Titanic 鐵達尼 - R 資料視覺化

李智慎 副統計分析師

2016 年 12 月 19 日

為何需要資料視覺化?

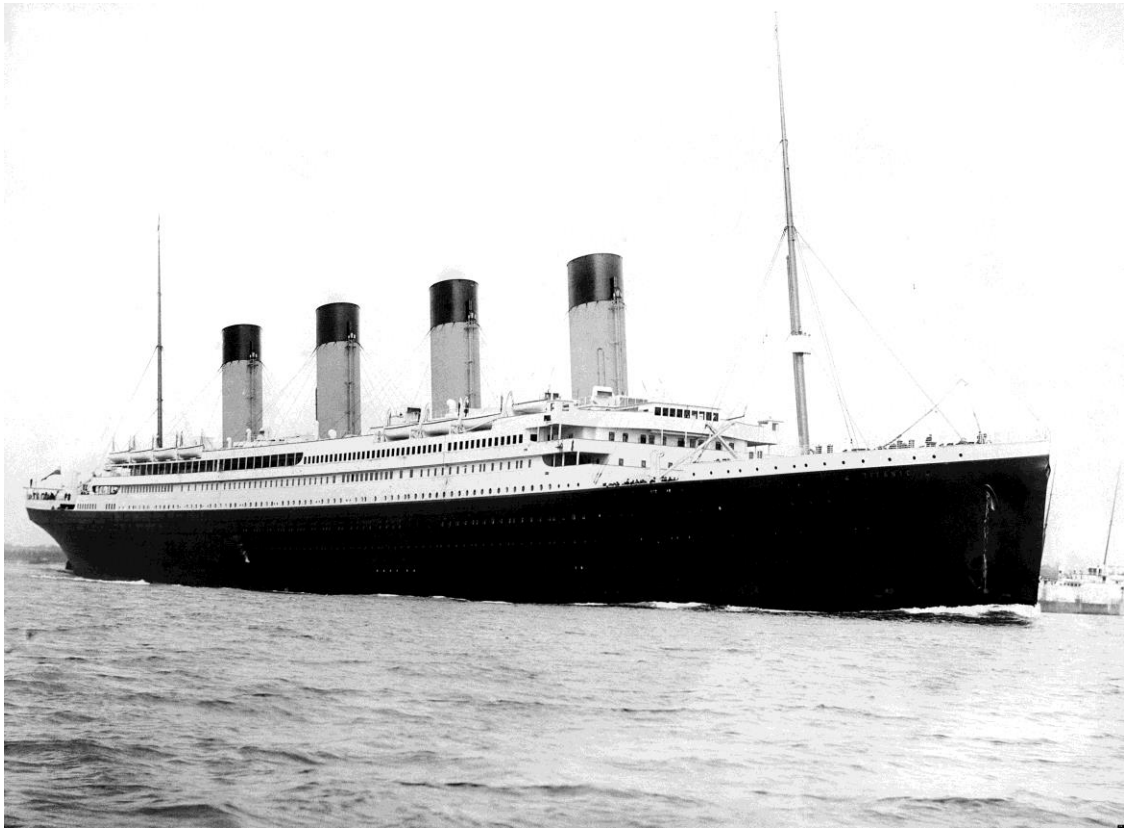
將資料做視覺化呈現能夠使人們很快感受資料整體訊息，利用顏色和圖形可使人更容易抓到資料的趨勢或重點。

R 套件安裝

在執行下列程式前，我們先安裝將會用到的套件並引入，其中 **titanic** 為可呼叫鐵達尼資料用的套件，**dplyr** 為資料處理套件，**ggplot2** 為資料視覺化套件，安裝程式碼如下

```
packageNames <- c("titanic", "dplyr", "ggplot2")
install.pacakges(pacakgeNames)
```

Titanic 鐵達尼乘客資料



鐵達尼號電影曾經風靡全世界，讀者可於維基百科查詢到鐵達尼沈船的詳細情形，以下我們引用維基百科資料進行簡單整理，作為資料視覺化的呈現範例。

鐵達尼號是一艘奧林匹克級郵輪，是當時最大的客運輪船，但因為人為錯誤，於 1912 年 4 月 14 日 23 點 40 分撞上冰山，事發 2 小時 40 分鐘後，即 4 月 15 日凌晨 02 點 20 分，船裂成兩半後沉入大西洋，死亡人數超越 1500 人，堪稱 20 世紀最大的海難事件，同時也是最廣為人知的海難之一。

這期 eNews 將使用 Titanic 乘客的資料做視覺化示範，首先，在 R 裡可用 library(titanic)引入套件。

```
library(titanic)
```

引入 **titanic** 套件後，即可呼叫資料 **titanic_train** 和 **titanic_test**，在此先用 **dplyr** 套件裡的 **glimpse** 指令觀看資料整體結構，記得使用套件內指令之前必須先引入套件(在"##"後的文字為指令執行結果)。

```
library(dplyr)

##
## 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

dim(titanic_train)

## [1] 891  12
```

由 dim( )的結果可知 titanic_train 有 891 筆資料 12 個變數，對於一般套件內建的資料通常都可用查詢 help( )的方式，來得知資料更多的資訊，例如想知道 titanic_train 資料的更多資訊，我們可執行「help(titanic_train)」或「?titanic_train」，用以查詢 titanic_train 相關資訊的網頁。

```
help(titanic)
?titanic
```

Titanic 資料來源為 - https://www.kaggle.com/c/titanic/data，可從 **Kaggle** 網站內得到變數更詳盡的資訊，其變數內容如下所示：

變數序	變數類別	變數名稱	變數內容
1	數值	PassengerId	編號
2	類別	Survived	是否存活 (0:否 1:是)
3	類別	Pclass	社會經濟地位等級 (1:上等 2:中等 3:低等)
4	字串	Name	姓名
5	類別	Sex	性別 (male:男 female:女)
6	數值	Age	年齡(年齡小於 1 會有小數點，預估的年齡以'XX.5'表示)

7	數值	SibSp	在船上的兄弟姊妹及配偶總數
8	數值	Parch	在船上的父母家長及子女總數
9	字串	Ticket	船票編號
10	數值	Fare	票價
11	字串	Cabin	座艙編號
12	類別	Embarked	登船港口 (C:瑟堡-法國城鎮 Q:皇后鎮-紐西蘭城市 S:南安普敦-英格蘭城市)

在套件內還有一資料 titanic_test 是 titanic_train 建模後用於預測使用，將兩組資料用 dplyr 的 bind_rows 指令合併，作為接下來要畫圖用的示範資料 titanic_all，其合併程式碼如下所示：

```
titanic_all <- bind_rows(titanic_train, titanic_test)
```

以下的 **R** 視覺圖形會以 **titanic_all** 資料進行示範。

---

1. Jack?Rose?在哪?

我們拿到乘客資料後，最想知道的關鍵問題，就是 Jack Dawson 和 Rose DeWitt Bukater 究竟登船與否。此時，我們可使用 R 的基本內建指令 grep( )來抓取內含 Jack 或 Rose 條件的名字並將其列出，程式碼如下：

```
grep("Jack|Rose", titanic_all$Name, value = TRUE)

## [1] "Brewe, Dr. Arthur Jackson"
## [2] "Aks, Mrs. Sam (Leah Rosen)"
## [3] "Rosenbaum, Miss. Edith Louise"
## [4] "Rosenshine, Mr. George (Mr George Thorne\")\""
```

結果證明資料裡無 Jack 和 Rose 這號人物，所以就讓我們認份的來畫圖吧！

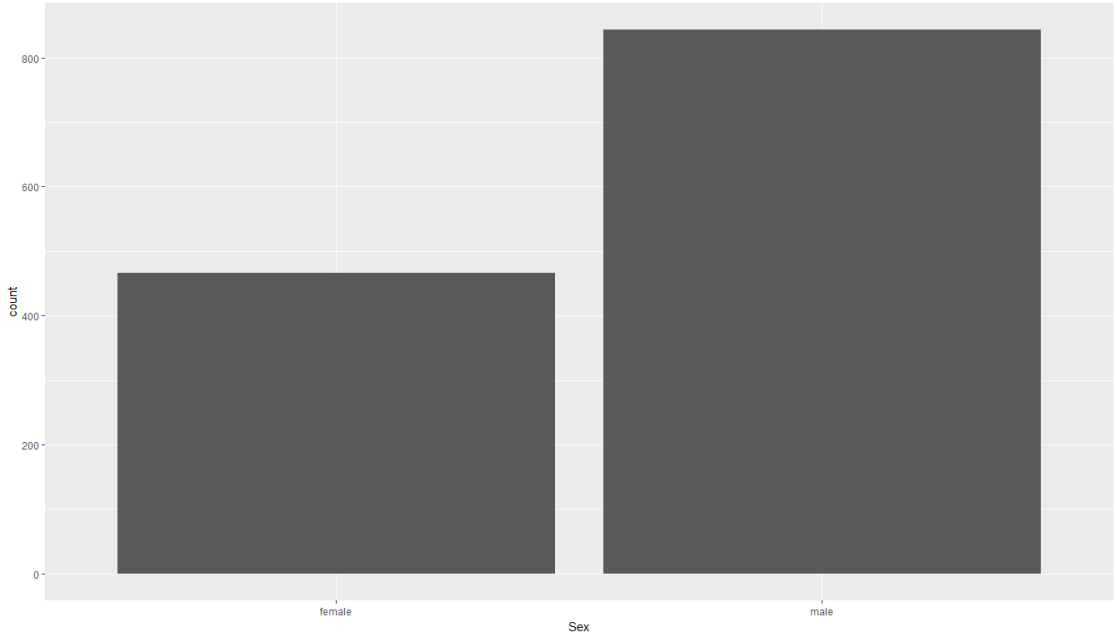
2. 長條圖

讀者欲使用套件功能時，記得要先引入套件

```
library(ggplot2)
```

ggplot2 套件使用起來相當直觀，舉例來說，乘客男女人數統計作長條圖，程式碼如下

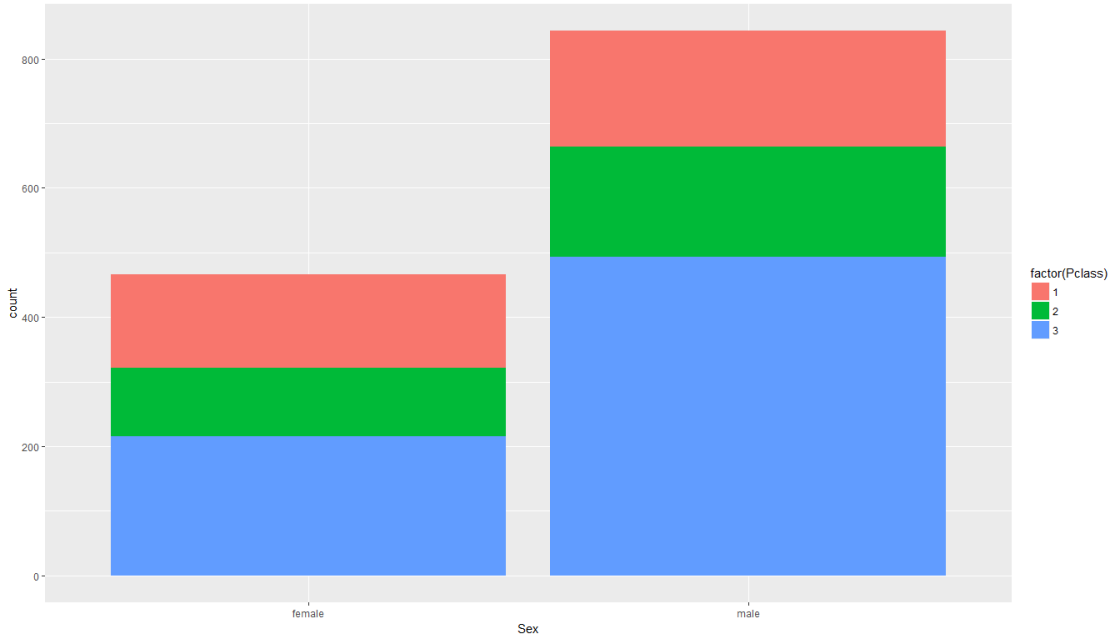
```
ggplot(data = titanic_all, aes(x = Sex)) +
  geom_bar()
```



上圖使用 ggplot( )將 data 參數設為 titanic_all，而 aes( )內則可設定 X 軸、Y 軸和其它圖形元素所對應的變數。上例將 Sex 變數指派為 X 軸，因 Sex 有 female 和 male 兩類別，且使用 geom_bar( )構圖，所以圖形會以長條狀(Bar)的幾何圖形(Geometry)呈現其結果，而高度所代表的為資料裡各性別的總筆數，在 ggplot2 套件裡每個繪圖元件彼此是用+符號做連接，這就是 ggplot2 基本寫法。

如果我們想再做更細部的分層，例如使用 Pclass 指令將男女切分不同等級作人數統計，那在 ggplot2 該怎麼做呢?其實非常簡單，只需在 aes( )內多加入 fill 參數的設定即可

```
ggplot(data = titanic_all, aes(x = Sex, fill = factor(Pclass))) +
  geom_bar()
```



如上圖所示，我們將 fill 參數設為 factor(Pclass)，factor( )為 R 基本內建的指令，可將變數轉變為類別型態，因 Pclass 原本為 int 也就是數值而且是整數的數值型態，可用 dplyr 套件裡的 glimpse( )指令觀看 titanic_all 資料

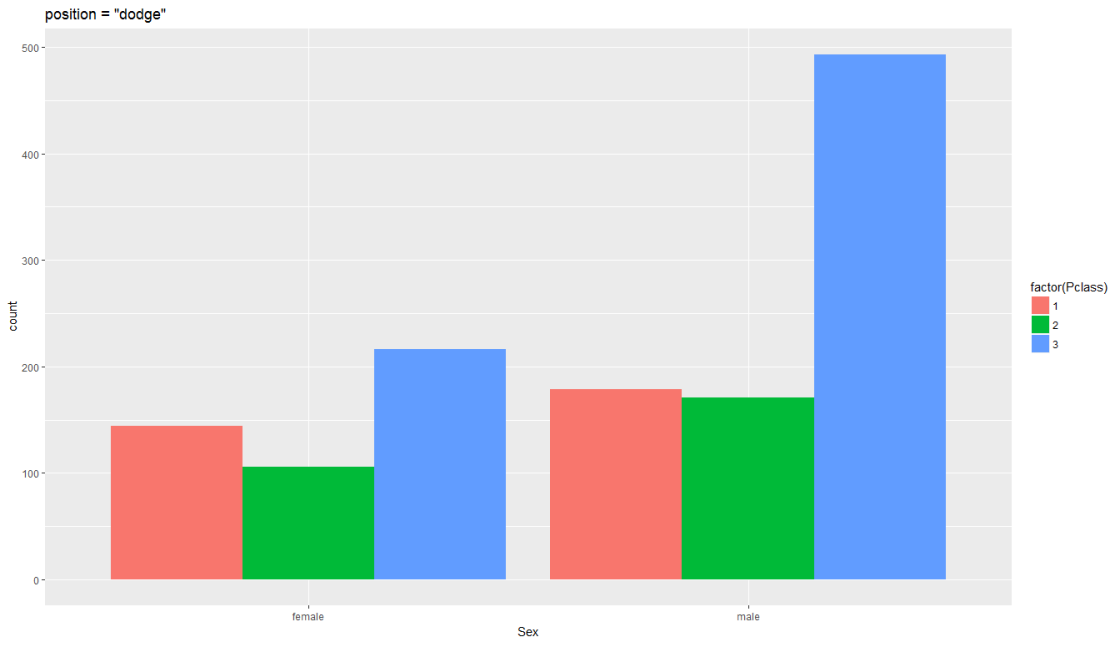
```
glimpse(titanic_all)

## Observations: 1,309
## Variables: 12
## $ PassengerId <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,...
## $ Survived    <int> 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0,...
## $ Pclass      <int> 3, 1, 3, 1, 3, 3, 1, 3, 3, 2, 3, 1, 3, 3, 3, 2, 3,...
## $ Name        <chr> "Braund, Mr. Owen Harris", "Cumings, Mrs. John Bra...
## $ Sex         <chr> "male", "female", "female", "female", "male", "mal...
## $ Age         <dbl> 22, 38, 26, 35, 35, NA, 54, 2, 27, 14, 4, 58, 20, ...
## $ SibSp       <int> 1, 1, 0, 1, 0, 0, 0, 3, 0, 1, 1, 0, 0, 1, 0, 0, 4,...
## $ Parch       <int> 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 1, 0, 0, 5, 0, 0, 1,...
## $ Ticket      <chr> "A/5 21171", "PC 17599", "STON/O2. 3101282", "1138...
## $ Fare        <dbl> 7.2500, 71.2833, 7.9250, 53.1000, 8.0500, 8.4583, ...
## $ Cabin       <chr> "", "C85", "", "C123", "", "", "E46", "", "", "", ...
## $ Embarked    <chr> "S", "C", "S", "S", "S", "Q", "S", "S", "S", "C", ...
```

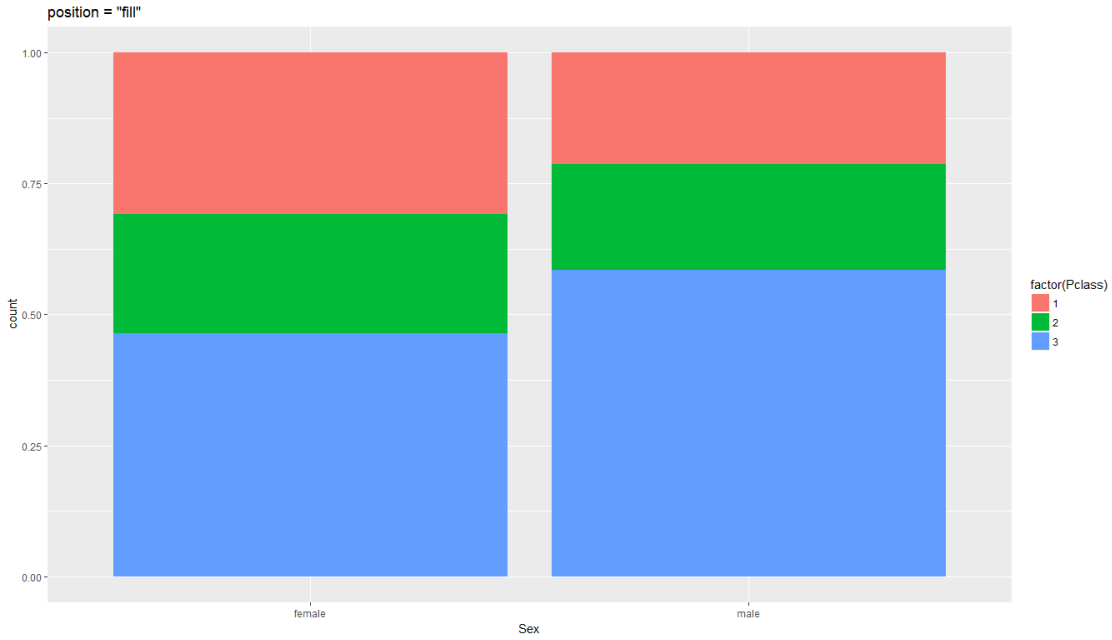
從預覽中可看到資料的總比數(Observations)、變數個數(Variables)、變數名($ name)、變數型態(<type>)和各變數前幾筆資料，其中為字符，為雙精度數值，如有類別型變數會以<fctr>顯示。

上例為堆疊模式的長條圖，而 ggplot 可以呈現多種模式，只需在 geom_bar( )內設定 position 參數即可，舉例如下

```
ggplot(data = titanic_all, aes(x = Sex, fill = factor(Pclass))) +
  geom_bar(position = "dodge") +
  ggtitle('position = "dodge"')
```




```
ggplot(data = titanic_all, aes(x = Sex, fill = factor(Pclass))) +
  geom_bar(position = "fill") +
  ggtitle('position = "fill"')
```



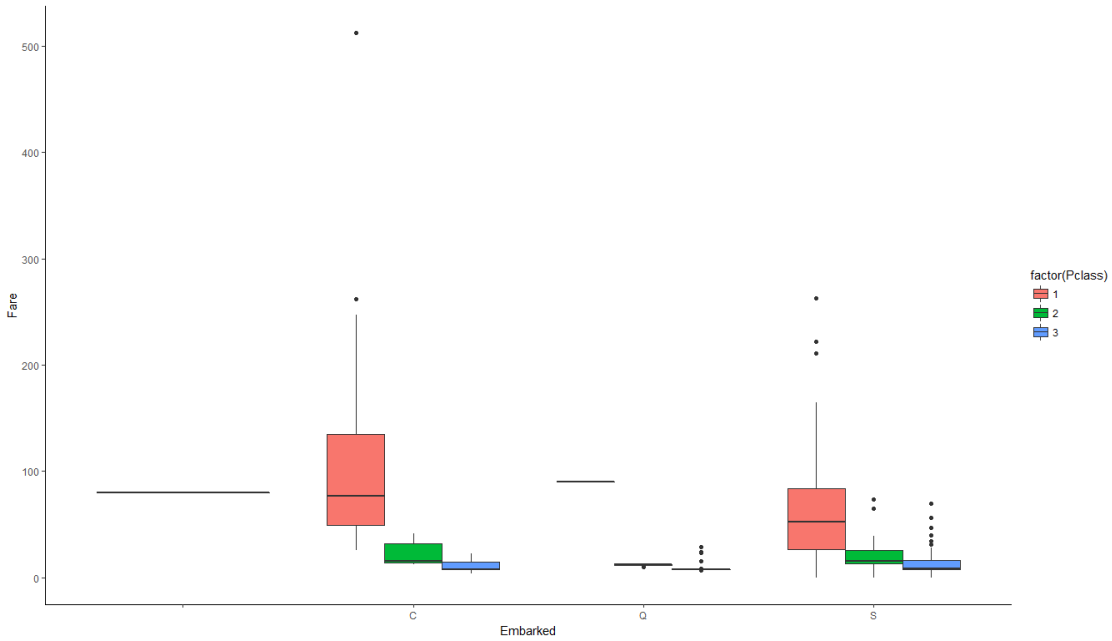
可看到"dodge"可使長條左右並列，"fill"則會以比例堆疊長條，而在上例的程式中還多加了 ggtitle( )指令，可在圖形上方多加標題。

盒鬚圖

使用 ggplot 畫盒鬚圖也非常簡單，舉例如下

```
ggplot(titanic_all, aes(x = Embarked, y = Fare, fill = factor(Pclass))) +
  geom_boxplot() +
  theme_classic()

## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```



上圖 X 軸為 Embarked 不同登船口，其中第一項為沒有登船港口資料的乘客故沒有名稱，Y 軸為票價，而程式裡有將 fill 設為 factor(Pclass)，所以圖形又會將各港口的乘客以不同的社經地位分割做盒鬚圖。總體而言，從瑟堡(C)登船的上等(1)乘客在票價(Fare)花費上較其他上等乘客高，而且其票價最高值遠遠高於其他人，可能間接顯示了當時法國貴族的經濟水平是比英國貴族高上一些，尤其是金字塔頂端的貴族。

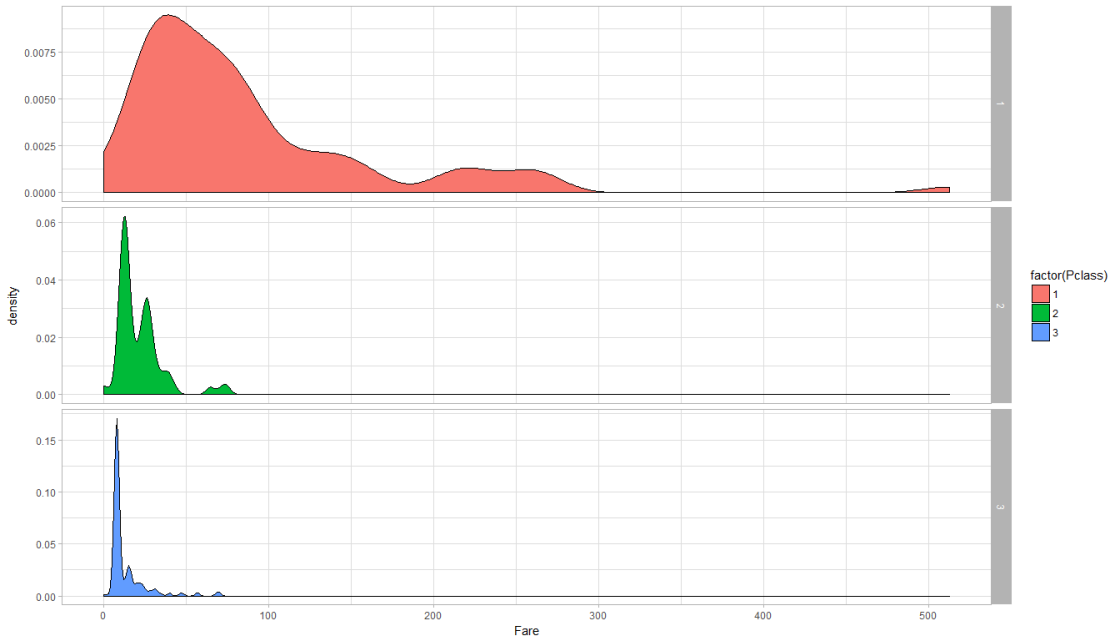
而在程式後方多加了 theme_classic( )，可為圖形變換樣式，在 ggplot 裡還有多種已經調配好的模板可供使用。

密度曲線圖

在 ggplot 套件可使用 geom_density( )做密度曲線圖，舉例如下

```
ggplot(titanic_all, aes(x = Fare, fill = factor(Pclass))) +
  geom_density() +
  theme_light() +
  facet_grid(Pclass ~ ., scales = "free")

## Warning: Removed 1 rows containing non-finite values (stat_density).
```



此範例是將對不同社經地位的乘客對他們票價變數做密度曲線，所以 X 軸令為 Fare，而 fill 曲線填充顏色則令為 factor(Pclass)，而較不一樣的地方在於這次多使用了 facet_grid( )，其功用在於可依照內部參數的設定個別作圖，例如內部 Pclass ~ .的意思是以 Pclass 各因子橫列依序作圖，假如是設定~ Pclass 則會直行排列作圖，而第二參數位置 scales = "free"則是使各圖的 Y 軸尺度自動調整。

折線圖

當我們在做資料視覺化時，有一大部分的時間花費在做資料處理上，接下來的範例會利用 dplyr 套件指令做簡單的資料處理，並且利用些數值做折線圖，在此 ggplot 舉例的畫線指令為 geom_line( )。

首先我們先來看資料處理的部分，先將整理好的資料命名為 lineData，其程式碼如下

```
lineData <- titanic_train %>%
  group_by(Sex, Pclass) %>%
  summarise(SurvivedAvg = mean(Survived))
```

在此，我們欲統計不同性別和社經地位乘客的存活率，因此，我們採用 Survived 變數之中的 titanic_train 數據進行資料處理並且分析，而其程式碼的%>%、group_by( )和 summarise( )均為 dplyr 套件內的指令。

1. %>%

   連接各指令的串接符號(PIPE)，會將前方的輸出結果輸入進後方的指令中。簡單舉例，"c(1, 2, 3) %>% max"裡串接的符號會將前方輸出的序列"c(1, 2, 3)"輸入進"max( )"裡，故程式碼最後得到的結果為 5。

2. group_by( )

   此指令會依照參數所指派的變數做分組，如上例會依照 2 種性別和 3 種社經地位總共$2 \times 3 = 6$種組合，但 group_by( )大部分都要搭配 summarise( ) 進行處理。

3. summarise( )

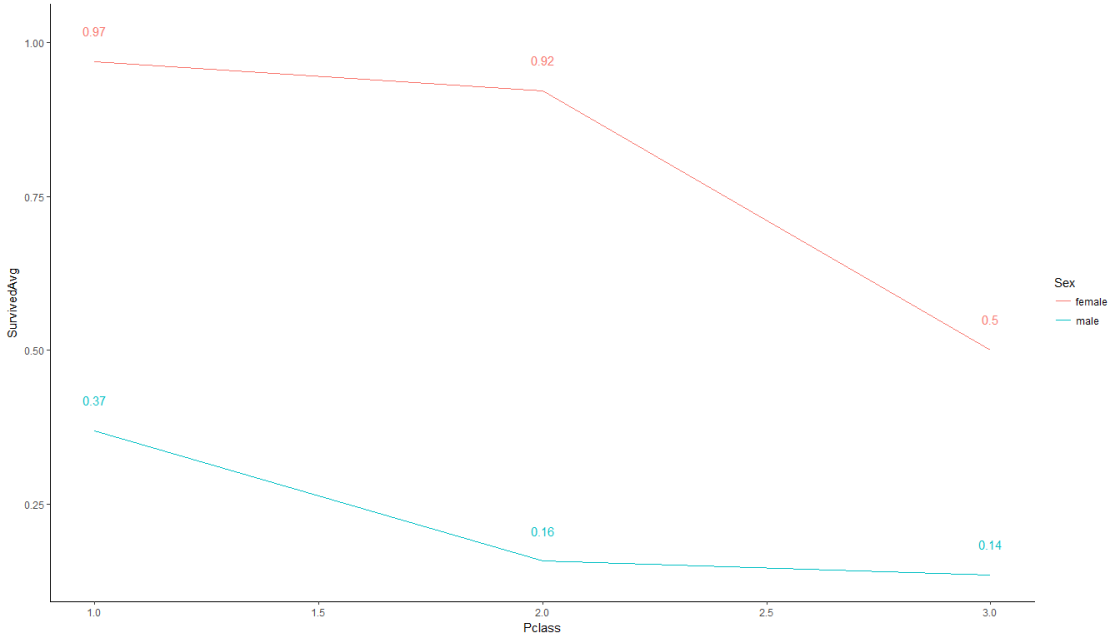
   此指令將可對 group_by( )所分的各個組別資料做計算，如同上例，Survived = mean(Survived) 之作用，就是對各組的 Survived 變數取平均值，並且再將之命名為 SurvivedAvg。

所得到的 lineData 資料內容如下

Sex	Pclass	SurvivedAvg
female	1	0.9680851
female	2	0.9210526
female	3	0.5000000
male	1	0.3688525
male	2	0.1574074
male	3	0.1354467

接著使用 **lineData** 資料做折線圖，其程式碼如下

```
ggplot(data = lineData, aes(x = Pclass, y = SurvivedAvg, color = Sex)) +
  geom_line() +
  geom_text(aes(label = round(SurvivedAvg,2)), nudge_y = 0.05, show.legend = FALSE) +
  theme_classic()
```



X 軸為社經地位，Y 軸表示為各群組乘客的存活機率 SurvivedAvg。由上圖可知，隨著社經地位的降低存活率也會顯著跟著下降，而另一方面女性的存活率不管在任何階級都遠高於男性，符合電影裡船員優先禮讓老弱婦孺先上救生船的情節。

趨勢曲線圖

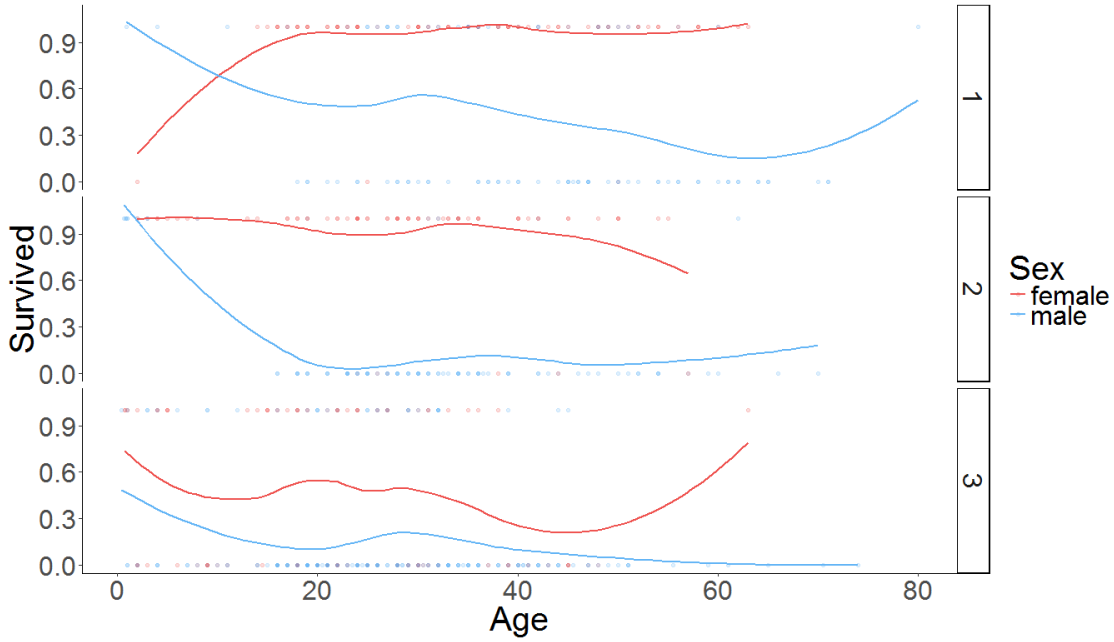
前面一個例子提到乘客的存活率，可能與老弱婦孺有關，在此，我們使用 geom_point( )和 geom_smooth( )來表示資料所呈現的趨勢，其程式碼如下

```
ggplot(data = titanic_train, aes(x = Age, y = Survived, color = Sex)) +
  geom_point(alpha = 0.2) +
  geom_smooth(se = FALSE) +
  facet_grid(Pclass ~ .) +
  theme_classic() +
  theme(text = element_text(size=30)) +
  scale_color_manual(values = c("#EF5350", "#64B5F6"))

## `geom_smooth()` using method = 'loess'

## Warning: Removed 177 rows containing non-finite values (stat_smooth).

## Warning: Removed 177 rows containing missing values (geom_point).
```



上例程式碼可將資料依顏色區別性別的不同和依社經地位不同各自做一張散佈圖，並且使用 ggplot 的 geom_smooth( )指令表現散佈圖的變化趨勢，其中參數設定 se = FALSE 為不顯現信賴區間的意思；theme(text = element_text(size = 30))的作用是調整文字的大小，在這裡是設為 30，而最後一段程式碼 scale_color_manual( )為調整 color 參數用，可看到內部 values = c("#EF5350", "#64B5F6")即是設定顏色 RGB 的 HEX 編碼，因此，各性別為以此數值做對應。

由上圖可看到小孩的生存機率確實是較為高的，且男性大概在 20 歲左右都有一小段凹處為生存機率局部低點，而社經地位中上的女性存活機率有非常高的趨勢。

總結

在這期 eNews 我們利用了鐵達尼乘客資料簡單的做了資料視覺化，但當然 R 和 ggplot2 套件可以做到的不僅僅只有這樣，但希望藉此能讓大家了解這套間操作各個繪圖元件的方式與概念。

參考網站

1. **ggplot2** - http://docs.ggplot2.org/current/
2. **dplyr** - https://cran.r-project.org/web/packages/dplyr/vignettes/introduction.html