Bioinformatics - R Dataframe and Plots

Wei-Hao, Lee

3/28/2019

## Outline

1. Review
   * 註解 (Comment)
   * 縮排 (Indent)
   * 作業講解 – 九九乘法表
2. 函式 (function)
3. Package installation
4. Dataframe
   * Check Data Format
   * Load or Read Data
   * Basic Operation
   * Output Data
   * Advanced technique (Depends teaching progress)
5. Plots
   * Scatter plots
   * Box plots
   * Line chart
   * Save plot

### 0. Review

#### 註解 (Comment)

註解就是替程式增添文字上的說明，方便日後重新閱讀程式能夠快速了解當初的原意。在R語言中只提供單行註解，利用# 你想打的東西放在井字號後都是註解喔～，舉例來說：

num <- "Wei-Hao" # 這是名字  
# 也可以獨立出一行作註解

同時也可以利用註解的功能，將部分的程式改為註解，就不會執行了！

age <- 18  
# age <- age + 2  
print(age)

## [1] 18

#### 縮排 (Indent)

縮排其實就是利用tab鍵使文字內縮，就像寫作文分段時會留空告訴讀者這是新的一段。在程式語言中，縮排可以幫助理解程式的邏輯以及架構，舉例來說：

age <- 30  
if (age > 18) {  
 print("Older than 18!") # 這裡就是按了一個tab鍵做程式的縮排  
}

## [1] "Older than 18!"

如果程式不使用縮排時會導致程式難以閱讀，看以下九九乘法表範例：

for (i in 1:9)  
{  
for (j in 1:9)  
{  
cat(i, "\*", j, "=", i\*j, "\t")  
}  
cat("\n")  
}

是不是變得難以理解了呢，如果我們的使用縮排就可以明顯看出層級關係了喔！

for (i in 1:9)  
{  
 for (j in 1:9)  
 {  
 cat(i, "\*", j, "=", i\*j, "\t")  
 }  
 cat("\n")  
}

#### 作業講解 – 九九乘法表 (僅供參考)

**程式並沒有一定的答案，只要寫得出來就是對的答案，這堂課只需要練習出寫出對的答案，無需追求執行速度**

# 現練習用for迴圈印出1~9並且分行印出  
for (i in 1:9)  
{  
 cat(i)  
 cat("\n") # \n 換行符號代表 newline  
}

## 1  
## 2  
## 3  
## 4  
## 5  
## 6  
## 7  
## 8  
## 9

如今我們可以另用i印出19，我們進一步的在每一行都印出19

for (i in 1:9)  
{  
 # 其實很簡單，我們在這邊在加入一個for迴圈，負責印出1~9  
 for (j in 1:9) # 切記這裡變數不可以在使用i  
 {  
 cat("i:j=", i, ":", j, "\t", sep="")  
 }  
 cat("\n")  
}

## i:j=1:1 i:j=1:2 i:j=1:3 i:j=1:4 i:j=1:5 i:j=1:6 i:j=1:7 i:j=1:8 i:j=1:9   
## i:j=2:1 i:j=2:2 i:j=2:3 i:j=2:4 i:j=2:5 i:j=2:6 i:j=2:7 i:j=2:8 i:j=2:9   
## i:j=3:1 i:j=3:2 i:j=3:3 i:j=3:4 i:j=3:5 i:j=3:6 i:j=3:7 i:j=3:8 i:j=3:9   
## i:j=4:1 i:j=4:2 i:j=4:3 i:j=4:4 i:j=4:5 i:j=4:6 i:j=4:7 i:j=4:8 i:j=4:9   
## i:j=5:1 i:j=5:2 i:j=5:3 i:j=5:4 i:j=5:5 i:j=5:6 i:j=5:7 i:j=5:8 i:j=5:9   
## i:j=6:1 i:j=6:2 i:j=6:3 i:j=6:4 i:j=6:5 i:j=6:6 i:j=6:7 i:j=6:8 i:j=6:9   
## i:j=7:1 i:j=7:2 i:j=7:3 i:j=7:4 i:j=7:5 i:j=7:6 i:j=7:7 i:j=7:8 i:j=7:9   
## i:j=8:1 i:j=8:2 i:j=8:3 i:j=8:4 i:j=8:5 i:j=8:6 i:j=8:7 i:j=8:8 i:j=8:9   
## i:j=9:1 i:j=9:2 i:j=9:3 i:j=9:4 i:j=9:5 i:j=9:6 i:j=9:7 i:j=9:8 i:j=9:9

有沒有發現，其實這已經就是九九乘法了嗎？我們再稍微修改一下印出的格式即可囉～

for (i in 1:9)  
{  
 for (j in 1:9)  
 {  
 cat(i, "\*", j, "=", i\*j, "\t") # 其實要印出兩數字相乘，直接在cat函式中數入i\*j就可以了喔！  
 }  
 cat("\n")  
}

## 1 \* 1 = 1 1 \* 2 = 2 1 \* 3 = 3 1 \* 4 = 4 1 \* 5 = 5 1 \* 6 = 6 1 \* 7 = 7 1 \* 8 = 8 1 \* 9 = 9   
## 2 \* 1 = 2 2 \* 2 = 4 2 \* 3 = 6 2 \* 4 = 8 2 \* 5 = 10 2 \* 6 = 12 2 \* 7 = 14 2 \* 8 = 16 2 \* 9 = 18   
## 3 \* 1 = 3 3 \* 2 = 6 3 \* 3 = 9 3 \* 4 = 12 3 \* 5 = 15 3 \* 6 = 18 3 \* 7 = 21 3 \* 8 = 24 3 \* 9 = 27   
## 4 \* 1 = 4 4 \* 2 = 8 4 \* 3 = 12 4 \* 4 = 16 4 \* 5 = 20 4 \* 6 = 24 4 \* 7 = 28 4 \* 8 = 32 4 \* 9 = 36   
## 5 \* 1 = 5 5 \* 2 = 10 5 \* 3 = 15 5 \* 4 = 20 5 \* 5 = 25 5 \* 6 = 30 5 \* 7 = 35 5 \* 8 = 40 5 \* 9 = 45   
## 6 \* 1 = 6 6 \* 2 = 12 6 \* 3 = 18 6 \* 4 = 24 6 \* 5 = 30 6 \* 6 = 36 6 \* 7 = 42 6 \* 8 = 48 6 \* 9 = 54   
## 7 \* 1 = 7 7 \* 2 = 14 7 \* 3 = 21 7 \* 4 = 28 7 \* 5 = 35 7 \* 6 = 42 7 \* 7 = 49 7 \* 8 = 56 7 \* 9 = 63   
## 8 \* 1 = 8 8 \* 2 = 16 8 \* 3 = 24 8 \* 4 = 32 8 \* 5 = 40 8 \* 6 = 48 8 \* 7 = 56 8 \* 8 = 64 8 \* 9 = 72   
## 9 \* 1 = 9 9 \* 2 = 18 9 \* 3 = 27 9 \* 4 = 36 9 \* 5 = 45 9 \* 6 = 54 9 \* 7 = 63 9 \* 8 = 72 9 \* 9 = 81

*稍微再思考一下，如何印出直行的九九乘法表呢！*

## 1 \* 1 = 1 2 \* 1 = 2 3 \* 1 = 3 4 \* 1 = 4 5 \* 1 = 5 6 \* 1 = 6 7 \* 1 = 7 8 \* 1 = 8 9 \* 1 = 9   
## 1 \* 2 = 2 2 \* 2 = 4 3 \* 2 = 6 4 \* 2 = 8 5 \* 2 = 10 6 \* 2 = 12 7 \* 2 = 14 8 \* 2 = 16 9 \* 2 = 18   
## 1 \* 3 = 3 2 \* 3 = 6 3 \* 3 = 9 4 \* 3 = 12 5 \* 3 = 15 6 \* 3 = 18 7 \* 3 = 21 8 \* 3 = 24 9 \* 3 = 27   
## 1 \* 4 = 4 2 \* 4 = 8 3 \* 4 = 12 4 \* 4 = 16 5 \* 4 = 20 6 \* 4 = 24 7 \* 4 = 28 8 \* 4 = 32 9 \* 4 = 36   
## 1 \* 5 = 5 2 \* 5 = 10 3 \* 5 = 15 4 \* 5 = 20 5 \* 5 = 25 6 \* 5 = 30 7 \* 5 = 35 8 \* 5 = 40 9 \* 5 = 45   
## 1 \* 6 = 6 2 \* 6 = 12 3 \* 6 = 18 4 \* 6 = 24 5 \* 6 = 30 6 \* 6 = 36 7 \* 6 = 42 8 \* 6 = 48 9 \* 6 = 54   
## 1 \* 7 = 7 2 \* 7 = 14 3 \* 7 = 21 4 \* 7 = 28 5 \* 7 = 35 6 \* 7 = 42 7 \* 7 = 49 8 \* 7 = 56 9 \* 7 = 63   
## 1 \* 8 = 8 2 \* 8 = 16 3 \* 8 = 24 4 \* 8 = 32 5 \* 8 = 40 6 \* 8 = 48 7 \* 8 = 56 8 \* 8 = 64 9 \* 8 = 72   
## 1 \* 9 = 9 2 \* 9 = 18 3 \* 9 = 27 4 \* 9 = 36 5 \* 9 = 45 6 \* 9 = 54 7 \* 9 = 63 8 \* 9 = 72 9 \* 9 = 81

### 1. 函式 (function)

function其實就是使用者自己定義的功能，且可以重複的使用。當一個程式大量撰寫重複的程式時，我們便可以撰寫成函式，方便使用。而我們其實已經看過很多函式像是as.integer(), mean()等。接下來就來介紹函式撰寫的方式：

# 這就是最簡的 function，沒有任何功能的函式  
# 函式名稱 <- function() {}  
simplest\_function <- function() {  
   
}  
  
simplest\_function() # 呼叫函式時，記得一定要寫括弧

## NULL

simplest\_function

## function() {  
##   
## }

我們開始在{}中，加入我們要使用的功能

# 記得函式也要遵守變數的命名原則  
hello\_world <- function() {  
 print("Hello World!")  
}  
  
hello\_world()

## [1] "Hello World!"

假設今天我們要撰寫我們自己的function專門計算平均值，代表我們的function要有能力接受我們傳入的數值在去做平均值的計算。方法其實很簡單：

# 在function 的“()”中設定我們要接受的資料  
class\_height <- 175  
  
# 這邊的numbers其實也是個變數，專門接收我們傳入的資料，我們通常稱這些變數為parameter參數  
bio\_means <- function(numbers) {   
 print(numbers)  
}  
  
bio\_means(class\_height)

## [1] 175

# 這邊要注意！我們可以看到numbers無法被印出來  
# 原因是，當function結束時，numbers這個變數也會跟著消失  
print(numbers)

## Error in print(numbers): object 'numbers' not found

我們也可以同時傳入多個資料：

name <- "Wei-Hao"  
id <- 123456789  
  
identification <- function(name, id) {  
 cat("name :", name)  
 cat("\n")  
 cat("id :", id)  
}  
identification(name, id)

## name : Wei-Hao  
## id : 123456789

# 如果你明確知道哪些參數資料需要什麼樣類型的資料  
# 也可以直接在function中，明確指出哪些參數等於哪些資料  
identification(name="Lee", id="0217047")

## name : Lee  
## id : 0217047

weird\_identification <- function(name, id) {  
 name <- "BA BA BA" # change both id and name  
 id <- "LA LA LA"  
 cat("name :", name)  
 cat("\n")  
 cat("id :", id)  
}  
weird\_identification(name, id)

## name : BA BA BA  
## id : LA LA LA

# 我們在identification更改的東西，並未直接的  
cat("name :", name, "\n", "id :", id)

## name : Wei-Hao   
## id : 123456789

# 跟identification 只差了function  
tricky\_identification <- function() {  
 cat("name :", name)  
 cat("\n")  
 cat("id :", id)  
}  
tricky\_identification()

## name : Wei-Hao  
## id : 123456789

從上述兩個function中講了一些蠻重要的觀念：

1. function中的參數，雖然名稱與傳入的變數相同，但是是不同的東西(看weird\_identification()的例子)，function中的參數是另外獨立屬於這個function的變數，而傳入的資料則會複製一份給參數。
2. 在看tricky\_identification()，我們可以觀察到，若function沒有參數時，但裡面有用到已經存在的變數，他便會自動的調取這些變數做使用！

我們剛剛的例子，都只是印出東西，如果我們想要有個函式計算平均值，並且把計算完後的平均值傳給某個變數該怎麼做呢，我們只需要使用return()就可以達成囉！

class\_heights <- c(123, 145, 135 , 175, 189)  
  
mean\_height <- function(class\_heights) {  
 total\_height <- 0  
 num\_of\_classmate <- length(class\_heights)  
   
 for (height in class\_heights)  
 {  
 total\_height <- total\_height + height  
 }  
 return(total\_height/num\_of\_classmate)  
}  
  
class\_mean\_height <- mean\_height(class\_heights)  
  
print(class\_mean\_height)

## [1] 153.4

上述都是function中蠻重要的觀念，還有一小部分沒有講到(Scope)，如果有興趣的同學可以點取連結，寫出更有彈性的function。 [Creating Functions](https://swcarpentry.github.io/r-novice-inflammation/02-func-R/index.html)

#### Practice.1 - Selection sort

Selection sort 是一種排序的方法，它的原理很簡單，請參考 [Selection Sort](https://en.wikipedia.org/wiki/Selection_sort) 你們要寫出一個function叫selection\_sort(series)，series是一連串的數字用vector存放，這個function要能夠回傳由小排到大的vector。請大家練習自己用for迴圈寫出來，不要使用sort()!! 題目有點困難，請同學思考一下，也可以參考網路上的解答，不要鑽研 big O的問題。

series <- c(0, 3, 1, 13, 84, 25, 91, 1, 2)

selection\_sort(series)

## [1] 0 1 1 2 3 13 25 84 91

### 2. Package installation

Package其實就是別人將他寫好的function整理成一個package，提供給別人使用。通常每個package都會有他主要處理的目的，像是dplyr就是專門為了處理資料所使用的package，ggplot2專門為了繪圖所使用的package，接下來我們會教大家如何安裝package。 我們只要在console中打入install.packages("dplyr")，就可以開始下載dplyr囉！

下載完後，並非可以直接使用，你只是將package下載到電腦中。要使用時要：

library(dplyr) # 可能會覺得有點怪，一下叫package一下叫library，這邊就請同學熟悉一下囉！

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

那麼這邊就麻煩同學下載今天要使用的package: dplyr, ggplot2, ggtheme, reshape2, magrittr

### 3. Dataframe

#### Check Data Format

dataframe是一種資料型態，最為大家熟知的就像是excel檔。然而我們在excel中可以任意使用裡面的格子，而且不需要有整齊的格式，但是在R語言甚至是python裡的pandas都是無法處理這種格式不一致的檔案。因此我們在讀取資料前，要先確人我們的資料格式正確且整齊，避免電腦無法匯入想要的資料。待會再進行資料處理時，你們也可以同時知道為什麼R語言並不支援讀取格式混亂的檔案。

首先這邊會推薦大家使用幾個好用的text editor。原先text editor其實就是提供使用編輯文字用的程式，但由於近幾年的開發，目前大多數的text editor都有編譯執行程式的功能。

* Visual Studio Code (Personal Recommendation)
* Atom
* Notepad ++
* … more

接下來要介紹幾種常見的格式.tsv, csv

# tsv file, 以tab鍵作為資料的分隔  
1 2 3 4 5 6  
  
# csv file, 以comma作為資料的分隔  
1,2,3,4,5,6

#### Load Data

讀取檔案有兩種方式，一種是透過R studio做匯入，另外一種透過絕對路徑(absolute path)再搭配read.csv() or read.table()，那麼接下來主要是示範如何透過絕對路徑讀取檔案。首先絕對路徑就是一段字串告訴電腦檔案的位置 ex: ~/Documents/TeacherAssistant/Bioninformatics/2019\_03\_28/Iris.csv (for Mac OSX, Linux，在mac中絕對路徑不好取得，這邊教大家使用一下偷吃步！如果會command line的同學請盡量使用command line), C:\Documents\Newsletters\Summer2018.pdf (for Windows，Windows的絕對路徑較好取得)。接下來就是讀取檔案

# 讀取工作路徑  
getwd()

## [1] "/Users/lee-weihao/Documents/TeacherAssistant/Bioninformatics/2019\_03\_28"

# 設定工作路徑  
# setwd()

# how to read csv file  
iris <- read.csv(file = "./Iris.csv", # file path  
 stringsAsFactors = FALSE, # 只要有文字的那行，會直接轉為factor，因使我們要取消此功能  
 row.names = 1, # 告訴電腦第一列資料是 row name  
 header = TRUE, # 第一行是 column 的名稱  
 check.names = FALSE) # check.name是幫使用者檢查 column and row name，有時候名稱會跑掉  
head(iris, n=5) # 顯示 iris dataframe 前五行

## 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

# how to read tsv file  
iris <- read.table(file = "./Iris.tsv", # file path  
 stringsAsFactors = FALSE, # 只要有文字的那行，會直接轉為factor，因使我們要取消此功能  
 row.names = 1, # 告訴電腦第一列資料是 row name  
 header = TRUE,  
 sep = "\t", # 設定文件是使用什麼符號做為分隔  
 check.names = FALSE) # check.name是幫使用者檢查 column and row name，有時候名稱會跑掉  
head(iris, n=5) # 顯示 iris dataframe 前五行

## 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

# how to rad excel format  
library(readxl)  
iris\_excel <- read\_excel(path = "./Iris.xlsm", # file path  
 sheet = 1, # number of sheet, or you can use the name of sheet you want  
 col\_names = TRUE,  
 col\_types = NULL,   
 na = "", # missing value  
 skip = 0) #Number of rows to skip before reading any data.

## New names:  
## \* `` -> `..1`

head(iris\_excel, n=5)

## # A tibble: 5 x 6  
## ..1 Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## <dbl> <dbl> <dbl> <dbl> <dbl> <chr>   
## 1 1 5.1 3.5 1.4 0.2 setosa   
## 2 2 4.9 3 1.4 0.2 setosa   
## 3 3 4.7 3.2 1.3 0.2 setosa   
## 4 4 4.6 3.1 1.5 0.2 setosa   
## 5 5 5 3.6 1.4 0.2 setosa

*補充說明，如果是讀取excel檔的話，read\_excel()會將資料整理成tibble的格式，看起來會與前面兩者有些不同，但是基本操作上是一模一樣的，但是有些功能會有異，如果想要讓他成為一般的dataframe，請使用as.data.frame()。*

class(iris\_excel)

## [1] "tbl\_df" "tbl" "data.frame"

iris\_excel <- as.data.frame(iris\_excel)  
class(iris\_excel)

## [1] "data.frame"

head(iris\_excel)

## ..1 Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 1 1 5.1 3.5 1.4 0.2 setosa  
## 2 2 4.9 3.0 1.4 0.2 setosa  
## 3 3 4.7 3.2 1.3 0.2 setosa  
## 4 4 4.6 3.1 1.5 0.2 setosa  
## 5 5 5.0 3.6 1.4 0.2 setosa  
## 6 6 5.4 3.9 1.7 0.4 setosa

如今你已經會讀取檔案了，通常在讀取完資料花一點時間，看一下你的資料是否有誤，不要做完分析後才發現原使數據有誤喔！

#### Basic operation

首先我們要練習如何讀取dataframe裡面的資料，像是讀取特定行數、利用行或列的名稱讀取該行該列。在R語言中有內建很好用的格式，讓使用者使用dataframe\_variable[row, column]。接下來為大家示範一下：

# row - 行; column - 列 (這堂課之後都以英文代表行列)  
# 首先先觀察資料  
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

# 假設我們要選取 Species這個 column  
iris[,"Species"] # 由於我們沒有要選取特定row，所以我們可以不用填寫row (這和python的pandas稍有不同)

## [1] "setosa" "setosa" "setosa" "setosa" "setosa"   
## [6] "setosa" "setosa" "setosa" "setosa" "setosa"   
## [11] "setosa" "setosa" "setosa" "setosa" "setosa"   
## [16] "setosa" "setosa" "setosa" "setosa" "setosa"   
## [21] "setosa" "setosa" "setosa" "setosa" "setosa"   
## [26] "setosa" "setosa" "setosa" "setosa" "setosa"   
## [31] "setosa" "setosa" "setosa" "setosa" "setosa"   
## [36] "setosa" "setosa" "setosa" "setosa" "setosa"   
## [41] "setosa" "setosa" "setosa" "setosa" "setosa"   
## [46] "setosa" "setosa" "setosa" "setosa" "setosa"   
## [51] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"  
## [56] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"  
## [61] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"  
## [66] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"  
## [71] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"  
## [76] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"  
## [81] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"  
## [86] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"  
## [91] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"  
## [96] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"  
## [101] "virginica" "virginica" "virginica" "virginica" "virginica"   
## [106] "virginica" "virginica" "virginica" "virginica" "virginica"   
## [111] "virginica" "virginica" "virginica" "virginica" "virginica"   
## [116] "virginica" "virginica" "virginica" "virginica" "virginica"   
## [121] "virginica" "virginica" "virginica" "virginica" "virginica"   
## [126] "virginica" "virginica" "virginica" "virginica" "virginica"   
## [131] "virginica" "virginica" "virginica" "virginica" "virginica"   
## [136] "virginica" "virginica" "virginica" "virginica" "virginica"   
## [141] "virginica" "virginica" "virginica" "virginica" "virginica"   
## [146] "virginica" "virginica" "virginica" "virginica" "virginica"

# 假設我們今天要選取 1~5 的row  
iris[1:5,]

## 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

# 假設我們今天要取出row 1~5 的Petal.Width  
iris[1:5, "Petal.Width"]

## [1] 0.2 0.2 0.2 0.2 0.2

# 也可以將Petal.Width改為他在第幾個column  
iris[1:5, 4]

## [1] 0.2 0.2 0.2 0.2 0.2

事實上R語言中有提供更方便的方式，讓使用者讀取某個column (column only!!) 就是$ dollar sign，接下來會見到他很多次，請大家習慣他的用法。

# dataframe\_name$column\_name  
iris$Species # same as iris[,"Species"]

## [1] "setosa" "setosa" "setosa" "setosa" "setosa"   
## [6] "setosa" "setosa" "setosa" "setosa" "setosa"   
## [11] "setosa" "setosa" "setosa" "setosa" "setosa"   
## [16] "setosa" "setosa" "setosa" "setosa" "setosa"   
## [21] "setosa" "setosa" "setosa" "setosa" "setosa"   
## [26] "setosa" "setosa" "setosa" "setosa" "setosa"   
## [31] "setosa" "setosa" "setosa" "setosa" "setosa"   
## [36] "setosa" "setosa" "setosa" "setosa" "setosa"   
## [41] "setosa" "setosa" "setosa" "setosa" "setosa"   
## [46] "setosa" "setosa" "setosa" "setosa" "setosa"   
## [51] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"  
## [56] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"  
## [61] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"  
## [66] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"  
## [71] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"  
## [76] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"  
## [81] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"  
## [86] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"  
## [91] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"  
## [96] "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"  
## [101] "virginica" "virginica" "virginica" "virginica" "virginica"   
## [106] "virginica" "virginica" "virginica" "virginica" "virginica"   
## [111] "virginica" "virginica" "virginica" "virginica" "virginica"   
## [116] "virginica" "virginica" "virginica" "virginica" "virginica"   
## [121] "virginica" "virginica" "virginica" "virginica" "virginica"   
## [126] "virginica" "virginica" "virginica" "virginica" "virginica"   
## [131] "virginica" "virginica" "virginica" "virginica" "virginica"   
## [136] "virginica" "virginica" "virginica" "virginica" "virginica"   
## [141] "virginica" "virginica" "virginica" "virginica" "virginica"   
## [146] "virginica" "virginica" "virginica" "virginica" "virginica"

# 假設我們今天想要確認 Species 中是否有 setosa  
iris$Species == "setosa" # 會回傳 true and false

## [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [12] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [23] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [34] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [45] TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE  
## [56] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [67] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [78] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [89] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [100] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [111] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [122] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [133] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [144] FALSE FALSE FALSE FALSE FALSE FALSE FALSE

# 那我們如何從原始的dataframe取出只含有 virginica species的 row呢  
iris[iris$Species == "virginica",]

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 101 6.3 3.3 6.0 2.5 virginica  
## 102 5.8 2.7 5.1 1.9 virginica  
## 103 7.1 3.0 5.9 2.1 virginica  
## 104 6.3 2.9 5.6 1.8 virginica  
## 105 6.5 3.0 5.8 2.2 virginica  
## 106 7.6 3.0 6.6 2.1 virginica  
## 107 4.9 2.5 4.5 1.7 virginica  
## 108 7.3 2.9 6.3 1.8 virginica  
## 109 6.7 2.5 5.8 1.8 virginica  
## 110 7.2 3.6 6.1 2.5 virginica  
## 111 6.5 3.2 5.1 2.0 virginica  
## 112 6.4 2.7 5.3 1.9 virginica  
## 113 6.8 3.0 5.5 2.1 virginica  
## 114 5.7 2.5 5.0 2.0 virginica  
## 115 5.8 2.8 5.1 2.4 virginica  
## 116 6.4 3.2 5.3 2.3 virginica  
## 117 6.5 3.0 5.5 1.8 virginica  
## 118 7.7 3.8 6.7 2.2 virginica  
## 119 7.7 2.6 6.9 2.3 virginica  
## 120 6.0 2.2 5.0 1.5 virginica  
## 121 6.9 3.2 5.7 2.3 virginica  
## 122 5.6 2.8 4.9 2.0 virginica  
## 123 7.7 2.8 6.7 2.0 virginica  
## 124 6.3 2.7 4.9 1.8 virginica  
## 125 6.7 3.3 5.7 2.1 virginica  
## 126 7.2 3.2 6.0 1.8 virginica  
## 127 6.2 2.8 4.8 1.8 virginica  
## 128 6.1 3.0 4.9 1.8 virginica  
## 129 6.4 2.8 5.6 2.1 virginica  
## 130 7.2 3.0 5.8 1.6 virginica  
## 131 7.4 2.8 6.1 1.9 virginica  
## 132 7.9 3.8 6.4 2.0 virginica  
## 133 6.4 2.8 5.6 2.2 virginica  
## 134 6.3 2.8 5.1 1.5 virginica  
## 135 6.1 2.6 5.6 1.4 virginica  
## 136 7.7 3.0 6.1 2.3 virginica  
## 137 6.3 3.4 5.6 2.4 virginica  
## 138 6.4 3.1 5.5 1.8 virginica  
## 139 6.0 3.0 4.8 1.8 virginica  
## 140 6.9 3.1 5.4 2.1 virginica  
## 141 6.7 3.1 5.6 2.4 virginica  
## 142 6.9 3.1 5.1 2.3 virginica  
## 143 5.8 2.7 5.1 1.9 virginica  
## 144 6.8 3.2 5.9 2.3 virginica  
## 145 6.7 3.3 5.7 2.5 virginica  
## 146 6.7 3.0 5.2 2.3 virginica  
## 147 6.3 2.5 5.0 1.9 virginica  
## 148 6.5 3.0 5.2 2.0 virginica  
## 149 6.2 3.4 5.4 2.3 virginica  
## 150 5.9 3.0 5.1 1.8 virginica

# 我們也可以檢查哪些sample的petal.width大於等於２  
iris[iris$Petal.Width >= 2,]

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 101 6.3 3.3 6.0 2.5 virginica  
## 103 7.1 3.0 5.9 2.1 virginica  
## 105 6.5 3.0 5.8 2.2 virginica  
## 106 7.6 3.0 6.6 2.1 virginica  
## 110 7.2 3.6 6.1 2.5 virginica  
## 111 6.5 3.2 5.1 2.0 virginica  
## 113 6.8 3.0 5.5 2.1 virginica  
## 114 5.7 2.5 5.0 2.0 virginica  
## 115 5.8 2.8 5.1 2.4 virginica  
## 116 6.4 3.2 5.3 2.3 virginica  
## 118 7.7 3.8 6.7 2.2 virginica  
## 119 7.7 2.6 6.9 2.3 virginica  
## 121 6.9 3.2 5.7 2.3 virginica  
## 122 5.6 2.8 4.9 2.0 virginica  
## 123 7.7 2.8 6.7 2.0 virginica  
## 125 6.7 3.3 5.7 2.1 virginica  
## 129 6.4 2.8 5.6 2.1 virginica  
## 132 7.9 3.8 6.4 2.0 virginica  
## 133 6.4 2.8 5.6 2.2 virginica  
## 136 7.7 3.0 6.1 2.3 virginica  
## 137 6.3 3.4 5.6 2.4 virginica  
## 140 6.9 3.1 5.4 2.1 virginica  
## 141 6.7 3.1 5.6 2.4 virginica  
## 142 6.9 3.1 5.1 2.3 virginica  
## 144 6.8 3.2 5.9 2.3 virginica  
## 145 6.7 3.3 5.7 2.5 virginica  
## 146 6.7 3.0 5.2 2.3 virginica  
## 148 6.5 3.0 5.2 2.0 virginica  
## 149 6.2 3.4 5.4 2.3 virginica

那要如何選取多個欄位，以及更改欄位順序呢？

# select multiple column  
sub\_iris <- iris[,c("Petal.Width", "Species")] # 這邊就無法使用$做多行選取喔！  
head(sub\_iris)

## Petal.Width Species  
## 1 0.2 setosa  
## 2 0.2 setosa  
## 3 0.2 setosa  
## 4 0.2 setosa  
## 5 0.2 setosa  
## 6 0.4 setosa

# changing column order  
sub\_iris <- iris[,c("Species", "Petal.Width")] # 大家是否有發現，我們挑選出來的column換按照我們輸入的順序改變喔  
head(sub\_iris) # 因此我們如果要調整個column順序的話，使用者就得乖乖的把你想要的順序寫出來喔！

## Species Petal.Width  
## 1 setosa 0.2  
## 2 setosa 0.2  
## 3 setosa 0.2  
## 4 setosa 0.2  
## 5 setosa 0.2  
## 6 setosa 0.4

在這邊為止，我們已經會了一些基本的操作，接下來我們會介紹一些常用的function：

# number of row  
nrow(iris)

## [1] 150

# number of column  
ncol(iris)

## [1] 5

# dimension  
dim(iris)

## [1] 150 5

# column names  
colnames(iris) # same as names(iris), but I prefer using colnames.

## [1] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width"   
## [5] "Species"

# row names  
rownames(iris) # 由於我們這次的資料並沒有rowname，所以看rownames意義不大，但是在下次上課我們就會用到囉

## [1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11"   
## [12] "12" "13" "14" "15" "16" "17" "18" "19" "20" "21" "22"   
## [23] "23" "24" "25" "26" "27" "28" "29" "30" "31" "32" "33"   
## [34] "34" "35" "36" "37" "38" "39" "40" "41" "42" "43" "44"   
## [45] "45" "46" "47" "48" "49" "50" "51" "52" "53" "54" "55"   
## [56] "56" "57" "58" "59" "60" "61" "62" "63" "64" "65" "66"   
## [67] "67" "68" "69" "70" "71" "72" "73" "74" "75" "76" "77"   
## [78] "78" "79" "80" "81" "82" "83" "84" "85" "86" "87" "88"   
## [89] "89" "90" "91" "92" "93" "94" "95" "96" "97" "98" "99"   
## [100] "100" "101" "102" "103" "104" "105" "106" "107" "108" "109" "110"  
## [111] "111" "112" "113" "114" "115" "116" "117" "118" "119" "120" "121"  
## [122] "122" "123" "124" "125" "126" "127" "128" "129" "130" "131" "132"  
## [133] "133" "134" "135" "136" "137" "138" "139" "140" "141" "142" "143"  
## [144] "144" "145" "146" "147" "148" "149" "150"

# looking from top  
head(iris, n=5)

## 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

# looking from bottom  
tail(iris, n=5)

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 146 6.7 3.0 5.2 2.3 virginica  
## 147 6.3 2.5 5.0 1.9 virginica  
## 148 6.5 3.0 5.2 2.0 virginica  
## 149 6.2 3.4 5.4 2.3 virginica  
## 150 5.9 3.0 5.1 1.8 virginica

# overview dataset  
summary(iris)

## Sepal.Length Sepal.Width Petal.Length Petal.Width   
## Min. :4.300 Min. :2.000 Min. :1.000 Min. :0.100   
## 1st Qu.:5.100 1st Qu.:2.800 1st Qu.:1.600 1st Qu.:0.300   
## Median :5.800 Median :3.000 Median :4.350 Median :1.300   
## Mean :5.843 Mean :3.057 Mean :3.758 Mean :1.199   
## 3rd Qu.:6.400 3rd Qu.:3.300 3rd Qu.:5.100 3rd Qu.:1.800   
## Max. :7.900 Max. :4.400 Max. :6.900 Max. :2.500   
## Species   
## Length:150   
## Class :character   
## Mode :character   
##   
##   
##

str(iris)

## 'data.frame': 150 obs. of 5 variables:  
## $ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...  
## $ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...  
## $ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...  
## $ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...  
## $ Species : chr "setosa" "setosa" "setosa" "setosa" ...

**接下來會教一些常用的小技巧：**

# add new column  
iris$Petal\_area <- iris$Petal.Length \* iris$Petal.Width  
head(iris)

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species Petal\_area  
## 1 5.1 3.5 1.4 0.2 setosa 0.28  
## 2 4.9 3.0 1.4 0.2 setosa 0.28  
## 3 4.7 3.2 1.3 0.2 setosa 0.26  
## 4 4.6 3.1 1.5 0.2 setosa 0.30  
## 5 5.0 3.6 1.4 0.2 setosa 0.28  
## 6 5.4 3.9 1.7 0.4 setosa 0.68

# add new column with condition  
iris$isLonger <- ifelse(iris$Petal.Length > 2, TRUE, FALSE)  
head(iris)

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species Petal\_area  
## 1 5.1 3.5 1.4 0.2 setosa 0.28  
## 2 4.9 3.0 1.4 0.2 setosa 0.28  
## 3 4.7 3.2 1.3 0.2 setosa 0.26  
## 4 4.6 3.1 1.5 0.2 setosa 0.30  
## 5 5.0 3.6 1.4 0.2 setosa 0.28  
## 6 5.4 3.9 1.7 0.4 setosa 0.68  
## isLonger  
## 1 FALSE  
## 2 FALSE  
## 3 FALSE  
## 4 FALSE  
## 5 FALSE  
## 6 FALSE

# add blank column  
iris$blank <- "" # also can replace with 0  
head(iris)

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species Petal\_area  
## 1 5.1 3.5 1.4 0.2 setosa 0.28  
## 2 4.9 3.0 1.4 0.2 setosa 0.28  
## 3 4.7 3.2 1.3 0.2 setosa 0.26  
## 4 4.6 3.1 1.5 0.2 setosa 0.30  
## 5 5.0 3.6 1.4 0.2 setosa 0.28  
## 6 5.4 3.9 1.7 0.4 setosa 0.68  
## isLonger blank  
## 1 FALSE   
## 2 FALSE   
## 3 FALSE   
## 4 FALSE   
## 5 FALSE   
## 6 FALSE

# remove multiple columns  
iris <- iris[,!colnames(iris) %in% c("isLonger", "blank")]  
head(iris)

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species Petal\_area  
## 1 5.1 3.5 1.4 0.2 setosa 0.28  
## 2 4.9 3.0 1.4 0.2 setosa 0.28  
## 3 4.7 3.2 1.3 0.2 setosa 0.26  
## 4 4.6 3.1 1.5 0.2 setosa 0.30  
## 5 5.0 3.6 1.4 0.2 setosa 0.28  
## 6 5.4 3.9 1.7 0.4 setosa 0.68

# remove single column  
iris$Petal\_area <- NULL  
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

# rename column  
colnames(iris)

## [1] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width"   
## [5] "Species"

colnames(iris)[1] <- "Sepal\_Length"  
colnames(iris)

## [1] "Sepal\_Length" "Sepal.Width" "Petal.Length" "Petal.Width"   
## [5] "Species"

colnames(iris)[1:4] <-c("Sepal\_Length", "Sepal\_Width", "Petal\_Length", "Petal\_Width")  
colnames(iris)

## [1] "Sepal\_Length" "Sepal\_Width" "Petal\_Length" "Petal\_Width"   
## [5] "Species"

# split dataframe by column  
iris\_Sepal <- iris[,c("Sepal\_Length", "Sepal\_Width")]  
head(iris\_Sepal)

## Sepal\_Length Sepal\_Width  
## 1 5.1 3.5  
## 2 4.9 3.0  
## 3 4.7 3.2  
## 4 4.6 3.1  
## 5 5.0 3.6  
## 6 5.4 3.9

iris\_Petal <- iris[,c("Petal\_Length", "Petal\_Width")]  
head(iris\_Petal)

## Petal\_Length Petal\_Width  
## 1 1.4 0.2  
## 2 1.4 0.2  
## 3 1.3 0.2  
## 4 1.5 0.2  
## 5 1.4 0.2  
## 6 1.7 0.4

# merge two dataframe using column bind  
iris\_flower <- cbind(iris\_Sepal, iris\_Petal)  
head(iris\_flower)

## Sepal\_Length Sepal\_Width Petal\_Length Petal\_Width  
## 1 5.1 3.5 1.4 0.2  
## 2 4.9 3.0 1.4 0.2  
## 3 4.7 3.2 1.3 0.2  
## 4 4.6 3.1 1.5 0.2  
## 5 5.0 3.6 1.4 0.2  
## 6 5.4 3.9 1.7 0.4

# split dataframe by row  
iris\_top\_5 <- iris[1:5,]  
head(iris\_top\_5, n = 5)

## 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

iris\_bottom\_5 <- tail(iris, n=5)  
head(iris\_bottom\_5, n=5)

## Sepal\_Length Sepal\_Width Petal\_Length Petal\_Width Species  
## 146 6.7 3.0 5.2 2.3 virginica  
## 147 6.3 2.5 5.0 1.9 virginica  
## 148 6.5 3.0 5.2 2.0 virginica  
## 149 6.2 3.4 5.4 2.3 virginica  
## 150 5.9 3.0 5.1 1.8 virginica

# merge two dataframe using row bind  
iris\_row\_bind <- rbind(iris\_top\_5, iris\_bottom\_5)  
head(iris\_row\_bind)

## 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  
## 146 6.7 3.0 5.2 2.3 virginica

### Plot

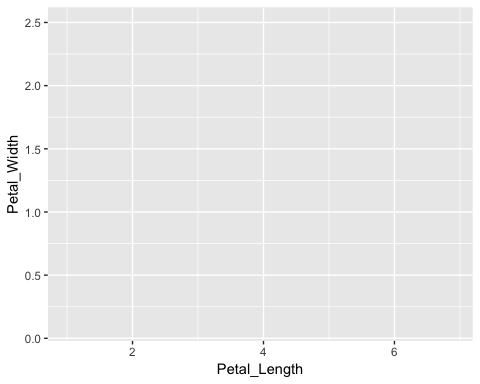
這邊推薦兩個很棒的網站，有助於大家繪製心目中的圖。

* [The R Graph Gallery](https://www.r-graph-gallery.com/)
* [STHDA](http://www.sthda.com/english/)

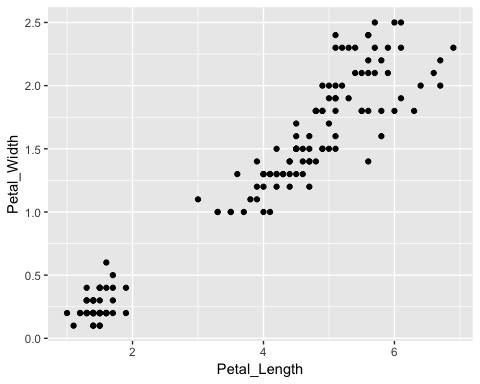
這邊要跟大家先說明，繪圖可能是R語言中最困難的部分，有很多的參數以及功能需要靠大家多使用才能熟練。雖然有人為了使用者開發了許多的package，可是仍有很多東西需要記憶，因此本堂課我會教大家必要的東西，其他的一些功能需要大家多多利用上面兩個網址去做詳細的參閱。

#### Scatter plot

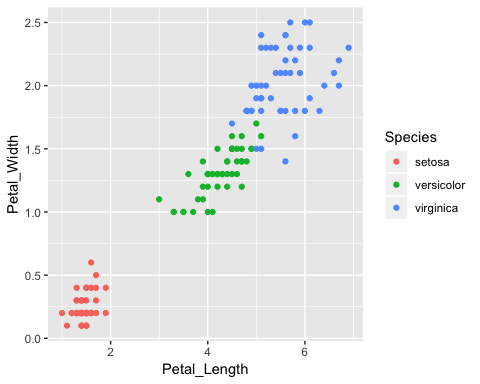
# 基本起手式，產生畫布  
library(ggplot2)  
g <- ggplot(data=iris, # indicate which data you want to use  
 aes(x=Petal\_Length, y=Petal\_Width)) # aes indicate x and y axis  
print(g)



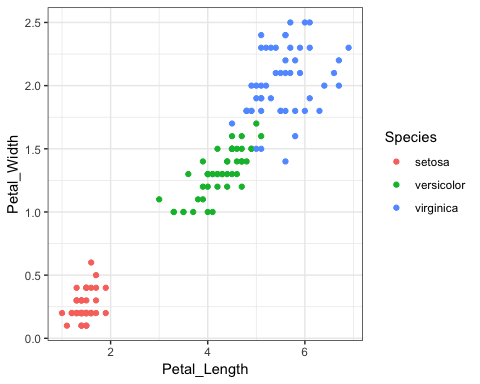
g + geom\_point() # 明確指明你要做哪一種繪圖方式，這邊選取的事散步圖或點圖



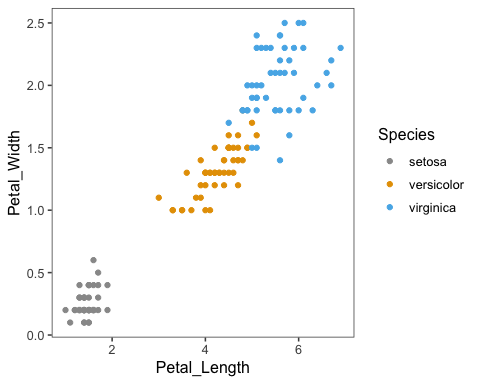
g + geom\_point(aes(color=Species)) # 由於我們有三種species，我們可以透過標記顏色觀察其分佈



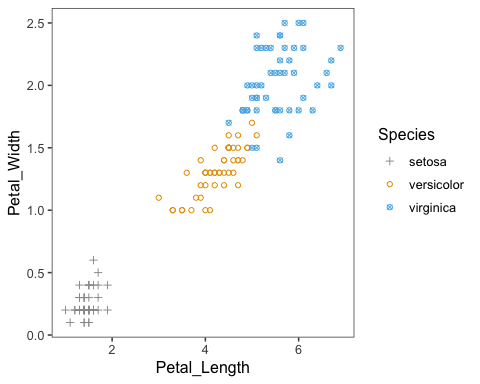
# 是不是覺得背景很煩呢？，這邊我們提供幾個別人設計好的模板  
library(ggthemes) # 提供數個模板，模板的樣式都是 theme\_????   
g + geom\_point(aes(color=Species)) + theme\_bw()

 **補充資料**：[ggthemes](https://www.rdocumentation.org/packages/ggthemes/versions/3.5.0)

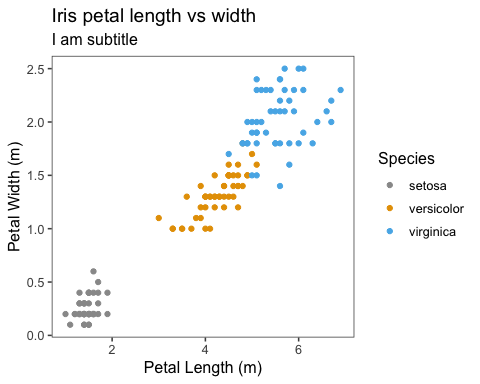
# fill color  
g + geom\_point(aes(color=Species)) + theme\_few() +  
 scale\_color\_manual(values=c("#999999", "#E69F00", "#56B4E9"))



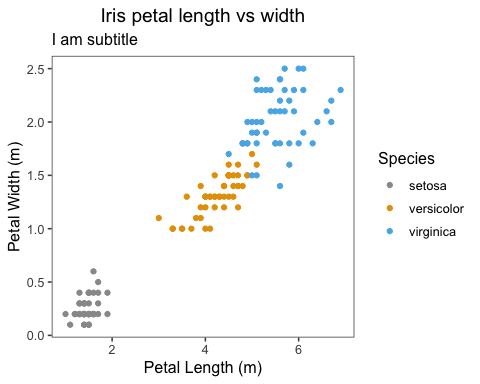
# chage shape  
g + geom\_point(aes(color=Species, shape=Species)) + theme\_few() +  
 scale\_color\_manual(values=c("#999999", "#E69F00", "#56B4E9")) +  
 scale\_shape\_manual(values=c(3,1,13))



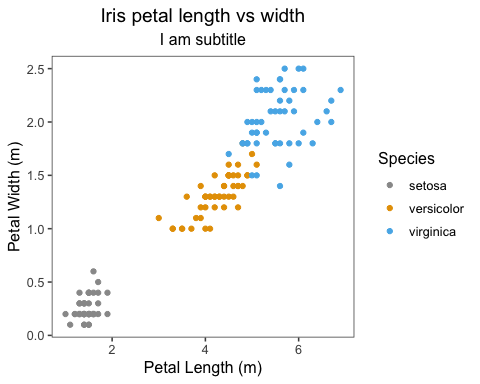
# modify plot and axis title  
g + geom\_point(aes(color=Species)) + theme\_few() +  
 scale\_color\_manual(values=c("#999999", "#E69F00", "#56B4E9")) +  
 labs(x="Petal Length (m)", y="Petal Width (m)",  
 title="Iris petal length vs width", subtitle = "I am subtitle")



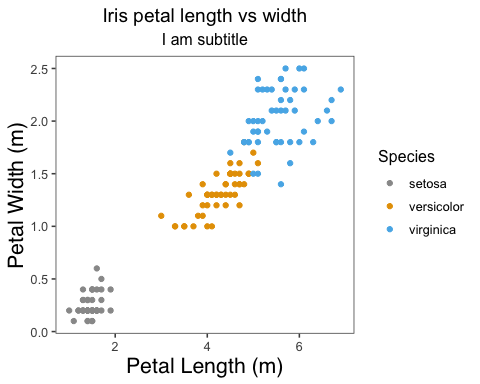
# move plot title to middle of plot  
g + geom\_point(aes(color=Species)) + theme\_few() +  
 scale\_color\_manual(values=c("#999999", "#E69F00", "#56B4E9")) +  
 labs(x="Petal Length (m)", y="Petal Width (m)",  
 title="Iris petal length vs width", subtitle = "I am subtitle") +  
 theme(plot.title = element\_text(hjust = .5)) ####



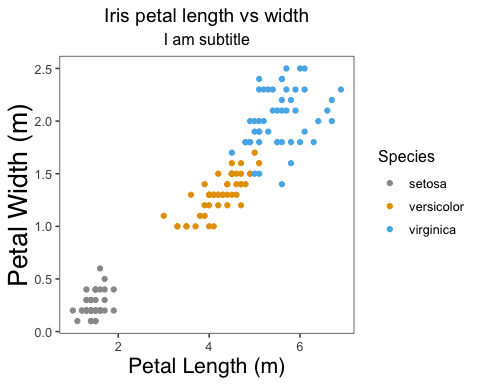
# move plot title to middle of plot  
g + geom\_point(aes(color=Species)) + theme\_few() +  
 scale\_color\_manual(values=c("#999999", "#E69F00", "#56B4E9")) +  
 labs(x="Petal Length (m)", y="Petal Width (m)",  
 title="Iris petal length vs width", subtitle = "I am subtitle") +  
 theme(plot.title = element\_text(hjust = .5),  
 plot.subtitle = element\_text(hjust = .5)) ####



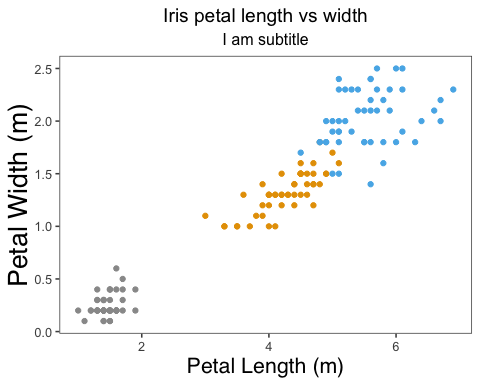
# modify axis title  
g + geom\_point(aes(color=Species)) + theme\_few() +  
 scale\_color\_manual(values=c("#999999", "#E69F00", "#56B4E9")) +  
 labs(x="Petal Length (m)", y="Petal Width (m)",  
 title="Iris petal length vs width", subtitle = "I am subtitle") +  
 theme(plot.title = element\_text(hjust = .5),  
 plot.subtitle = element\_text(hjust = .5),  
 axis.title = element\_text(family="Helvetica", size = 16)) ####



# modify x and y axis respectively  
g + geom\_point(aes(color=Species)) + theme\_few() +  
 scale\_color\_manual(values=c("#999999", "#E69F00", "#56B4E9")) +  
 labs(x="Petal Length (m)", y="Petal Width (m)",  
 title="Iris petal length vs width", subtitle = "I am subtitle") +  
 theme(plot.title = element\_text(hjust = .5),  
 plot.subtitle = element\_text(hjust = .5),  
 axis.title = element\_text(family="Helvetica"), ####  
 axis.title.x = element\_text(size = 16), ####  
 axis.title.y = element\_text(size = 20)) ####

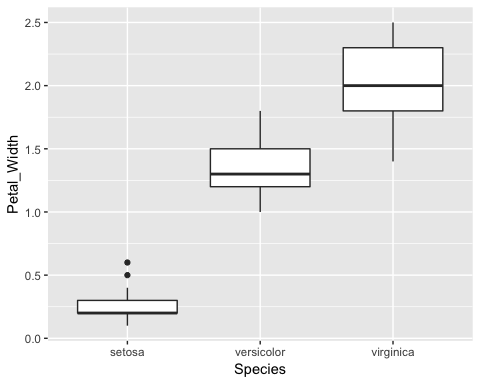


# hide legend  
final <- g + geom\_point(aes(color=Species)) + theme\_few() +  
 scale\_color\_manual(values=c("#999999", "#E69F00", "#56B4E9")) +  
 labs(x="Petal Length (m)", y="Petal Width (m)",  
 title="Iris petal length vs width", subtitle = "I am subtitle") +  
 theme(plot.title = element\_text(hjust = .5),  
 plot.subtitle = element\_text(hjust = .5),  
 axis.title = element\_text(family="Helvetica"),  
 axis.title.x = element\_text(size = 16),  
 axis.title.y = element\_text(size = 20),  
 legend.position = "none") # also can use c(x\_position, y\_position) to change position  
print(final)

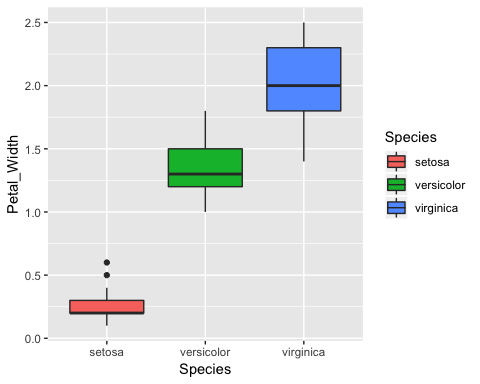


#### Box plot

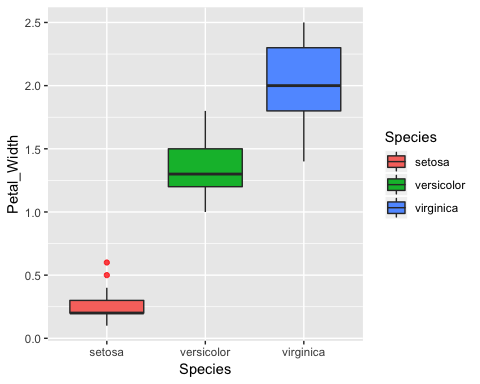
# set x and y axis data  
b <- ggplot(data=iris, aes(x=Species, y=Petal\_Width))  
  
# boxplot  
b + geom\_boxplot()



# fill colour into boxplot  
b + geom\_boxplot(aes(fill=Species))



# change outlier colour  
b + geom\_boxplot(aes(fill=Species),  
 outlier.color = "red",  
 outlier.alpha = .75)

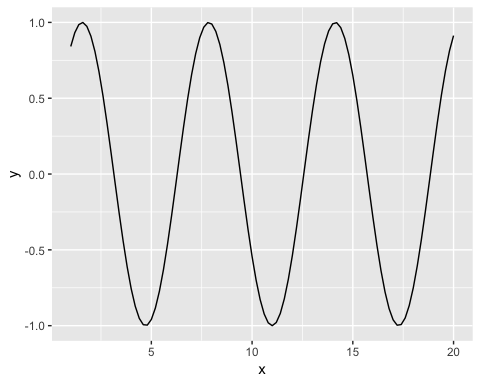


#### Line chart

line\_segment <- pretty(c(1:20), 100)  
y <- sin(line\_segment)  
  
sin\_data <- data.frame(x=line\_segment, y=y)  
head(sin\_data)

## x y  
## 1 1.0 0.8414710  
## 2 1.2 0.9320391  
## 3 1.4 0.9854497  
## 4 1.6 0.9995736  
## 5 1.8 0.9738476  
## 6 2.0 0.9092974

ggplot(sin\_data, aes(x=x, y=y)) + geom\_line()



#### Save plot

圖表的可以儲存成pdf(), jpeg(), tiff()…，這邊為大家介紹兩個最常見的格式tiff and pdf。

output\_pdf\_path <- "./iris.pdf"  
# save as pdf  
pdf(file = output\_pdf\_path,  
 width = 5, height = 5)  
print(final)  
dev.off()

## quartz\_off\_screen   
## 2

output\_tiff\_path <- "./iris.tiff"  
# save as tiff  
tiff(file = output\_tiff\_path,  
 width = 5, height = 5, units = "in", res = 300)  
print(final)  
dev.off()

## quartz\_off\_screen   
## 2

#### Remind

由於這次內容也很多，希望同學可以花一點時間理解，在後續課程我會使用今天上課的內容，程式要常用才會記得住，有時候可以試著不使用excel，使用R來分析看看，加深自己的印象。如果有問題的話可以寄E-mail。 E-mail : [steve24563@gmail.com](mailto:steve24563@gmail.com)