

D19 使用 Matplotlib 繪製各種常用圖表

[簡報閱讀](#)[範例與作業](#)[問題討論](#)[學習心得\(完成\)](#)[重要知識點](#)[前期回顧](#)[製作繪圖板 Subplots](#)[製作繪圖板 Axes](#)[製作 3D 繪圖板](#)[知識點回顧](#)[延伸閱讀](#)

重要知識點



- 完成今日課程後你應該可以了解
 - 各種常見圖表適用情境範例展示
 - 經典圖表(直方、分箱、...)
 - 多重子圖表(subplot)
 - 其他圖型：密度圖、等高線圖

前期回顧

```
3 import numpy as np
4
5 # 準備數據 ... 假設我要畫一個sin波 從0~180度
6
7 x = np.arange(0,180)
8 y = np.sin(x * np.pi / 180.0)
9
10 # 開始畫圖
11
12 # 設定要畫的的x,y數據list....
13
14 plt.plot(x,y)
15
16 # 設定圖的範圍，不設的話，系統會自行決定
17 plt.xlim(-30,390)
18 plt.ylim(-1.5,1.5)
19
20 # 照需要寫入x 軸和y軸的 label 以及title
21
22 plt.xlabel("x-axis")
23 plt.ylabel("y-axis")
24 plt.title("The Title")
25
26 # 在這個指令之前，都還在做畫圖的動作
27 # 這個指令算是 "秀圖"
28 plt.show()
```

製作繪圖板 Subplots

matplotlib 的圖像都位於 Figure 物件，所以我們可以設定一個畫板，並在這畫板中配置子版

`plt.subplot(a, b, c)`

a：代表 X 軸的分割

b：代表 y 軸的分割

`subplot(2,1,1)`

`subplot(2,1,2)`

`subplot(1,2,1)`

`subplot(1,2,2)`

`subplot(2,2,1)`

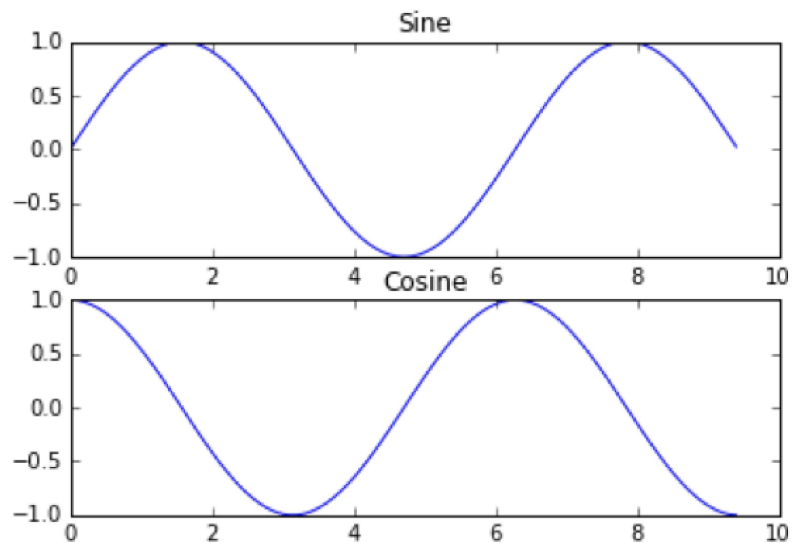
`subplot(2,2,2)`

`subplot(2,2,3)`

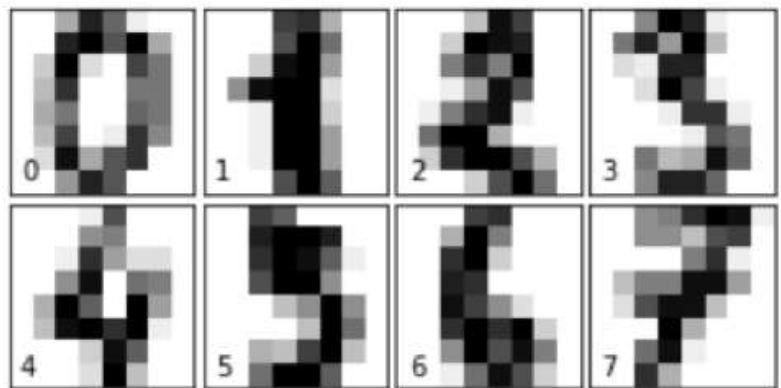
`subplot(2,2,4)`

We can plot different things in the same figure using the **subplot** function.

```
2 import numpy as np
3 # 載入 matplotlib
4 import matplotlib.pyplot as plt
5
6 x = np.arange(0, 3 * np.pi, 0.1)
7 y_sin = np.sin(x)
8 y_cos = np.cos(x)
9 # 設定雙格畫板大小
10 plt.subplot(2, 1, 1)
11 plt.plot(x, y_sin)
12 plt.title("Sine")
13
14 # 設定雙格畫板大小
15 plt.subplot(2, 1, 2)
16 plt.plot(x, y_cos)
17 plt.title("Cosine")
18
19 plt.show()
```



```
3 # 載入 matplotlib
4 import matplotlib.pyplot as plt
5
6 # 載入 `digits`
7 digits = datasets.load_digits()
8
9 # 設定圖形的大小 (寬, 高)
10 fig = plt.figure(figsize=(4, 2))
11
12 # 調整子圖形
13 fig.subplots_adjust(left=0, right=1, bottom=0,
14
15 # 把前 8 個手寫數字顯示在子圖形
16 for i in range(8):
17     # 在 2 x 4 網格中第 i + 1 個位置繪製子圖形, 並
18     ax = fig.add_subplot(2, 4, i + 1, xticks =
19     # 顯示圖形, 色彩選擇灰階
20     ax.imshow(digits.images[i], cmap = plt.cm.b
21     # 在左下角標示目標值
22     ax.text(0, 7, str(digits.target[i]))
23
24 # 顯示圖形
25 plt.show()
```



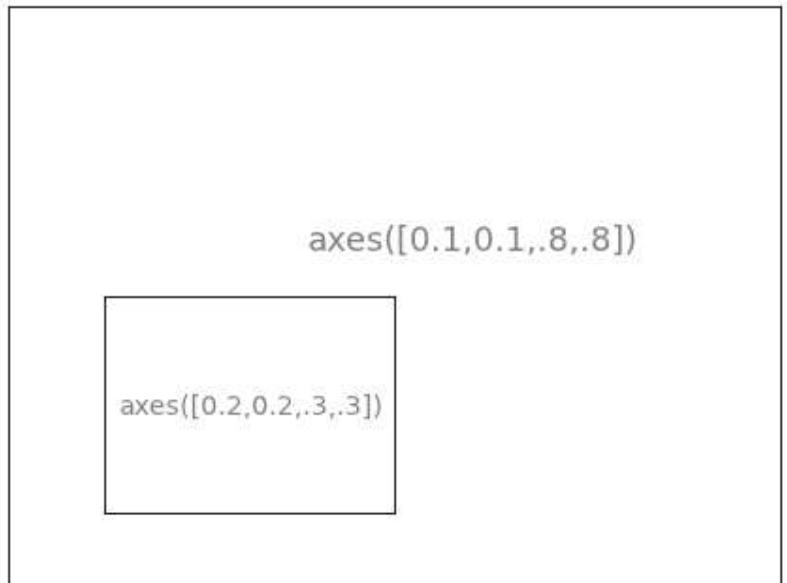
製作繪圖板 Axes

- 但是可以將圖放置在圖中的任何位置。因此，如果要在較大的圖中放置較小的圖，則可以使用軸。

```

1  #import matplotlib.pyplot as plt
2
3  #決定最外框
4  plt.axes([0.1,0.1,.8,.8])
5  plt.xticks([], plt.yticks([]))
6  plt.text(0.6,0.6, 'axes([0.1,0.1,.8,.8])',ha='c')
7
8  #決定內框
9  plt.axes([0.2,0.2,.3,.3])
10 plt.xticks([], plt.yticks([]))
11 plt.text(0.5,0.5, 'axes([0.2,0.2,.3,.3])',ha='c')
12
13 plt.show()

```

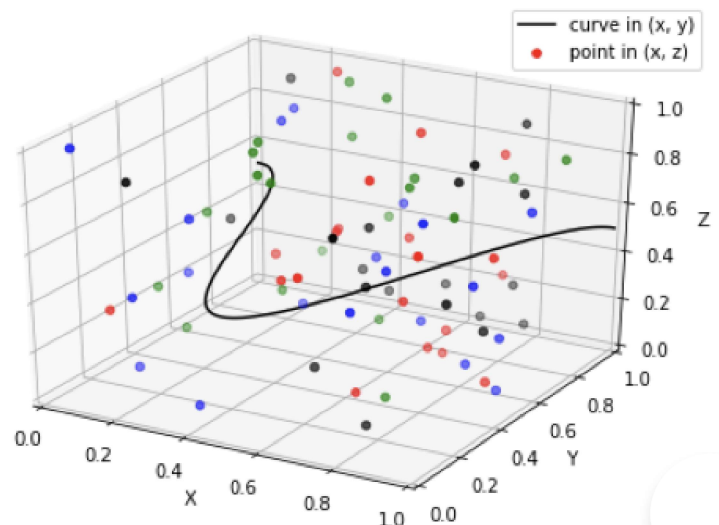


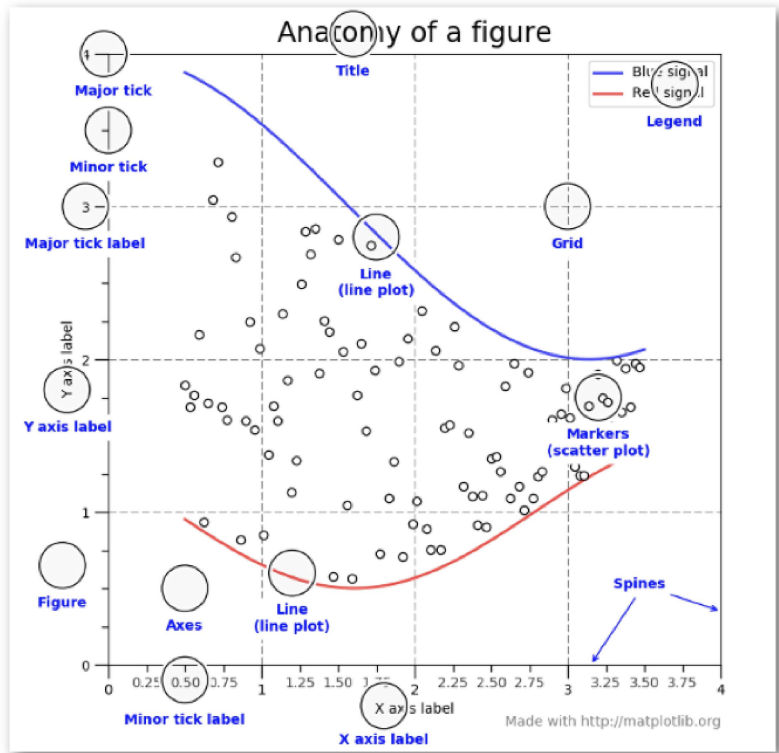
製作 3D 繪圖板

- 主要把想要觀察的重點與場景實現兩種交互
 - 一種是可以操縱場景從而能夠從不同的角度觀察模型
 - 一種是擁有添加與操作修改模型物件的能力

```

1  #導入必要的模組
2  import numpy as np
3  import matplotlib.pyplot as plt
4  from mpl_toolkits.mplot3d import Axes3D
5
6  # 創建一個3d坐標系
7  fig = plt.figure()
8  ax = Axes3D(fig)
9  #直接查詢參數與設定
10 #help(plt.plot)
11 #help(np.random.sample)
12
13 # 利用x軸和y軸繪製sin曲線
14 x = np.linspace(0, 1, 100) # linspace創建等差陣列
15 y = np.cos(x * 2 * np.pi) / 2 + 0.5
16 # 通過zdir = 'z' 將資料繪製在z軸，zs = 0.5 則是將資料繪製在z = 0.5的平面上
17 ax.plot(x, y, zs = 0.5, zdir = 'z', color = 'black')
    
```





建議以下步驟學習如何使用 Matplotlib：

1. 學習 Matplotlib 的基本術語,具體來說就是什麼是 Figure 和 Axes。
2. 一直使用面向對象的介面，養成習慣。
3. 用基礎的 pandas 繪圖開始可視化。
4. 使用 seaborn 進行稍微複雜的數據可視化。
5. 使用 Matplotlib 自訂 pandas 或 seaborn 視覺化。

延伸閱讀

Python資料分析 (五) Matplotlib做視覺化

網站：[數據分析那些事](#)

- Matplotlib API 入門，教導如何使用

刻度、標籤和圖例

過程型的 `pyplot` 介面

互動式使用，有 `xlim`, `xticks`, `xticklabels` 之類的方法，它們分別控制圖表的範圍，刻度位置，刻度標籤等。調用時不帶引數，如 `plt.xlim()` 返回當前的X軸繪圖範圍。調用時帶引數，如 `plt.xlim([0, 10])` 會將X軸的範圍設定為 0 到 10。

更為物件導向的原生 `matplotlib` API。

添加圖例的方式：添加 `subplot` 的時候傳入 `label` 引數

```
In [107]: fig = plt.figure(); ax = fig.add_subplot(1, 1, 1)
```

```
In [108]: ax.plot(randn(1000).cumsum(), 'k', label='one')
```

```
Out[108]: [<matplotlib.lines.Line2D at 0x11d814240>]
```

```
In [109]: ax.plot(randn(1000).cumsum(), 'k--', label='two')
```

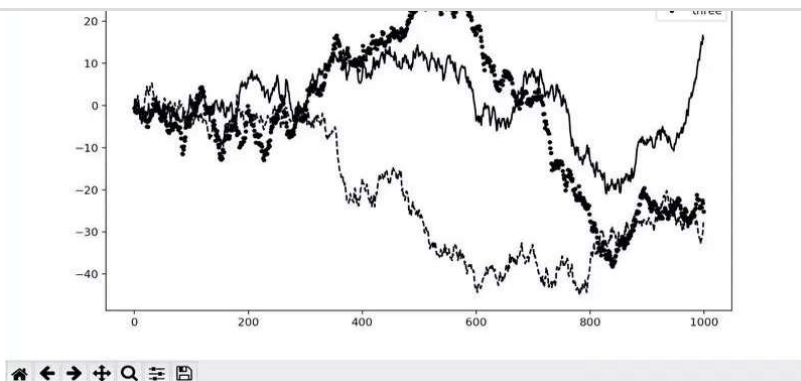
```
Out[109]: [<matplotlib.lines.Line2D at 0x11d81bcc0>]
```

```
In [110]: ax.plot(randn(1000).cumsum(), 'k.', label='three')
```

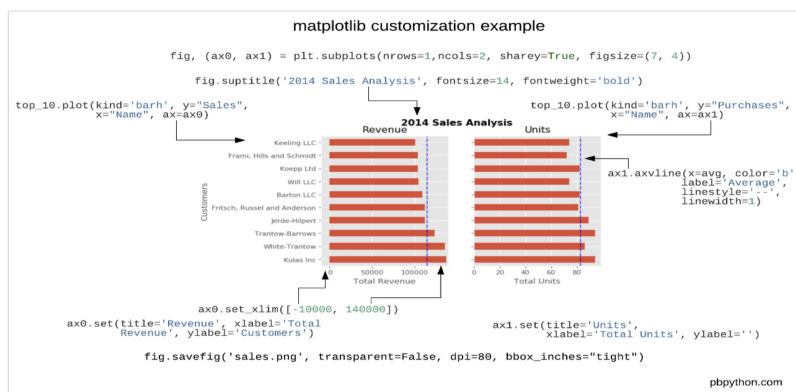
```
Out[110]: [<matplotlib.lines.Line2D at 0x11d82d668>]
```

```
In [111]: ax.legend(loc='best')
```

```
Out[111]: <matplotlib.legend.Legend at 0x11d82d438>
```



Matplotlib 客製化



[下一步：閱讀範例與完成作業](#)