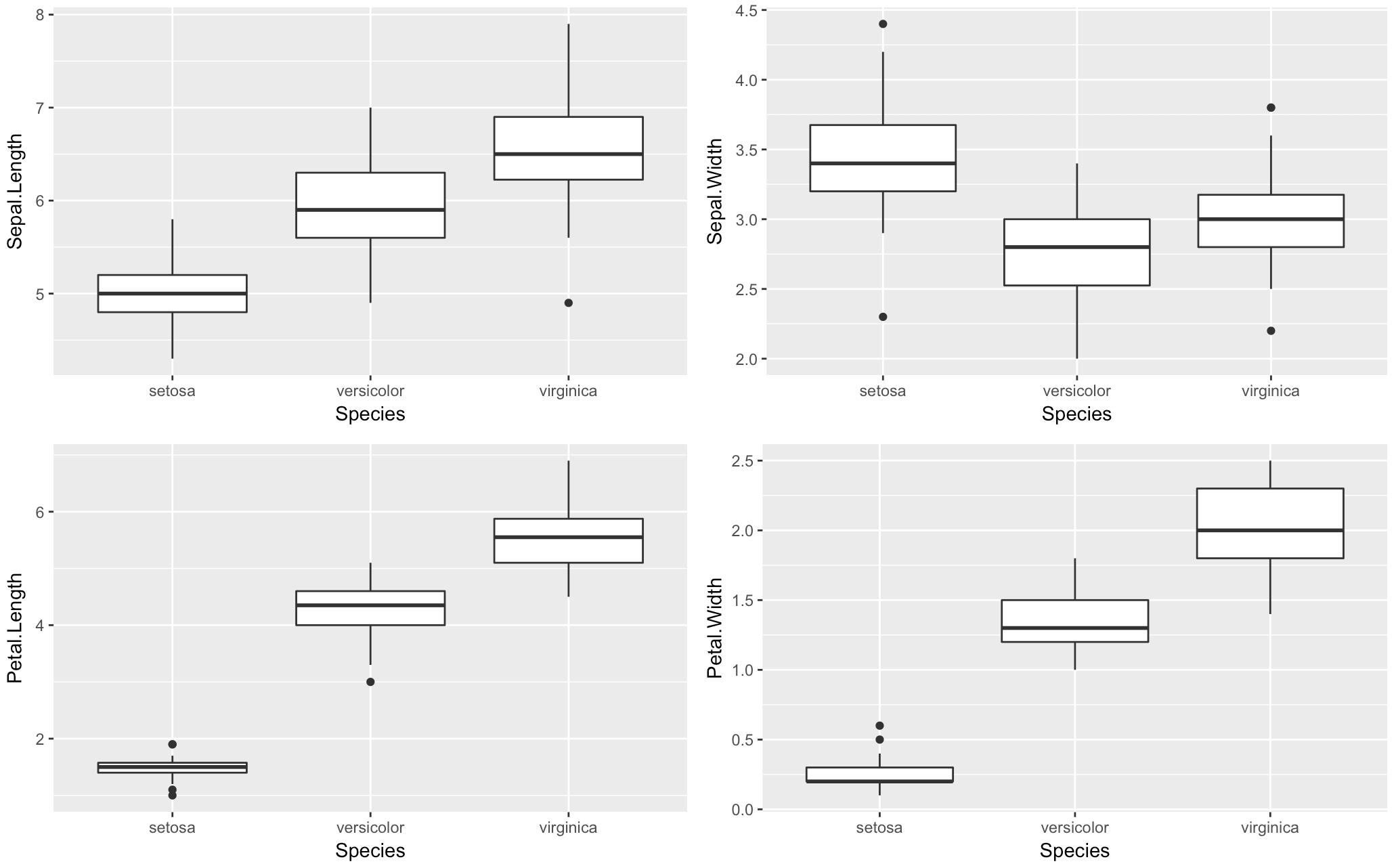


圖 13-23 ggplot2 多個圖形

### 滄海一粟

如同 **Base Plotting System** 一樣，雖然我們已經介紹了一定程度的技巧，但是這些功能相較 ggplot2 豐富的參數與功能實在是太微不足道。請您同樣也要耐著性子，在使用時查詢大量的文件、Google 與 StackOverflow。ggplot2 非常好的地方是它的[文件](http://docs.ggplot2.org/current/)除了文字說明以外，還有豐富的示範程式與圖形，對使用者非常友善。

### 小結

好啦！第十三天的內容就是這麼多，今天說明的是受到廣大 R 語言使用者喜愛的 **ggplot2** 基礎應用，希望您還喜歡這個繪圖套件！

### 練習題

###### 將一個畫布切割成為 2X2 個區塊，並使用 **ggplot2** 繪製任意四種圖形。

### 延伸資訊

* [R Graphics Cookbook](http://shop.oreilly.com/product/0636920023135.do)
* [Graphs with ggplot2](http://www.cookbook-r.com/Graphs/)
* [ggplot2.org](http://docs.ggplot2.org/current/)