



圖表繪製

賴璉錡

lclai.t11@o365.fcu.edu.tw

資料分析

- 資料分析是Python的一個重要且發展蓬勃的項目，將龐大且複雜的數據透過整理分類後，用易懂的方式或方法來解決問題或進行決策。
- 資料分析通常分為以下步驟
 - 定義問題
 - 蒐集數據
 - 清理數據
 - 開始資料分析
 - 數據解讀

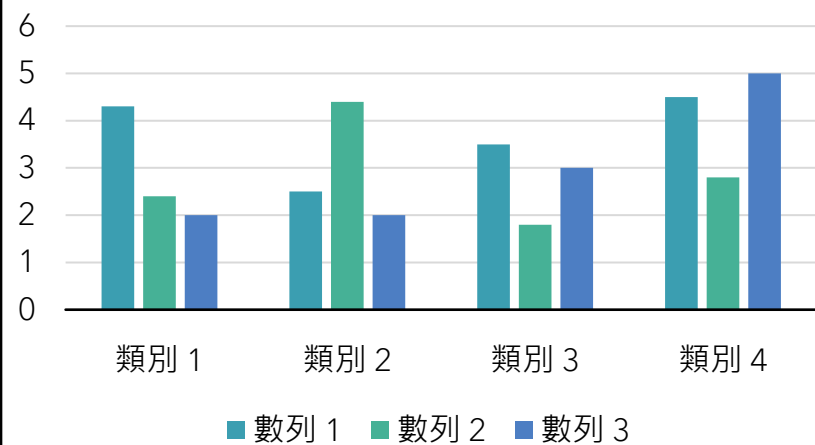
原始資料長相

- 清楚、但不易解讀
 - 無法看出趨勢
 - 對於類別或連續性的資料不好判讀

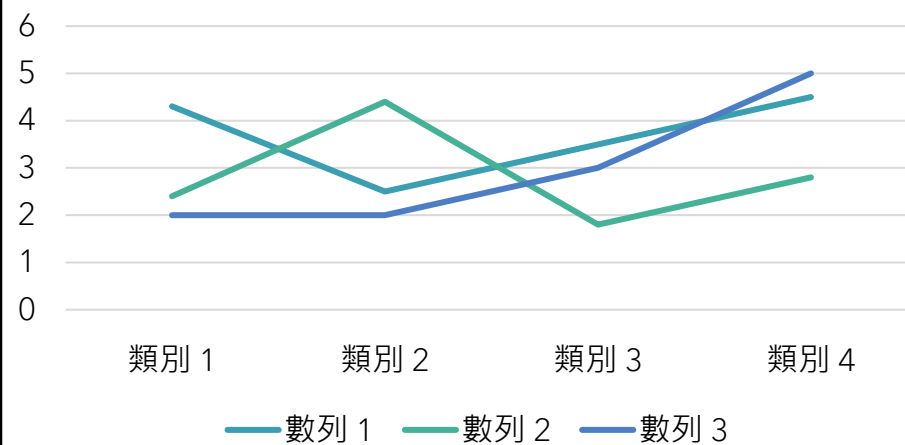
年別	類型	觀光遊憩區	細分	縣市別	1月	2月
2012	國家公園	陽明山國家公園	陽明山遊客中心	臺北市	12187	18612
2012	國家公園	陽明山國家公園	陽明書屋	臺北市	2597	3617
2012	國家公園	陽明山國家公園	陽明公園	臺北市	161000	463000
2012	國家公園	陽明山國家公園	大屯遊憩區	臺北市	17047	29079
2012	國家公園	陽明山國家公園	龍鳳谷遊憩區	臺北市	21201	22134
2012	國家公園	陽明山國家公園	小油坑遊憩區	臺北市	20839	16853
2012	國家公園	玉山國家公園	塔塔加遊憩區	南投縣	83806	59526
2012	國家公園	玉山國家公園	梅山遊客中心	高雄市	4846	1854
2012	國家公園	玉山國家公園	南安遊客中心	花蓮縣	37578	19373
2012	國家公園	玉山國家公園	玉山管理處遊客服	南投縣	13262	9460
2012	國家公園	玉山國家公園	排雲山莊	南投縣	371	80

圖表種類

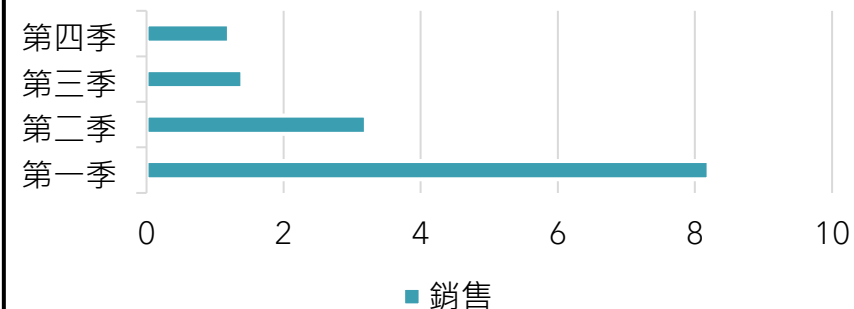
長條圖



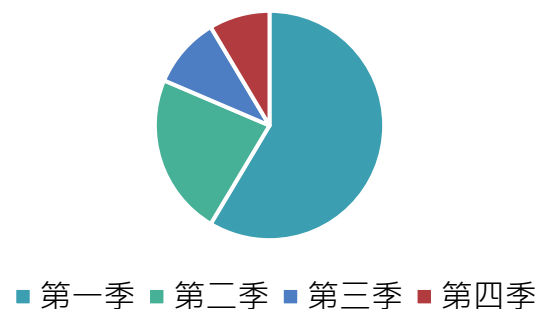
折線圖



橫條圖



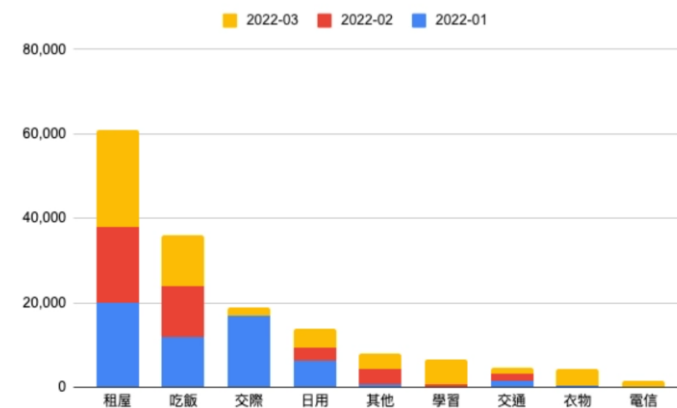
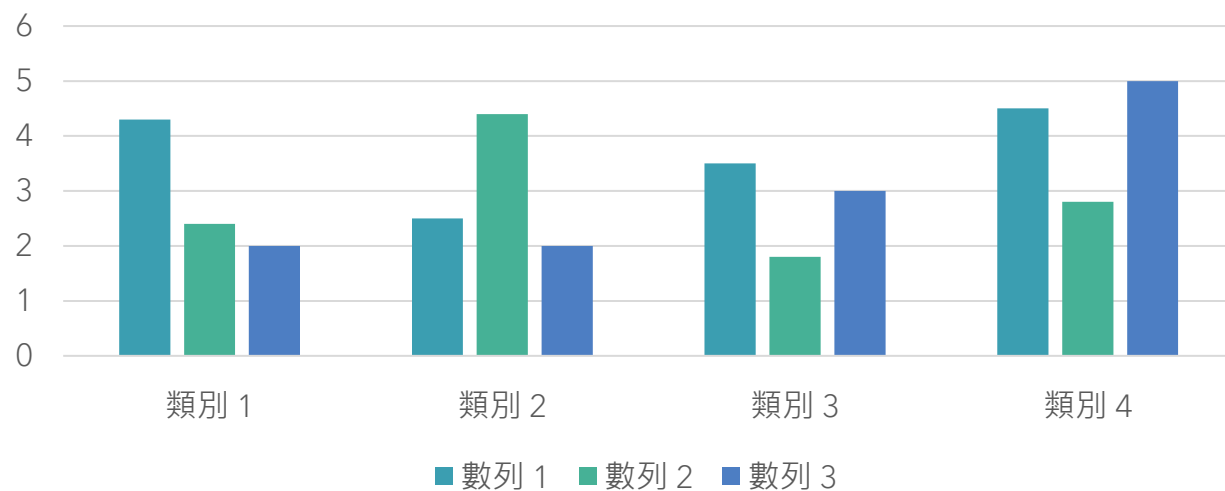
圓餅圖



長條圖

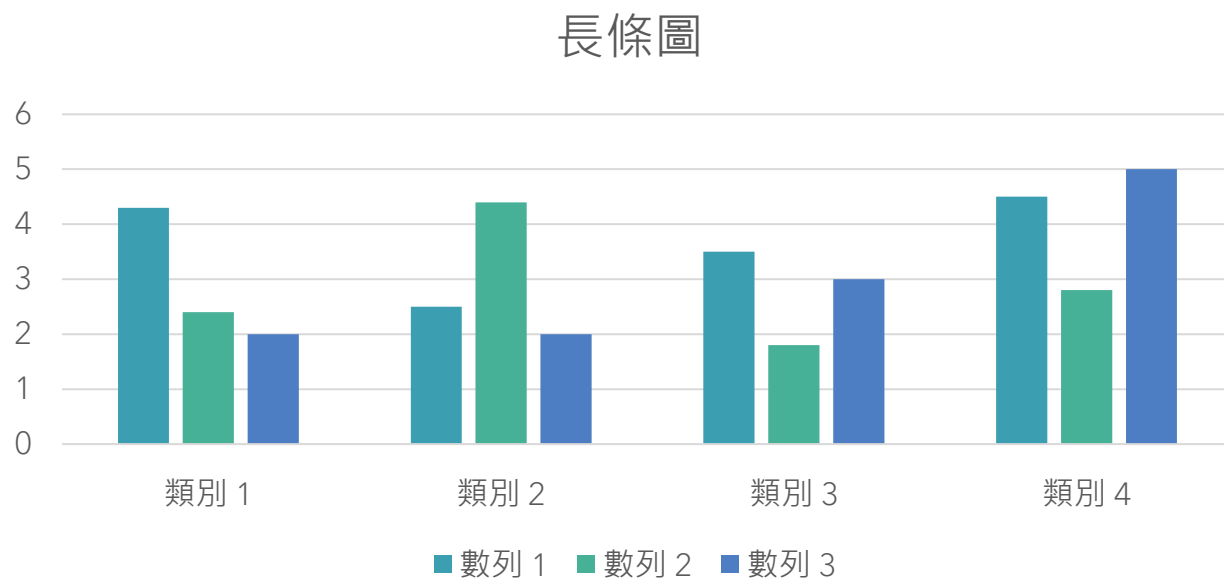
- 柱狀圖、長條圖
 - 適合呈現各資料類別的數量、或是時間日期內資料的變化
 - 類別底下如果還有其他小類別的話，這種圖也很適合

長條圖



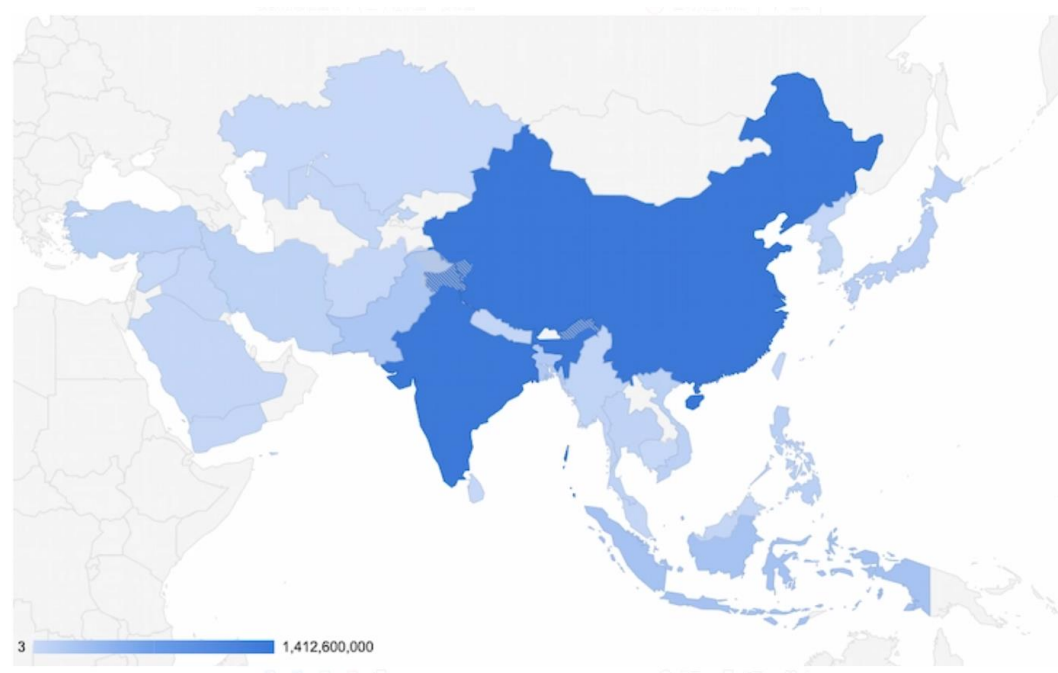
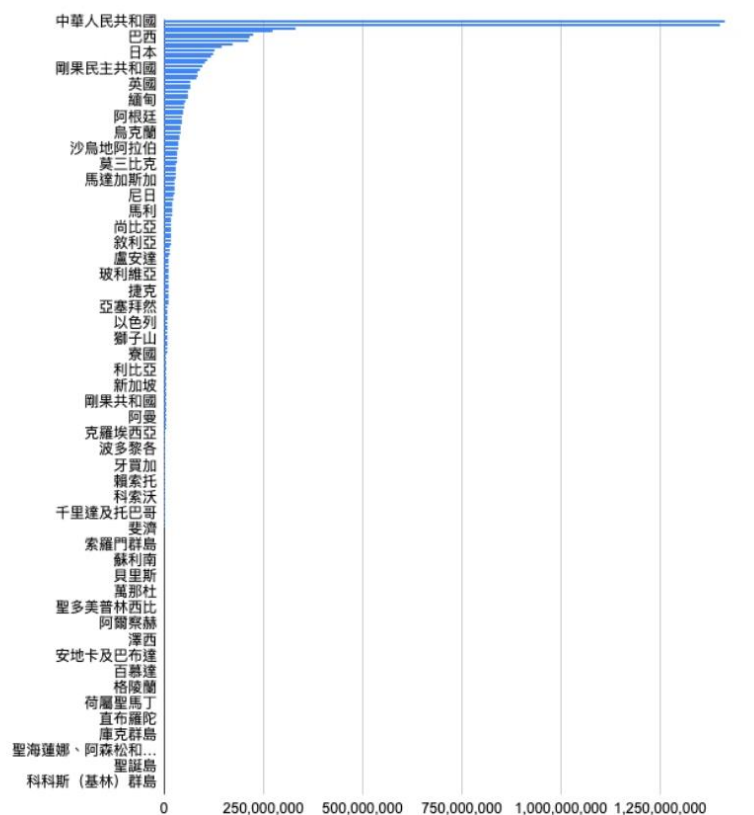
長條圖

- 顯示資料間的差異，或類別間數值的大小



長條圖

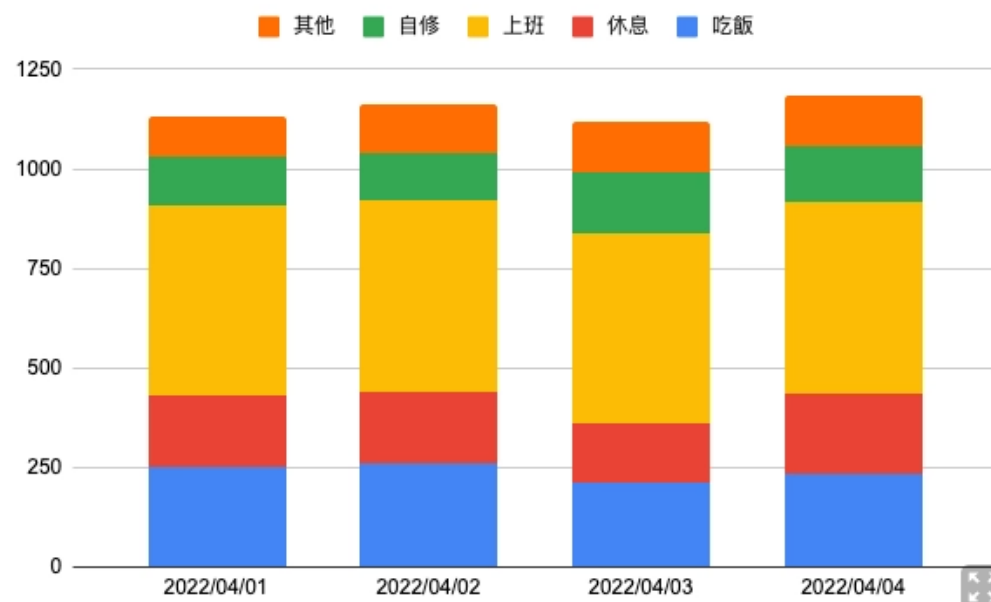
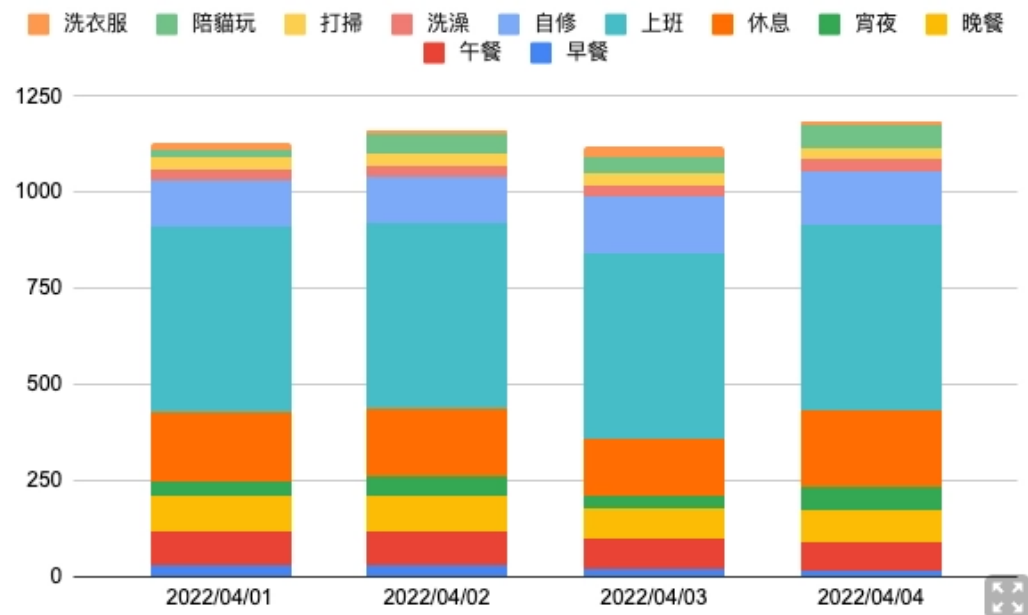
- 類別太多



資料來源: [我該用哪個圖表？\(二\) 柱狀圖、長條圖 | 方格子 vocus](#)

長條圖

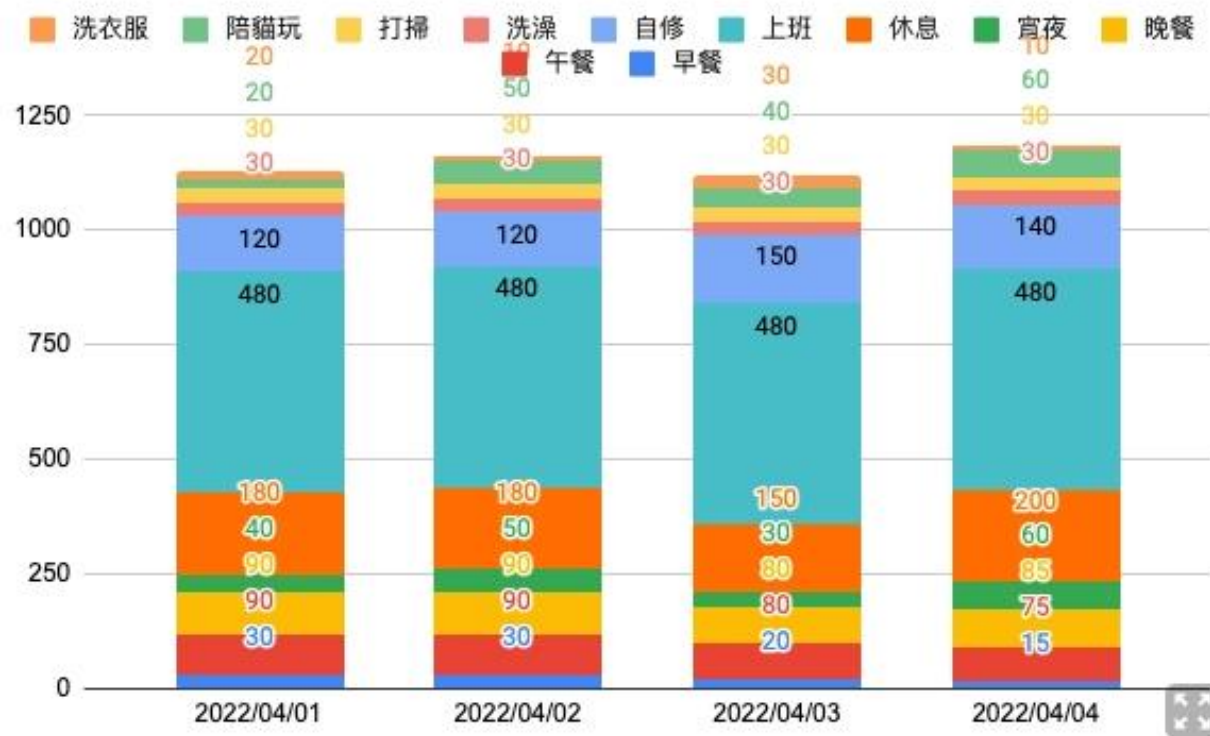
- 太細的資料



資料來源: [我該用哪個圖表？\(二\) 柱狀圖、長條圖 | 方格子 vocus](#)

長條圖

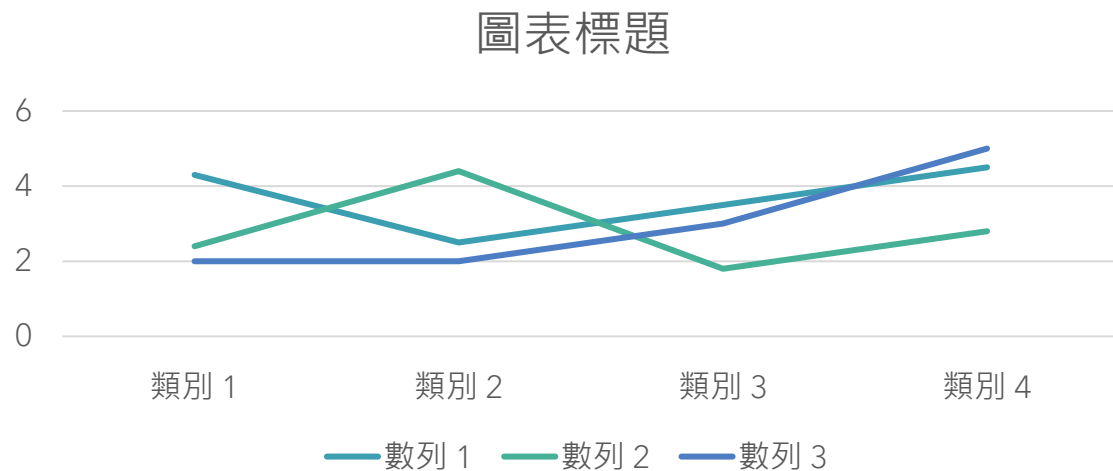
- 資料標籤



資料來源: [我該用哪個圖表？ \(二\) 柱狀圖、長條圖 | 方格子 vocus](#)

折線圖

- 適合呈現資料在一定時間內的變化或趨勢
- 通常橫軸（X軸）是時間、縱軸（Y軸）是資訊



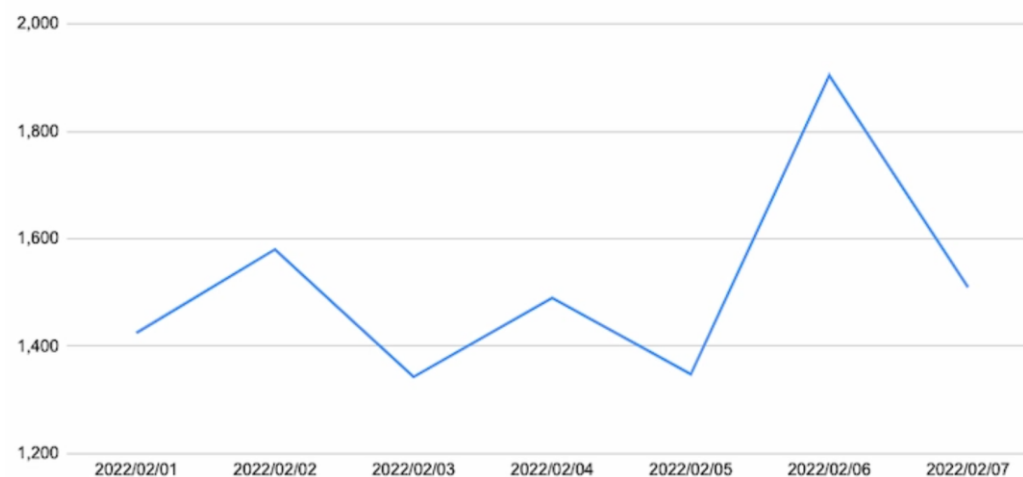
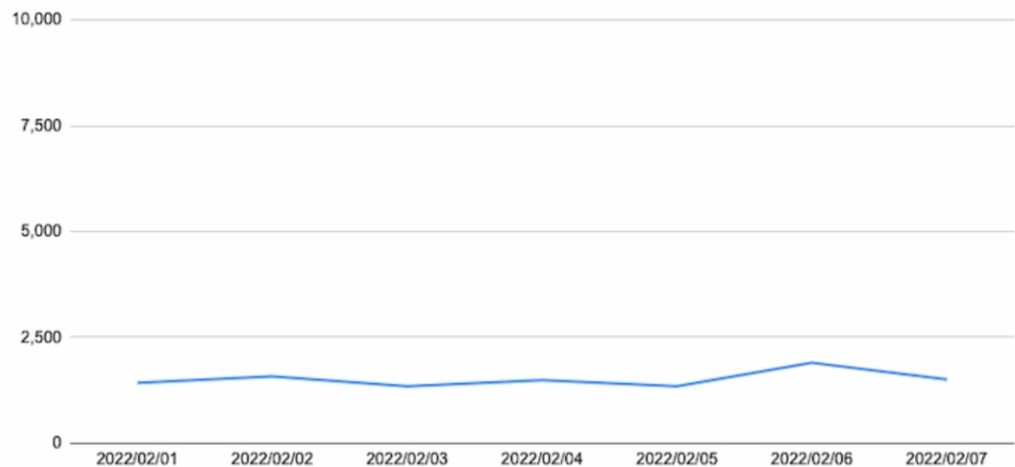
折線圖

- 折線圖能讓人聯想到資料趨勢 (通常往右為新資料)、連續性等，且能快速辨別最高點、最低點、往上或下的趨勢等等。
- 通常 x 軸是時間， y 軸為資訊。



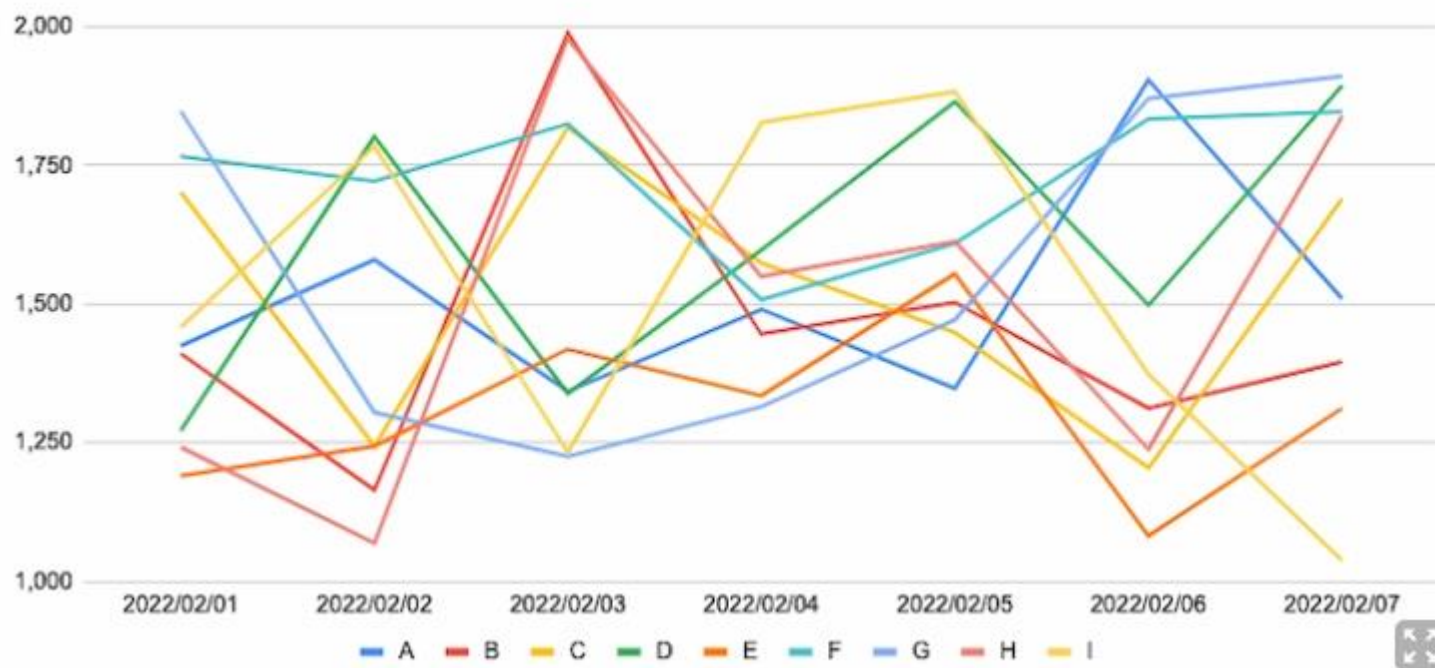
折線圖

- 軸的刻度要合理、要均勻分布



折線圖

- 折線要明顯，勿失去意義



Matplotlib

- Python的視覺化套件
- 提供多種圖表的函式庫
 - 曲線圖、直方圖、散點圖、折線圖等。
- 套件 `import` 方法

```
import matplotlib.pyplot as plt
```

資料集

- [歷年國內主要觀光遊憩據點遊客人數月別統計 | 政府資料開放平臺 \(data.gov.tw\)](https://data.gov.tw)

歷年國內主要觀光遊憩據點遊客人數月別統計

資料內容提供歷年國內主要遊憩據點遊客人數月別統計

評分此資料集：

☆☆☆☆☆

平均 3.25 (16 人次投票)

瀏覽次數：33100 下載次數：9423 意見數：7

DCAT 詞彙

列印

主要欄位說明 *粗體欄位為資料標準欄位	年別、類型、觀光遊憩區、細分、縣市、 縣市別 、1月、2月、3月、4月、5月、6月、7月、8月、9月、10月、11月、12月、合計
資料資源下載網址	<div><div> CSV</div><div>檢視資料</div><div>歷年國內主要觀光遊憩據點遊客人數月別統計</div></div>
提供機關	交通部觀光署

資料集

- 年別、類型、觀光遊憩區、細分、縣市別、1月、2月、3月、4月、5月、6月、7月、8月、9月、10月、11月、12月、合計

年別	類型	觀光遊憩區	細分	縣市別	1月	2月	3月	4月	5月
2012	國家公園	陽明山國家公園	陽明山遊憩區	臺北市	12187	18612	18580	13329	14179
2012	國家公園	陽明山國家公園	陽明書屋	臺北市	2597	3617	5221	4317	3497
2012	國家公園	陽明山國家公園	陽明公園	臺北市	161000	463000	705200	85600	38600
2012	國家公園	陽明山國家公園	大屯遊憩區	臺北市	17047	29079	38558	34332	43906
2012	國家公園	陽明山國家公園	龍鳳谷遊憩區	臺北市	21201	22134	62447	64691	16243
2012	國家公園	陽明山國家公園	小油坑遊憩區	臺北市	20839	16853	24528	27840	18760
2012	國家公園	玉山國家公園	塔塔加遊憩區	南投縣	83806	59526	63451	71541	64384
2012	國家公園	玉山國家公園	梅山遊客中心	高雄市	4846	1854	1190	1849	1092
2012	國家公園	玉山國家公園	南安遊客中心	花蓮縣	37578	19373	21987	20037	18384
2012	國家公園	玉山國家公園	玉山管理處	南投縣	13262	9460	9322	10362	10233
2012	國家公園	玉山國家公園	排雲山莊	南投縣	371	80	376	1476	1511

圖表套件中文化

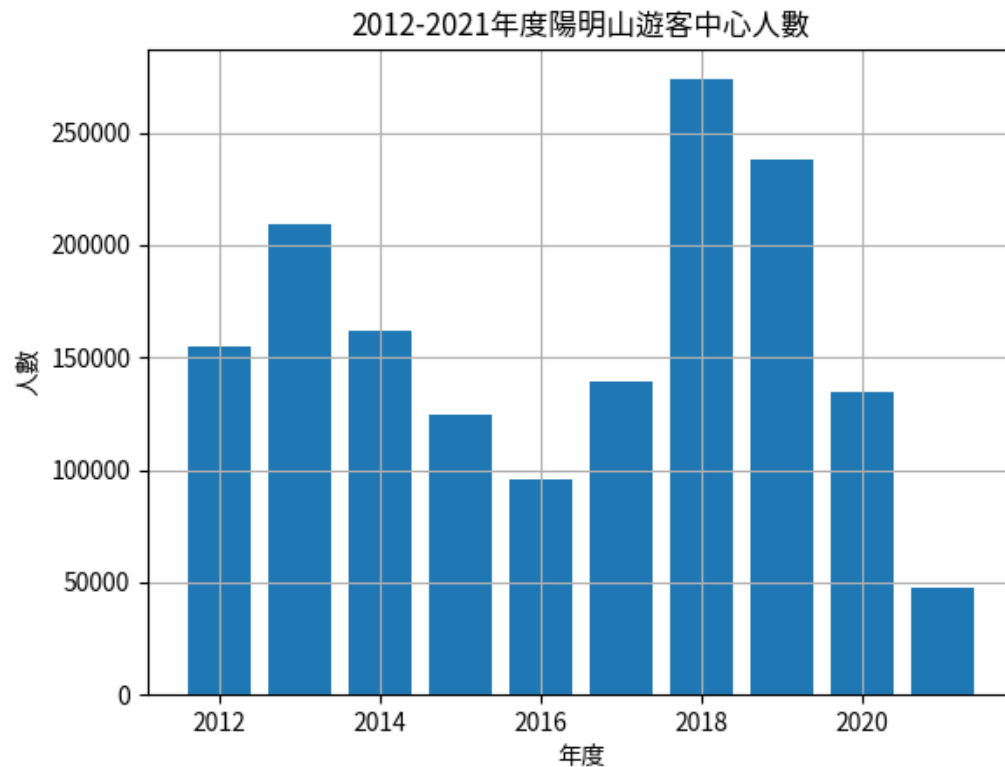
- 將「TaipeiSansTCBeta-Regular.ttf」這個字形放入專案資料夾中
- import 以下套件並設定字形

```
import matplotlib as mpl
import matplotlib.pyplot as plt
from matplotlib.font_manager import fontManager

fontManager.addfont('TaipeiSansTCBeta-Regular.ttf')
mpl.rc('font', family='Taipei Sans TC Beta')
```

長條圖繪製 - 陽明山遊客中心

- 歷年陽明山遊客中心的遊客人數
 - 資料年度:2012-2023
 - 資料欄位:合計



長條圖繪製 - 陽明山遊客中心

- 歷年陽明山遊客中心的遊客人數
 - 先篩選資料
 - 設定x軸、y軸資料
 - 繪製長條圖
 - 設定標題、標籤與格線
 - 儲存圖表

長條圖繪製 - 陽明山遊客中心

- 讀入資料

```
data = pd.read_csv("觀光.csv", encoding='UTF-8')

# 篩選資料
centers = data[data['細分'] == "陽明山遊客中心"]
```

	年別	類型	觀光遊憩區	細分	...	10月	11月	12月	合計
0	2012	國家公園	陽明山國家公園	陽明山遊客中心	...	11515.0	6896.0	9631.0	154586
287	2013	國家公園	陽明山國家公園	陽明山遊客中心	...	8156.0	7192.0	6792.0	209378
574	2014	國家公園	陽明山國家公園	陽明山遊客中心	...	8593.0	8530.0	6505.0	162185
869	2015	國家公園	陽明山國家公園	陽明山遊客中心	...	10575.0	11370.0	8977.0	124241
1164	2016	國家公園	陽明山國家公園	陽明山遊客中心	...	0.0	0.0	0.0	95994
1469	2017	國家公園	陽明山國家公園	陽明山遊客中心	...	0.0	0.0	0.0	139602
1774	2018	國家公園	陽明山國家公園	陽明山遊客中心	...	16738.0	18460.0	20950.0	273547
2100	2019	國家公園	陽明山國家公園	陽明山遊客中心	...	17686.0	16880.0	15873.0	238097
2426	2020	國家公園	陽明山國家公園	陽明山遊客中心	...	9071.0	8622.0	5340.0	134879
2742	2021	國家公園	陽明山國家公園	陽明山遊客中心	...	1127.0	NaN	NaN	47593

- 設定資料與繪製長條圖

```
# 設定x軸和y軸的數據
x = centers['年別']
y = centers['合計']

# 繪製長條圖
plt.bar(x, y)
```

長條圖繪製 - 陽明山遊客中心

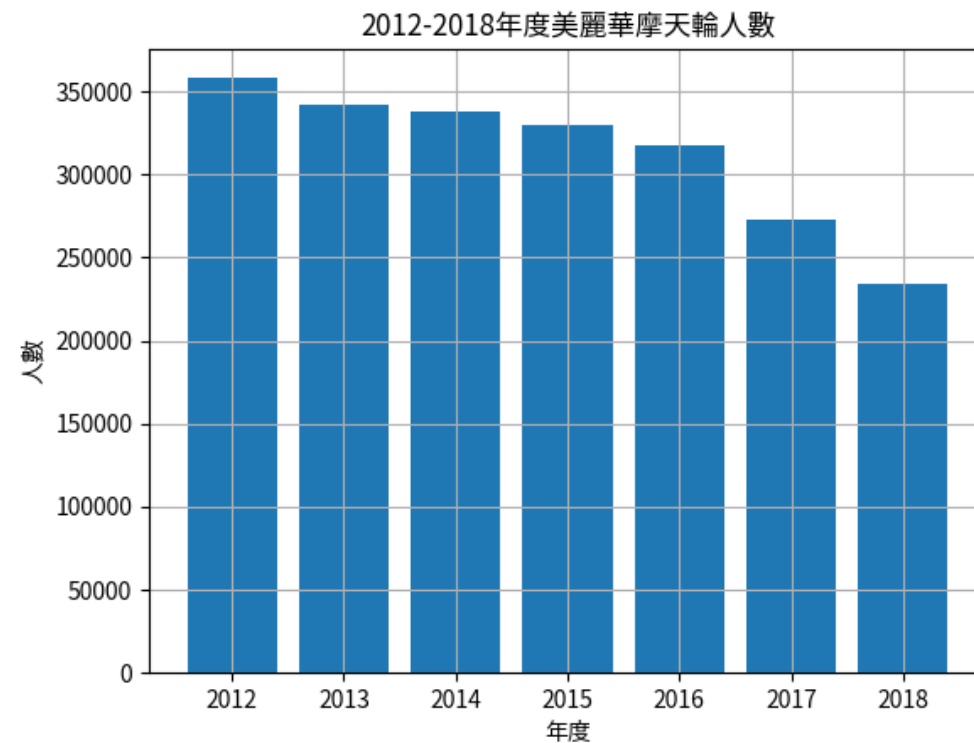
- 設定圖表屬性與儲存

```
# 設定標題、標籤與格線
plt.title('2012-2021年度陽明山遊客中心人數')
plt.xlabel('年度')
plt.ylabel('人數')
plt.grid(True)

# 儲存圖表
plt.savefig('center_bar_chart.png')
```

Practice: 2018-2023 美麗華摩天輪遊客人數

- 計算2018-2023美麗華摩天輪人數



Practice: 2018-2023 美麗華摩天輪遊客人數

```
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
from matplotlib.font_manager import fontManager

fontManager.addfont('TaipeiSansTCBeta-Regular.ttf')
mpl.rc('font', family='Taipei Sans TC Beta')

data = pd.read_csv("觀光.csv", encoding='UTF-8')

# 篩選資料
m = data[data['細分'] == "美麗華摩天輪"]
print(m)

# 設定x軸和y軸的數據
x_data = []
y_data = []
for i, r in m.iterrows():
    if r['年別'] >= 2018:
        x_data.append(r['年別'])
        y_data.append(r['合計'])

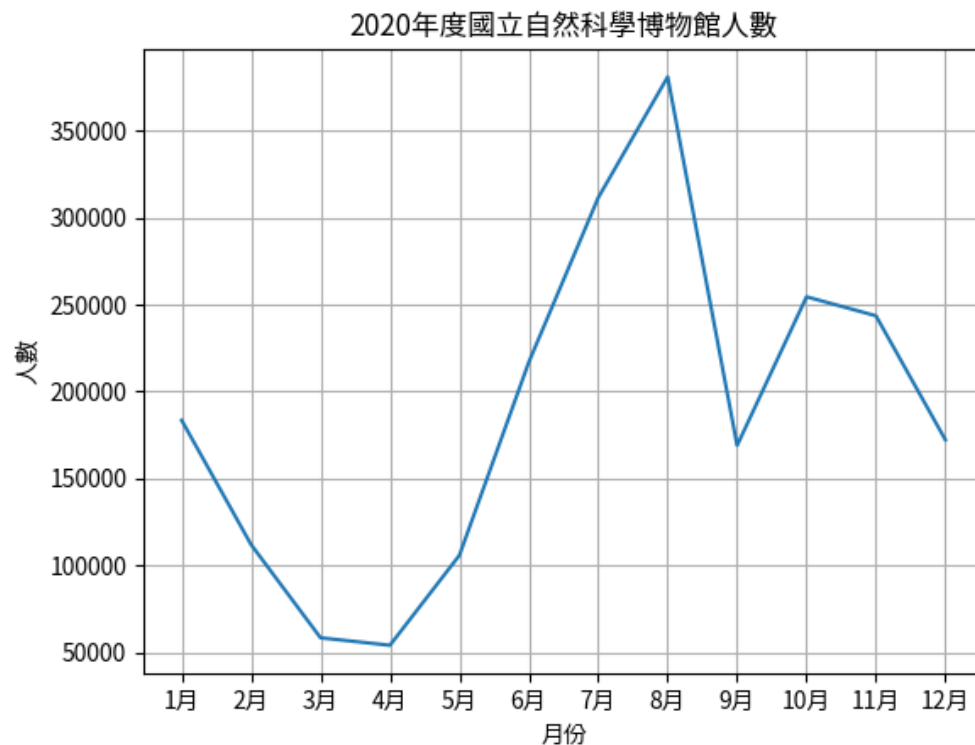
# 繪製長條圖
plt.bar(x_data, y_data)

# 設定標題、標籤與格線
plt.title('2012-2018年度美麗華摩天輪人數')
plt.xlabel('年度')
plt.ylabel('人數')
plt.grid(True)

# 儲存圖表
plt.savefig('m.png')
```

折線圖繪製 - 2020國立自然科學博物館全年人數

- 2020國立自然科學博物館全年人數資料
- 資料欄位:年別、1-12月



長條圖繪製 - 陽明山遊客中心

- 讀入資料

```
data = pd.read_csv("觀光.csv", encoding='UTF-8')  
  
# 篩選資料  
museum = data[(data['細分'] == "國立自然科學博物館") & (data['年別'] == 2020)]
```

- 設定資料與繪製長條圖

```
# 設定x軸和y軸的數據  
x = museum.columns[5:17]  
y = museum.iloc[0, 5:17]  
  
# 繪製長條圖  
plt.plot(x, y)
```

長條圖繪製 - 陽明山遊客中心

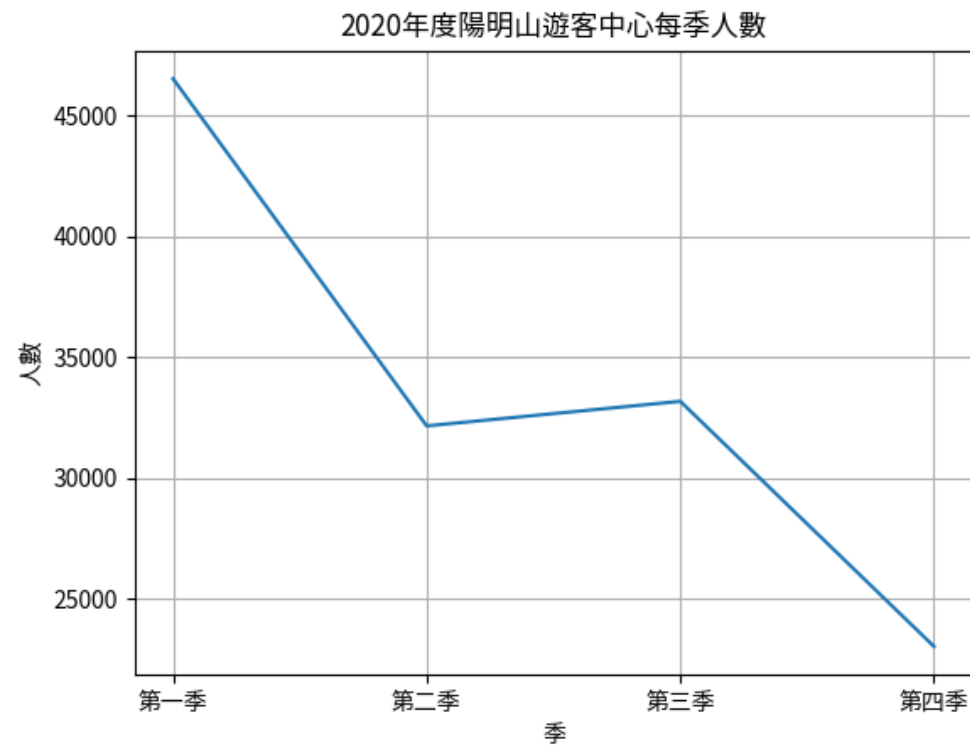
- 設定圖表屬性與儲存

```
# 設定標題、標籤與格線
plt.title('2020年度國立自然科學博物館人數')
plt.xlabel('月份')
plt.ylabel('人數')
plt.grid(True)

# 儲存圖表
plt.savefig('2020_meseum.png')
```

Practice: 2023陽明山遊客中心每季遊客折線圖

- 計算 2023 陽明山遊客中心每一季(3個月)的遊客折線圖



Practice: 2023陽明山遊客中心每季遊客折線圖

```
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
from matplotlib.font_manager import fontManager

fontManager.addfont('TaipeiSansTCBeta-Regular.ttf')
mpl.rc('font', family='Taipei Sans TC Beta')

data = pd.read_csv("觀光.csv", encoding='UTF-8')

# 篩選資料
center = data[(data['細分'] == "陽明山遊客中心") & (data['年別'] == 2023)]

# 設定x軸和y軸的數據
x = ['第一季', '第二季', '第三季', '第四季', ]
y = [center.iloc[0, 5:8].sum(), center.iloc[0, 8:11].sum(), center.iloc[0, 11:14].sum(), center.iloc[0, 14:17].sum()]
print(y)

# 繪製長條圖
plt.plot(x, y)

# 設定標題、標籤與格線
plt.title('2023年度陽明山遊客中心每季人數')
plt.xlabel('季')
plt.ylabel('人數')
plt.grid(True)

# 儲存圖表
plt.savefig('2023_陽明山遊客中心.png')
```

Numpy

- NumPy是Python在進行科學運算時，一個非常基礎的Package。
 - 提供非常高效能的多維陣列(multi-dimensional array)數學函式庫
 - 方便有用的線性代數(Linear Algebra)及傅立葉轉換(Fourier Transform)能力
 - 利用NumPy Array替代Python List



Numpy - 建立陣列及初始值

- 一維陣列

```
import numpy as np  
  
a_list = [1, 2, 3, 4]  
np_a_list = np.array(a_list) [1 2 3 4]
```

- 二維陣列

```
import numpy as np  
  
a_list = [1, 2, 3, 4]  
b_list = [1.5, 2.5, 3.5, 4.5]  
np_ab_list = np.array([a_list, b_list])  
[[1.  2.  3.  4. ]  
 [1.5 2.5 3.5 4.5]]
```

Numpy - 建立陣列及初始值

- 直接用numpy的方法建立陣列

```
np.zeros((2, 3))      # 建立一個2x3全為0的陣列
np.ones((2, 3, 4))    # 建立一個2x3x4全為1的陣列
np.arange(1, 10, 2)   # 建立一個由1開始，不超過10，間隔值為2的均勻數值陣列
np.linspace(0, 10, 5) # 建立一個0到10之間，均勻的5個數值陣列
np.full((3,2), 8)     # 建立一個3x2全為8的陣列
np.eye(5)             # 建立一個5x5的單位矩陣
np.random.random((2,3)) # 建立一個2x3的隨機值矩陣
```

Numpy - arange

- `range(a, b, x)` : 建立一個從 `a` 開始到 `b-1` 結束，間隔為 `x` 的列表。

- **`x` 必須為整數**

```
range(1, 10, 3) [1, 4, 7]
```

```
>>> range(1, 10, 0.5)
Traceback (most recent call last):
  File "<input>", line 1, in <module>
TypeError: 'float' object cannot be interpreted as an integer
```

- `np.arange(a, b, x)` : 建立一個從 `a` 開始到 `b-1` 結束

```
>>> np.arange(1, 10, 0.5)
array([1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5, 5. , 5.5, 6. , 6.5, 7. ,
       7.5, 8. , 8.5, 9. , 9.5])
```

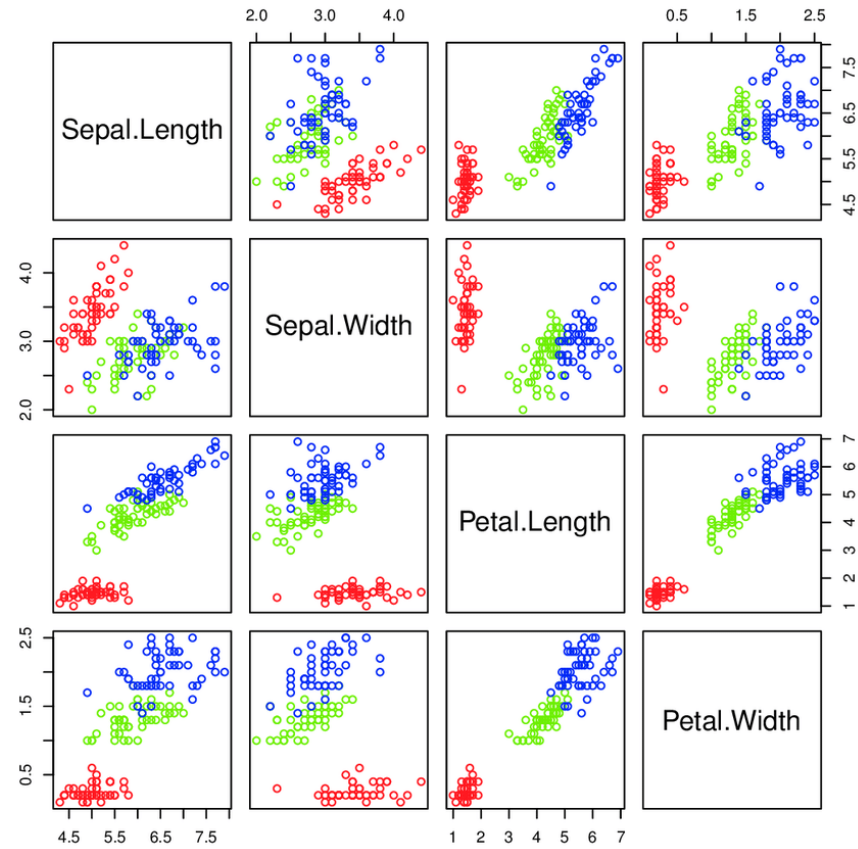
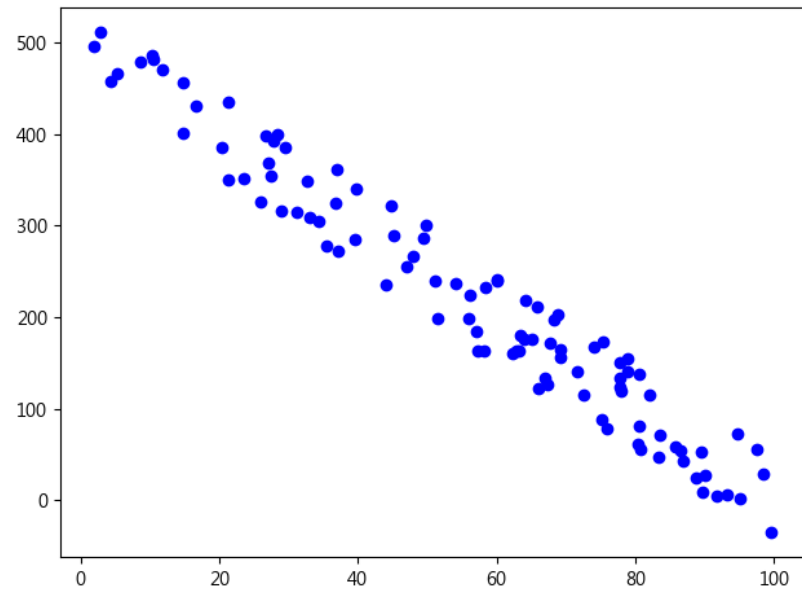

Numpy - 陣列操作

- `.sum()` : 回傳 `array` 中元素的總和
- `.mean()` : 回傳 `array` 中元素的平均
- `.sort()` : 排列 `array` 中的元素
- `.reshape()` : 在不改變元素數值的情況下修改排列方式

```
>>> a = np.array([7, 2, 8, 4, 9, 6])
>>> a
array([7, 2, 8, 4, 9, 6])
>>> a.sum()
36
>>> a.mean()
6.0
```

```
>>> a.sort()
>>> a
array([2, 4, 6, 7, 8, 9])
>>> a.reshape(3,2)
array([[2, 4],
       [6, 7],
       [8, 9]])
```

散佈圖



散佈圖

- 散佈圖具有 X 軸和 Y 軸。
 - X 是帶有獨立變數的水平線，
 - Y 是帶有相依變數的垂直線。
 - 在代表兩個坐標的交點處製作一個標記或點。
- 相關性
 - 線性或非線性：可以通過資料點形成線性（直接）相關，如果是非線性相關則可能顯示曲線關係。
 - 弱或強：相關性越強，點和點之間的距離就越近。相關性弱的話則是有較多散開的資料點。

使用 **numpy** 產生數據並標示於散佈圖

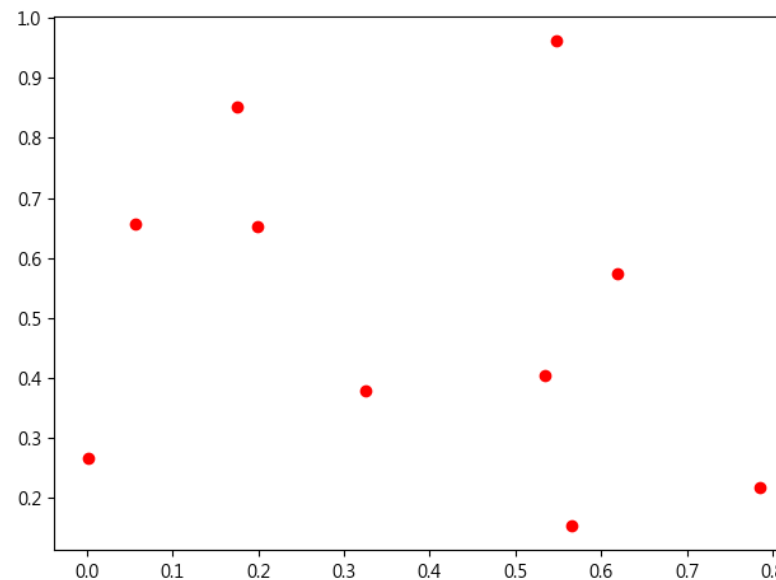
- 利用 `np.random.random()` 的方法隨機產生10個0-1的數值。
- 使用 `.scatter()` 的方法繪製散佈圖。

```
import numpy as np
import matplotlib.pyplot as plt

x_data = np.random.random(10)
y_data = np.random.random(10)

plt.scatter(x_data, y_data, color='red')

plt.show()
```



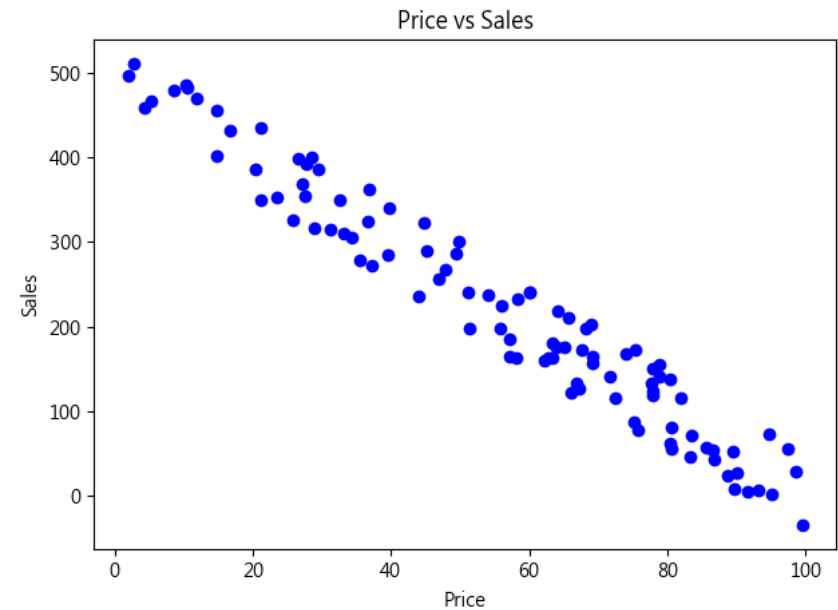
使用 **pandas** 讀入資料並繪製圖表

- 將 sales_data.csv 讀入為 DataFrame。
- 把 Price 與 Sales 畫成散佈圖

```
import matplotlib.pyplot as plt
import pandas as pd

df = pd.read_csv("sales_data.csv")

plt.scatter(df['Price'], df['Sales'], color='blue')
plt.xlabel("Price")
plt.ylabel("Sales")
plt.title("Price vs Sales")
plt.show()
```



使用 **pandas** 讀入資料並繪製圖表

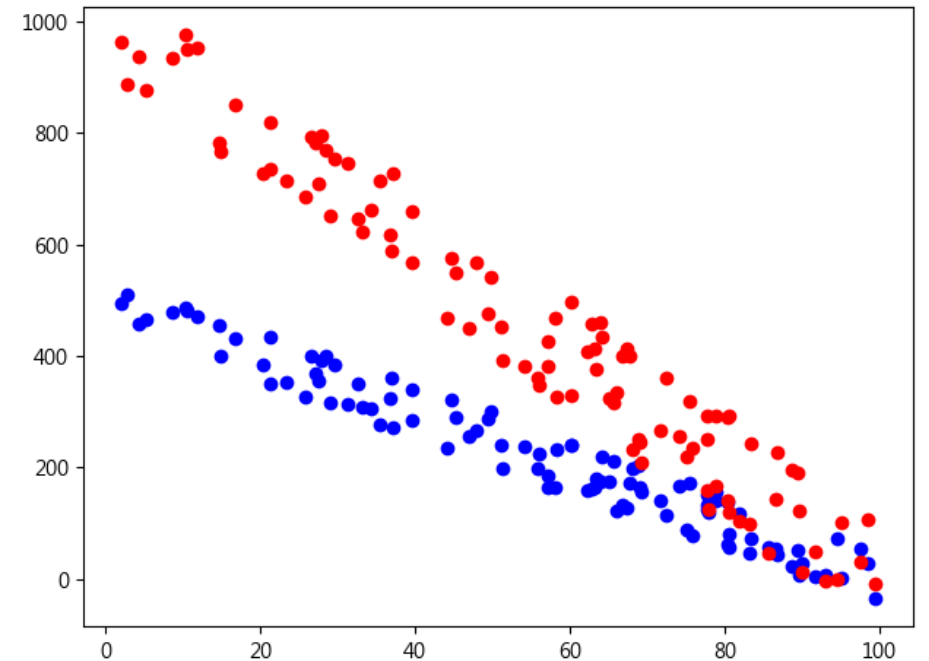
- 將資料讀入為 DataFrame。
- 把 Price 與 Sales 畫成藍色點的散佈圖。
- 把 Price 與 Inventory 畫成紅色點的散佈圖。

```
import matplotlib.pyplot as plt
import pandas as pd

# 假設我們有以下的點數據
df = pd.read_csv("sales_data.csv")

plt.scatter(df['Price'], df['Sales'], color='blue')
plt.scatter(df['Price'], df['Inventory'], color='red')

plt.show()
```



使用 **pandas** 讀入資料並繪製圖表

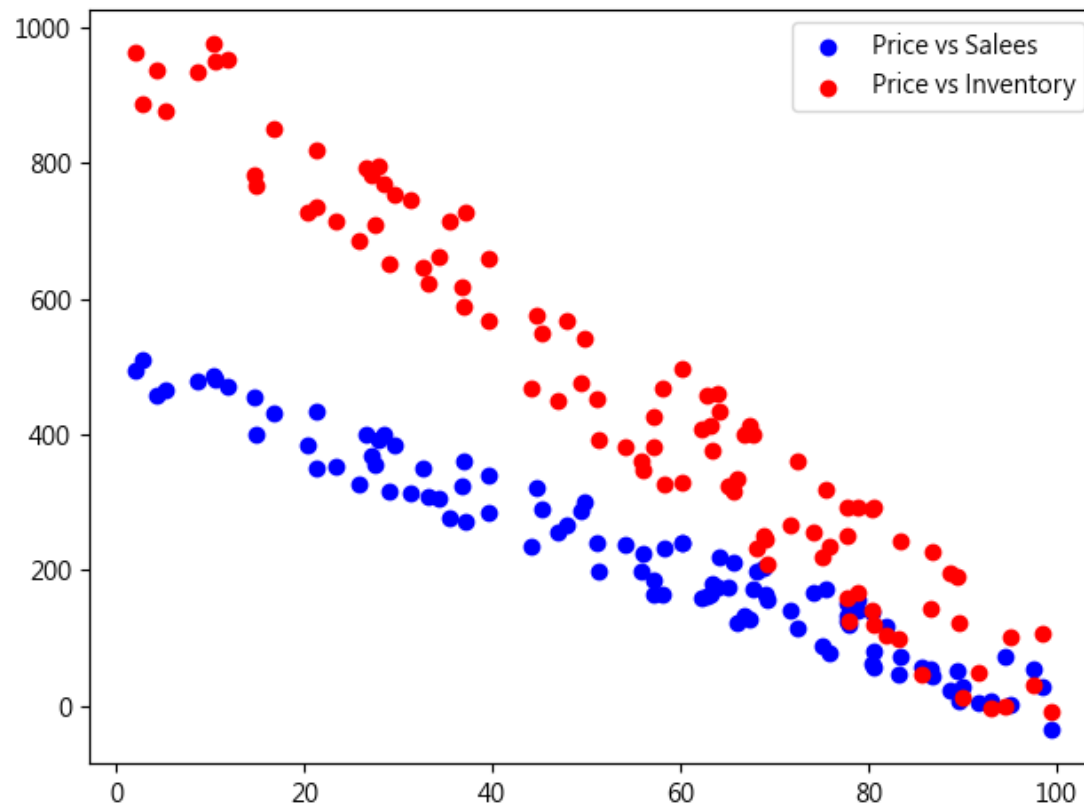
- 加入標籤

```
import matplotlib.pyplot as plt
import pandas as pd

df = pd.read_csv("sales_data.csv")

plt.scatter(df['Price'],
            df['Sales'],
            color='blue',
            label='Price vs Salees')
plt.scatter(df['Price'],
            df['Inventory'],
            color='red',
            label='Price vs Inventory')

plt.legend()
plt.show()
```

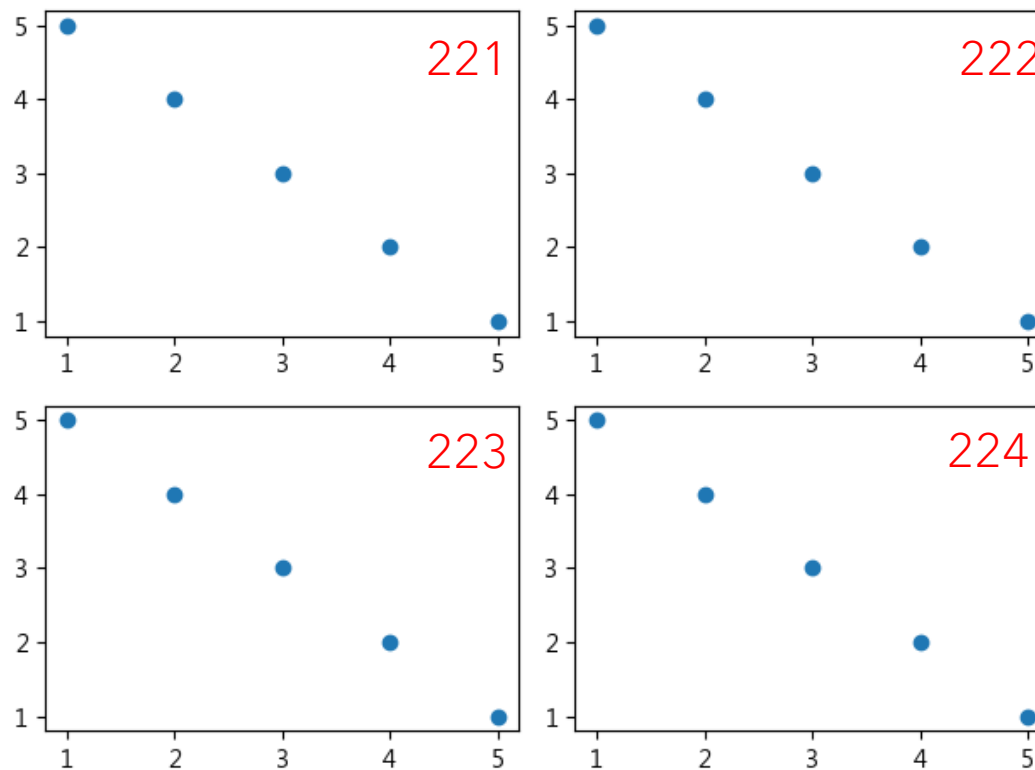


subplot - 建立多個子圖表

- plt 建立一個 figure 後，就能透過 subplot 的方法來繪製多個子圖表

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [5, 4, 3, 2, 1]
fig = plt.figure()
plt.subplot(221)
plt.scatter(x, y)
plt.subplot(222)
plt.scatter(x, y)
plt.subplot(223)
plt.scatter(x, y)
plt.subplot(224)
plt.scatter(x, y)
plt.show()
```



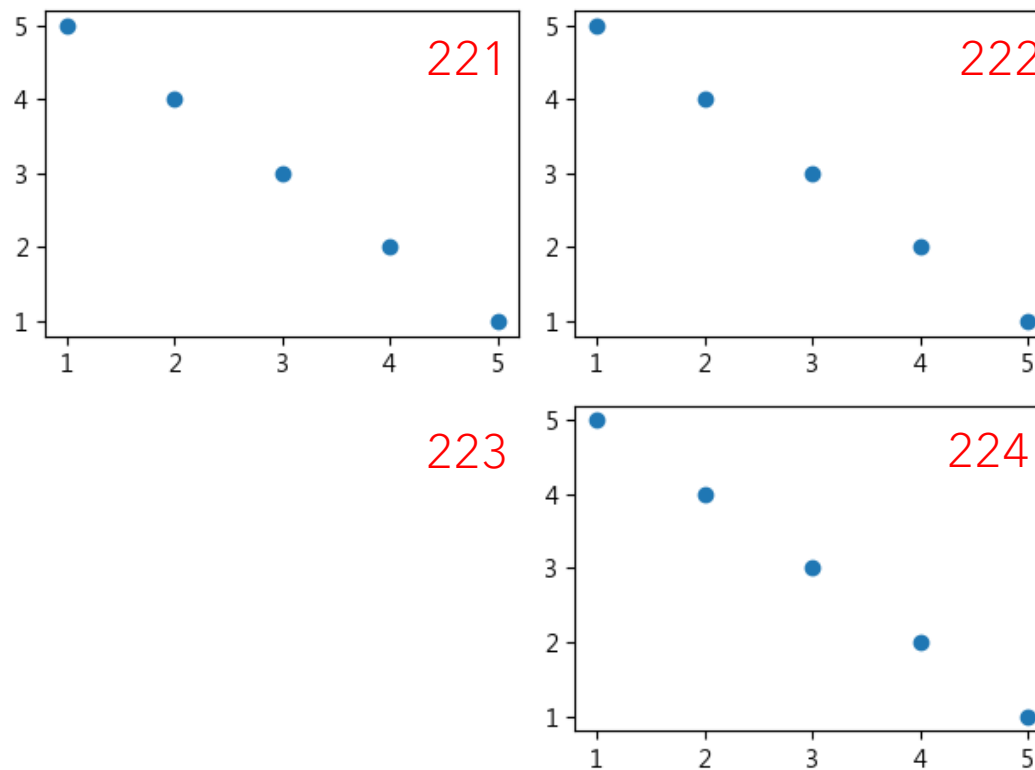
subplot - 建立多個子圖表

- plt 建立一個 figure 後，就能透過 subplot 的方法來繪製多個子圖表

```
import matplotlib.pyplot as plt

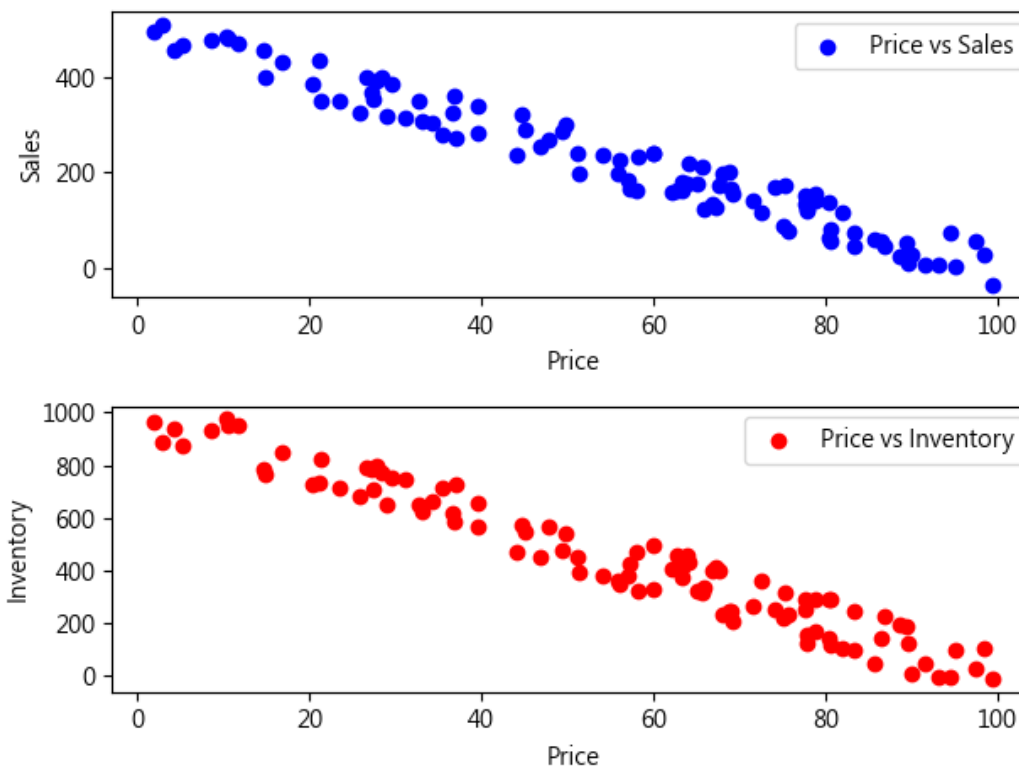
x = [1, 2, 3, 4, 5]
y = [5, 4, 3, 2, 1]
fig = plt.figure()
plt.subplot(221)
plt.scatter(x, y)
plt.subplot(222)
plt.scatter(x, y)

plt.subplot(224)
plt.scatter(x, y)
plt.show()
```



Practice - subplot繪製子圖表

- 將 `sales_data.csv` 的
以下欄位分別畫出散佈圖
 - Price vs Sales
 - Price vs Inventory

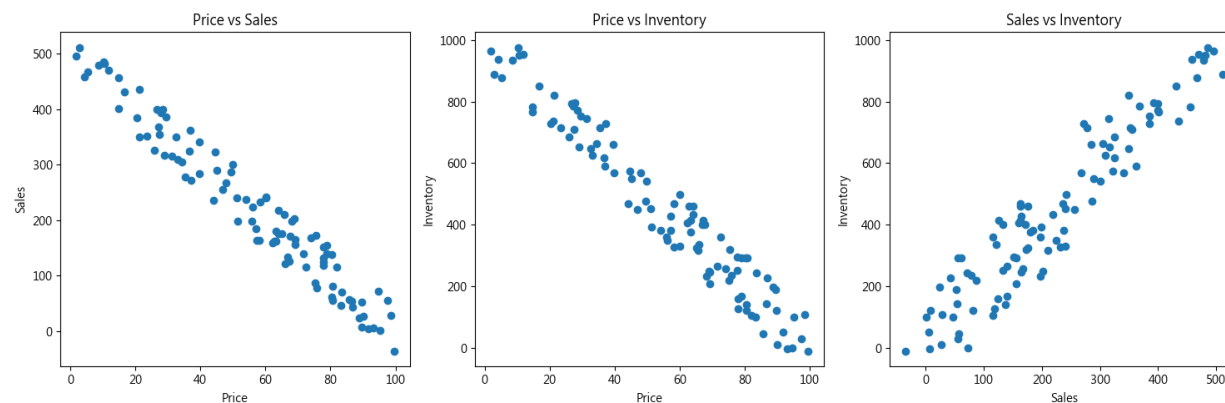


figsize

- 繪製子圖表時，可以設定 `figure` 方法裡的 `figsize` 來設定整體圖形的大小。

```
plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1) # 1行3列的第1個位置
plt.scatter(data['Price'], data['Sales'])
plt.title('Price vs Sales')
plt.xlabel('Price')
plt.ylabel('Sales')
```



HW: 圖表繪製

- [歷年國內主要觀光遊憩據點遊客人數月別統計.csv](#)
- 2018-2023國立自然科學博物館歷月遊客人數折線圖
- 2023各類型旅遊人數長條圖
- [weather data.csv](#) 裡有4個欄位的資料，請分別畫出這四個欄位之間的散佈圖觀察趨勢。
 - Temperature: 溫度
 - Ice_Cream_Sales: 冰淇淋銷售量
 - Swimming_Pool_Usage: 游泳池使用人數
 - Cold_Drink_Sales: 冷飲銷售數量

