

Python 資料視覺化工具與常見統計圖表介紹

[簡報閱讀](#)[範例與作業](#)[問題討論](#)[學習心得\(完成\)](#)

重要知識點

資料視覺化的好處

Python 資料視覺化主要
套件

安裝 Matplotlib

Matplotlib 操作方式



重要知識點



- 了解資料視覺化應用
 - 常見的統計圖形之使用情境與時機
- 完成今日課程後你應該可以了解
 - Python 資料視覺化主要套件適用場。

資料視覺化的好處

在分析的過程中

- 比方說用熱點圖來看你的 Deep learning 的 model 是對圖片中哪一部分的看得較重要，可以降維之後將資料視覺化去看資料在空間中的分佈

呈現成果，以利於表達資料處理結果

- 篩選及醒目提示資料
- 結果呈現

視覺化的前置工作

- 制定好題目先有清楚目的才有好視覺化
- 釐清測量尺度
- 資料處理與探勘

Python 資料視覺化主要套件

Matplotlib (靜態的套件)：基礎 2D 及 3D 可視化

- 是一個**最基礎**的 Python 可視化庫，作圖風格接近 MATLAB，所以稱為 matplotlib。一般都是從 matplotlib 上手 Python 數據可視化，然後開始做縱向與橫向拓展。
- matplotlib 的圖像都位於 Figure 物件中
- 基礎的套件
 - 精典圖表(直方、分箱、...), 多重子圖表(subplot)
 - 其他圖型: 密度圖、等高線圖

- 是一個基於 matplotlib 的進階視覺化效果庫，針對的點主要是資料挖掘和機器學習中的變數特徵選取，seaborn 可以用短小的代碼去繪製描述更多維度資料的可視化效果圖。
- Seaborn (靜態的套件)
 - 可以搭配 Matplotlib 建構更直觀的視覺化效果
 - 客製化圖表，進行統計數據可視化：圖表風格與顏色主題 圖表的分面繪製

Bokeh：交互資訊可視化

- 數據分析的一個業務場景就是用數據講故事，交互資訊可視化的工具就在此凸顯出了優勢。
- 是一個用於做瀏覽器端交互可視化的庫，實現分析師與數據的交互。
 - Bokeh (動態的套件，類似於 D3.js)
 - 不再需要編輯 HTML 與 JavaScript 便能製作網頁前端視覺化。

Basemap：地圖可視化

- 地理資訊數據也會是部分數據分析師的業務場景。
- 對於這類數據，可能傳統的 matplotlib/seaborn，互動屬性的 bokeh

庫。

安裝 Matplotlib

本日先行介紹 Matplotlib



- Matplotlib requires the following dependencies:
 - [Python](#) (≥ 3.6)
 - [FreeType](#) (≥ 2.3)
 - [libpng](#) (≥ 1.2)
 - [NumPy](#) (≥ 1.11)
 - [setuptools](#)
 - [cycler](#) ($\geq 0.10.0$)
 - [dateutil](#) (≥ 2.1)
 - [kiwisolver](#) ($\geq 1.0.0$)
 - [pyparsing](#)
- 使用 python 安裝
 - `Python -m pip install -U pip`
 - `Python -m pip install -U matplotlib`
- 在 Jupiter cell 安裝
 - `!pip install matplotlib`

- `conda list` (確認是否安裝成功)

Matplotlib 操作方式

Matplotlib 的全域 `pyplot` 模組互動操作

- 若是只有一張圖的話
- `import matplotlib.pyplot as plt`

物件導向形式的操作方式

- 若是有多張圖的話
- 透過控制 *figure* (*.SVG)和 *axis* (資料維度) 來操作。
- 其中 *figure* 和全域 *pyplot* 部分屬性相同。例如：`fig.text()` 對應到 `plt.fig_text()`

Matplotlib：常用參數說明

`matplotlib.pyplot.plot([x], y, [fmt], *, data=None, **kwargs)`參數說明

- `x, y`：x/y的數據特徵
- `[fmt]:format_string`：控制繪圖的格式字串，由顏色字元、風格字元和標記字元組成，包括點線的顏色、點的形狀、線的形狀
- `data`：數據集，或是 `dataframe`
- `**kwargs`：第二組或更多，（`x`，`y`，`format_string`）

繪製多組數據，舉個例子：

```
plot(x, y, data = data, color='green', marker='o',  
linestyle='dashed', linewidth=2, markersize=12)
```

- color：標記顏色
- maker：標記的型態
- linestyle：線條的型態
- linewidth：線寬
- markersize：標記的大小

顏色字元	說明	顏色字元	說明
'b'	藍色	'm'	洋紅色 magenta
'g'	綠色	'y'	黃色
'r'	紅色	'k'	黑色
'c'	青綠色 cyan	'w'	白色
'#008000'	RGB某顏色	'0.8'	灰度值字串

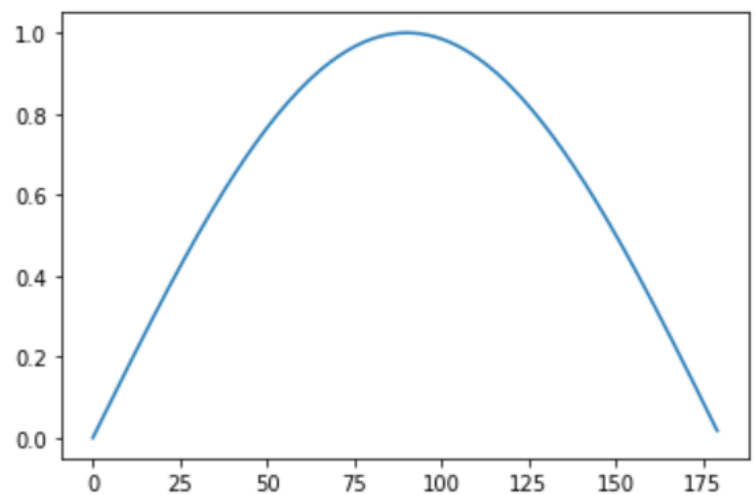
風格字元	說明
'—'	實線
'-'	破折線
'-.'	點劃線
'..'	虛線
','',''	無線條

標記字元	說明	標記字元	說明	標記字元	說明
'.'	點標記	'D'	菱形標記	'h'	豎六邊形標記
','	像素標記（極小點）	'd'	長菱形標記	'H'	橫六邊形標記
'o'	實心圓標記	' '	垂直線標記	'+'	十字標記
'v'	倒三角標記	'1'	下花三角標記	'x'	x標記
'^'	上三角標記	'2'	上花三角標記	's'	實心方形標記
'>'	右三角標記	'3'	左花三角標記	'p'	實心五角標記
'<'	左三角標記	'4'	右花三角標記	'*'	星形標記

```

1  # 載入需要的...
2
3  import matplotlib.pyplot as plt
4  import numpy as np
5
6  # 準備數據 ... 假設我要畫一個sin波 從0~180度
7
8  x = np.arange(0,180)
9  y = np.sin(x * np.pi / 180.0)
10
11 # 開始畫圖
12
13 # 設定要畫的的x,y數據list....
14
15 plt.plot(x,y)
16 # 在這個指令之前，都還在做畫圖的動作
17 # 這個指令算是 "秀圖"
18 plt.show()

```





9+



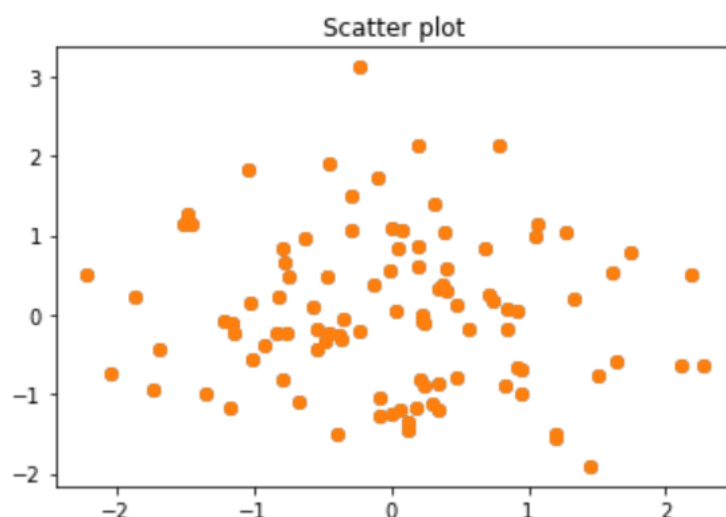
```
3 import matplotlib.pyplot as plt
4 import numpy as np
5
6 # 準備數據 ... 假設我要畫一個sin波 從0~180度
7
8 x = np.arange(0,180)
9 y = np.sin(x * np.pi / 180.0)
10
11 # 開始畫圖
12
13 # 設定要畫的的x,y數據list....
14
15 plt.plot(x,y)
16
17 # 設定圖的範圍，不設的話，系統會自行決定
18 plt.xlim(-30,390)
19 plt.ylim(-1.5,1.5)
20
21
22 # 照需要寫入 x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

散點圖：Scatter Plots

呈現資料在空間分布的情形

```
1 X = np.random.normal(0, 1, 100)
2 Y = np.random.normal(0, 1, 100)
3 plt.scatter(X, Y)
4 plt.title("Scatter plot")
```


s	大小
c	顏色
alpha	透明度
marker	型式



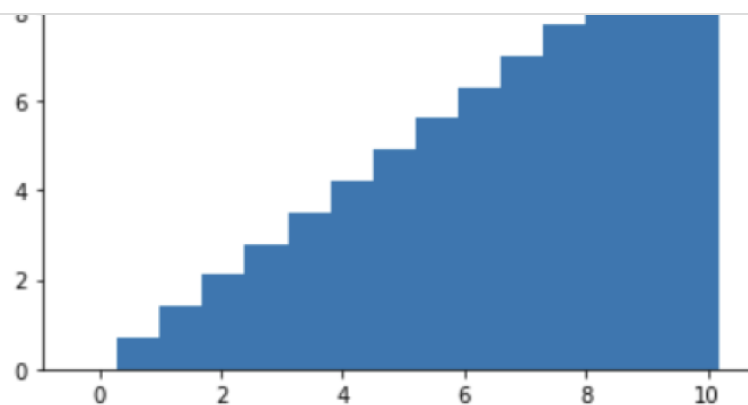
散點圖：Bar Plots

長條圖、散布圖一樣簡單，用 `plt.bar` 以及 `plt.scatter` 裡面放你的資料，以及在參數設定風格即可。

長條圖主要用來呈現兩個維度的資料，一個為 X 軸另一個則為 Y 軸(當然這邊指的是二維的狀況，較為常見)

```

1 x = np.arange(0., 10., 0.7)
2 y = np.arange(0., 10., 0.7)
3
4 plt.bar(x, y)
```



知識點回顧

優點：

- 相較於其他視覺化套件，matplotlib 算是最歷史悠久，因此有很多的教學文章或是範例可參考
- 畫圖功能最齊全，基本上沒什麼圖表畫不出來的

缺點：

- 圖表不好看(舊版的 matplotlib 很醜，但新版的 matplotlib 其實也算好看，尤其在 style 功能出來之後可以自行切換圖表的風格)
- 畫圖指令複雜

教學：

- 首先要使用 matplotlib 的話，跟 numpy 還有 pandas 一樣起手式先輸入 `matplotlib.pyplot as plt`。如果要畫折線圖使用 `plt.plot`，一開始我們先只放一個參數，這樣只會有 y 的資料，x 會是預設的 0,1,2,3...。顯示出圖片需要加上 `.show()` 否則只會顯示出這樣的訊息

延伸閱讀

Matplotlib : Visualization with Python

網站：matplotlib.org.cn

- 主要是在 [Gallery](#) 裡面有眾多範例可以提供使用參考
- 此庫包含您可以使用 Matplotlib 做的許多事情的範例。按一下任何影像以查看完整的影像和原始碼
- 線條、條形和標記
- 圖像、輪廓和欄位
- 子圖、軸和圖 (重點的基礎)
- 文本、標籤和註釋 (客製化使用的基礎)
- 餅圖和極坐標圖
- 軸網格
- 統計



Matplotlib 中文

Matplotlib中文網、Matplotlib官方中文文檔。

快速了解→

學習深度學習→

Matplotlib 是什麼？

Matplotlib 是一個Python 的2D繪圖庫，它以各種硬拷貝格式和跨平台的交互式環境生成出版質量級別的圖形。

為什麼選擇Matplotlib？

如果某天你發現自己要學習Matplotlib, 很可能是因為: 1、Matplotlib 是一個非常強大的Python 畫圖工具；2、手中有許多數據, 可是不知道該怎麼呈現這些數據。

Matplotlib 能幫你？

繪製線圖、散點圖、等高線圖、條形圖、柱狀圖、3D 圖形、甚至是圖形動畫等等。

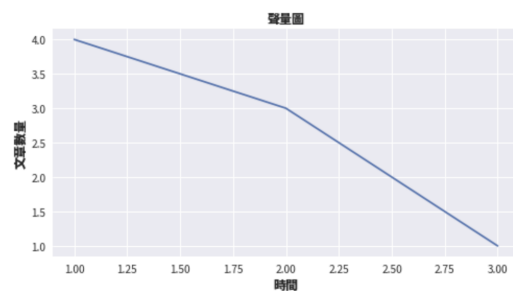
解決視覺化套件中文顯示問題

- 最基本核心是「字體」問題。通常 Anaconda 在安裝 Matplotlib 的時候會順手幫你安裝一些字體，如：DejaVu Sans, Bitstream Vera Sans, Computer Modern Sans Serif, sans-serif...等。但是這些字體本身是沒有內含中文字體的。所以每當要顯示中文時會去抓設定檔中預設的字體時，預設字體中沒有中文字型
- 解法有兩種：
 - 1. 每次程式都設定字體絕對路徑，參考範例如下：
 - [Using a ttf font file in matplotlib](#)
 - 2. 設定 rcParams 參數與 matplotlibrc 設定檔

解法有兩種：

1. 每次程式都，設定字體絕對路徑

在 Matplotlib.font_manager 這個套件底下有個 FontProperties 類別，可以指定 fname 字體路徑就可以產生 Matplotlib 指定字體的物件。官方也有是相似的 [範例檔](#)、[github text file](#)。所以每個只要有顯示中文的地方就都要加上 fontproperties 這項參數，就可以顯示中文了。只是每一個標題都要設定其實非常麻煩，所以通常會建議還是用第二種方法設定好參數檔。



[下一步：閱讀範例與完成作業](#)



[AI共學社群](#)

[我的](#)



9+

