資料聚合與視覺化

Data Aggregation & Visualization

資料聚合
Data Aggregation

聚合函數

- 聚合（Aggregation）
  - count（個數/筆數）
  - sum（總和）
  - mean（平均）
  - median（中位數）
  - std（標準差）
  - var（變異數）
  - first（第一個非NaN）
  - last（最後一個非NaN）

Grouping

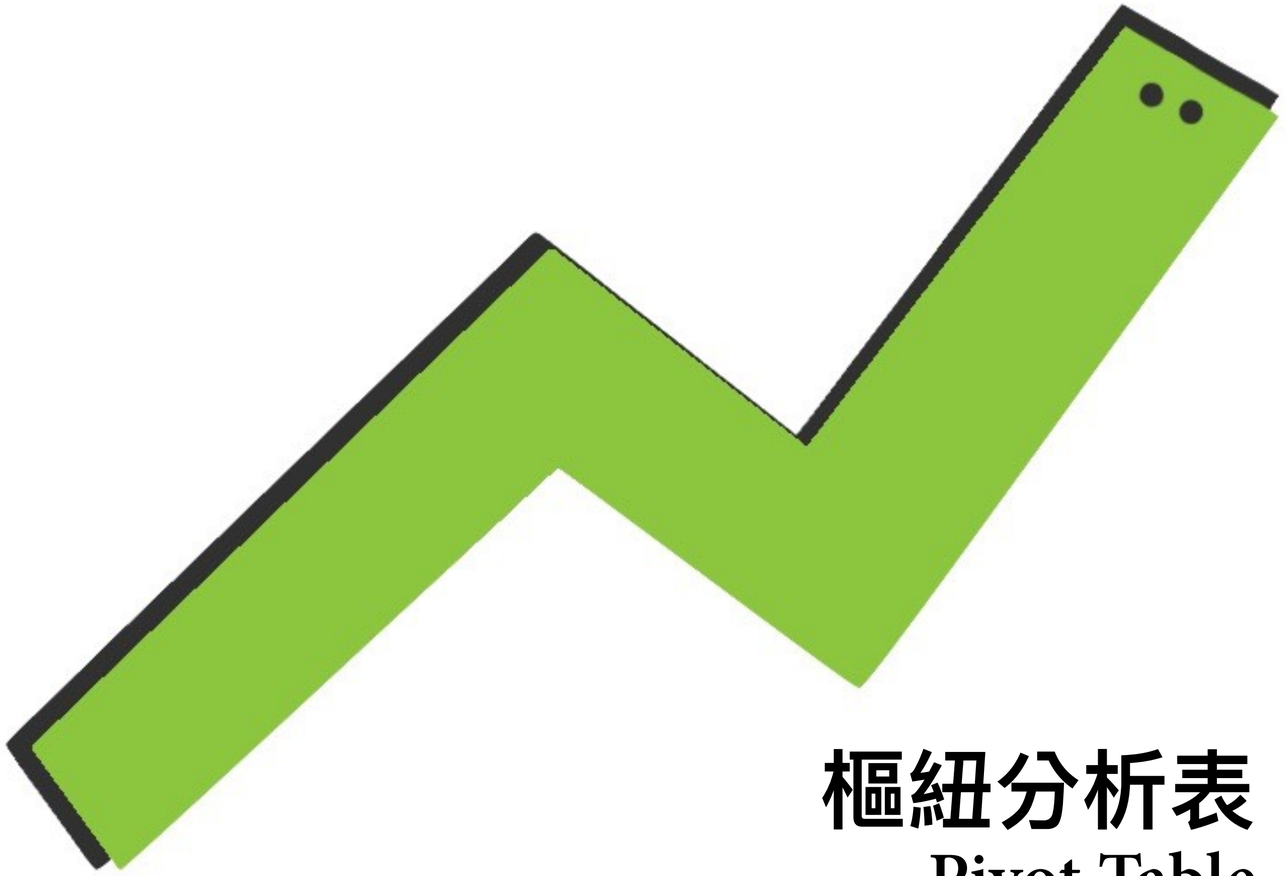
- DataFrame.groupby([*columns*]).aggregation_function()
  - e.g. df.groupby('key1').sum()
  - e.g. df.groupby(['key1','key2']).sum()

	count1	count2	key1	key2
0	10	1.2	A	A
1	20	3.4	B	B
2	23	4.5	C	D
3	42	7.4	B	B
4	51	4.4	B	D
5	76	5.5	A	A
6	65	3.4	D	C
7	80	1.2	E	D

Grouping →

	count1	count2
key1		
A	86	6.7
B	113	15.2
C	23	4.5
D	65	3.4
E	80	1.2

		count1	count2
key1	**key2**		
A	**A**	86	6.7
B	**B**	62	10.8
	D	51	4.4
C	**D**	23	4.5
D	**C**	65	3.4
E	**D**	80	1.2



樞紐分析表
Pivot Table

樞紐分析表（Pivot Table）

- 資料來源：政府開放資料平台 - 腸病毒健保就診狀況
- 問題：不同縣市每年就診類別（住院、門診）的總人次是多少？

	年	週	就診類別	縣市	腸病毒健保就診人次	健保就診總人次
0	2008	14	住院	台中市	0	104
1	2008	14	住院	台北市	2	151
2	2008	14	住院	台東縣	0	14
3	2008	14	住院	台南市	0	15
4	2008	14	住院	宜蘭縣	0	44
5	2008	14	住院	花蓮縣	0	16
6	2008	14	住院	金門縣	0	1
7	2008	14	住院	屏東縣	0	17
8	2008	14	住院	苗栗縣	0	1
9	2008	14	住院	桃園市	0	139
102	2008	14	門診	台東縣	13	926
103	2008	14	門診	台南市	75	10781
104	2008	14	門診	宜蘭縣	7	2426
105	2008	14	門診	花蓮縣	11	1517
106	2008	14	門診	金門縣	3	398
107	2008	14	門診	南投縣	25	2894
108	2008	14	門診	屏東縣	58	4267
109	2008	14	門診	苗栗縣	16	3244

樞紐分析 →

	縣市	南投縣	台中市	台北市	台南市	台東縣	嘉義市	嘉義縣	基隆市	宜蘭縣
年	就診類別									
2008	住院	14	765	739	62	6	11	30	78	13
	門診	10044	55077	26852	41396	2934	8260	4002	4759	7288
2009	住院	153	986	820	115	59	186	237	124	297
	門診	6645	46014	17825	24913	2806	6779	2948	4229	7042

樞紐分析表（Pivot Table）

- DataFrame.pivot_table(*values, index, columns, aggfunc*)
  - 不同縣市　每年　就診類別（住院、門診）的 總人次 是多少？
    column　index1　index2　aggfunc (sum)
  - e.g. df.pivot_table(values='腸病毒健保就診人次', index=['年','就診類別'], columns='縣市', aggfunc='sum')

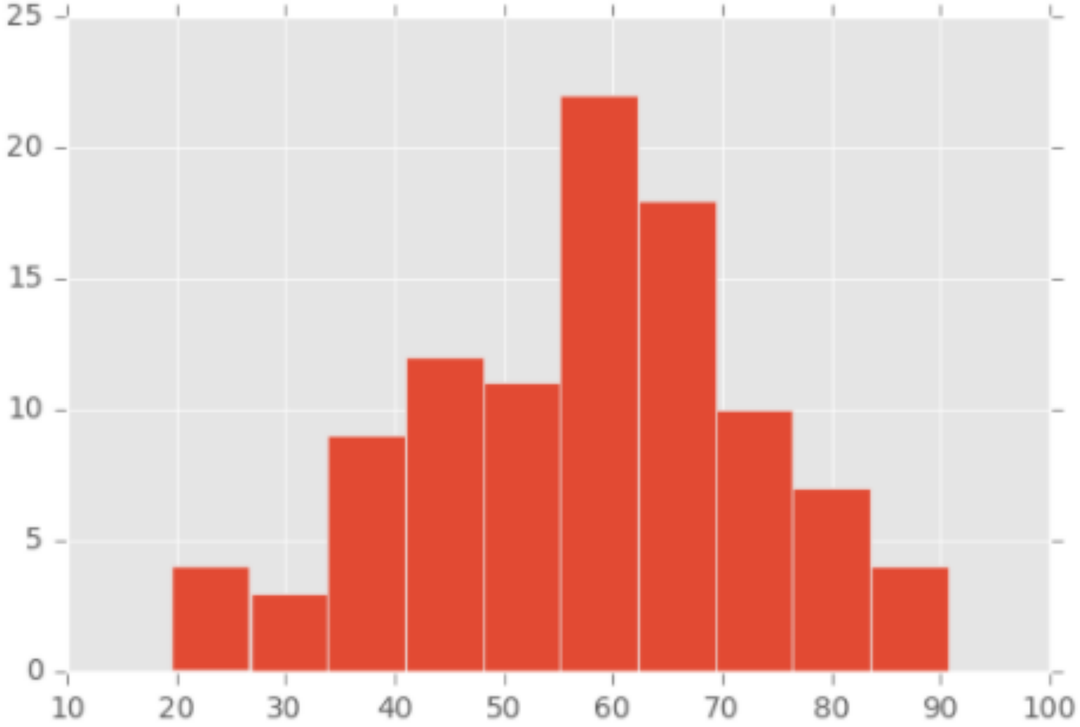
	縣市	南投縣	台中市	台北市	台南市	台東縣	嘉義市	嘉義縣	基隆市	宜蘭縣
年	就診類別									
2008	住院	14	765	739	62	6	11	30	78	13
	門診	10044	55077	26852	41396	2934	8260	4002	4759	7288
2009	住院	153	986	820	115	59	186	237	124	297
	門診	6645	46014	17825	24913	2806	6779	2948	4229	7042

Matplotlib 基礎視覺化

直方圖、長條圖、折線圖、散佈圖、箱形圖、圓餅圖

直方圖 (Histogram)

- 適用於呈現數據分佈的情形，如：成績分佈
  - matplotlib.pyplot.hist (x,bins)

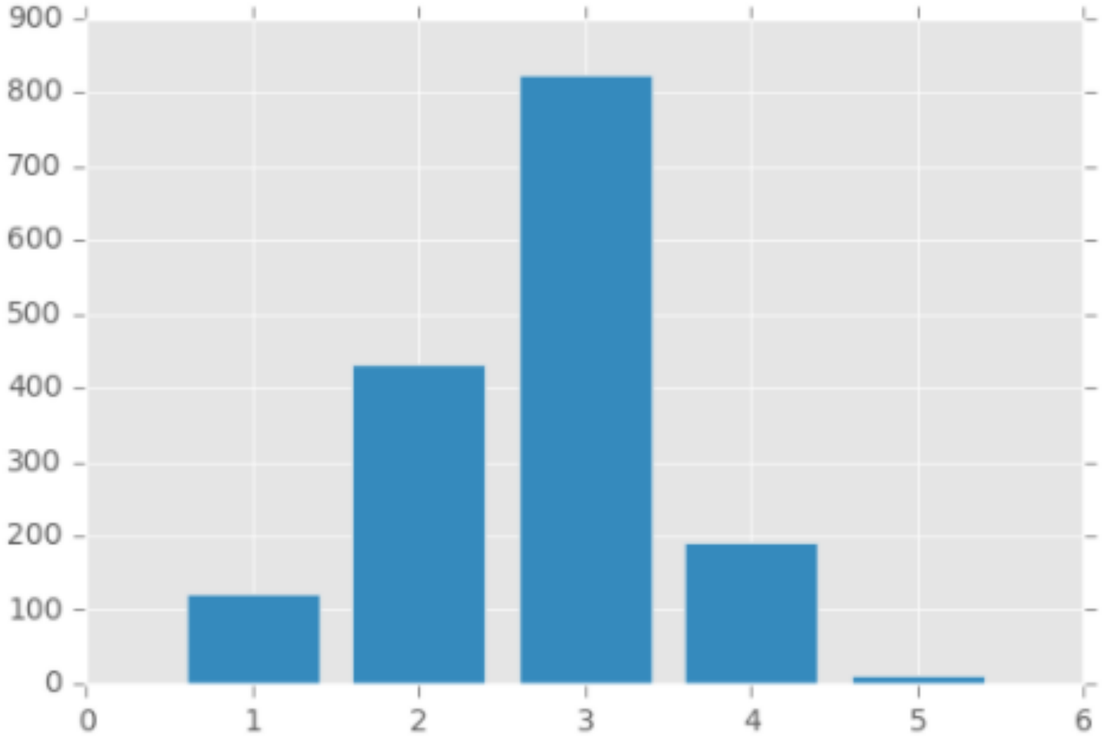


Notes

- x: 原始資料
- bins: 分箱/分組的範圍設定

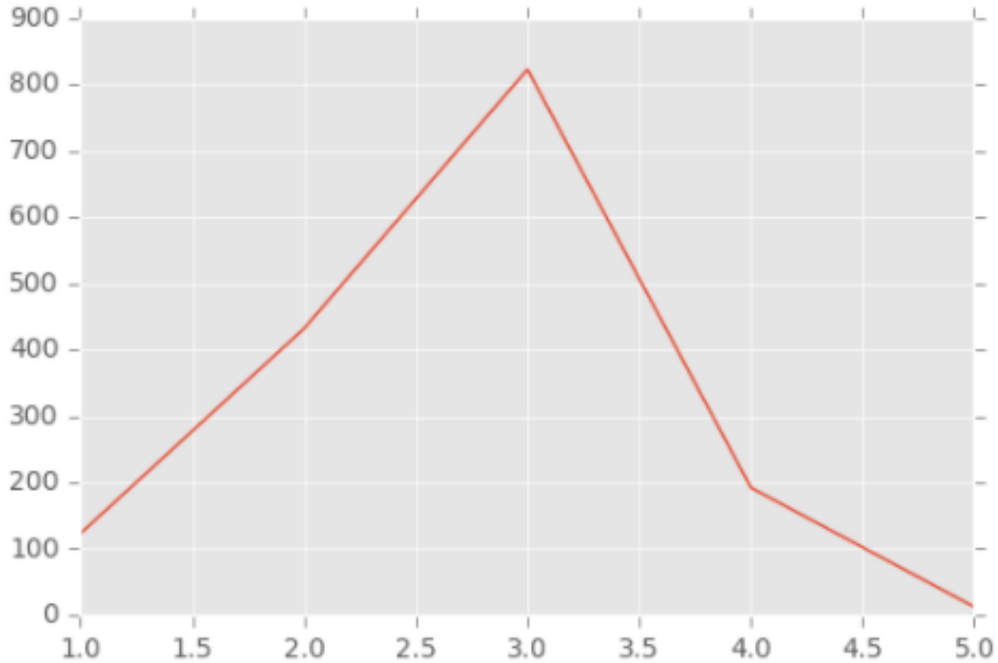
長條圖 (Bar Chart)

- 適用於呈現數據大小的比較
  - matplotlib.pyplot.bar (x, y)



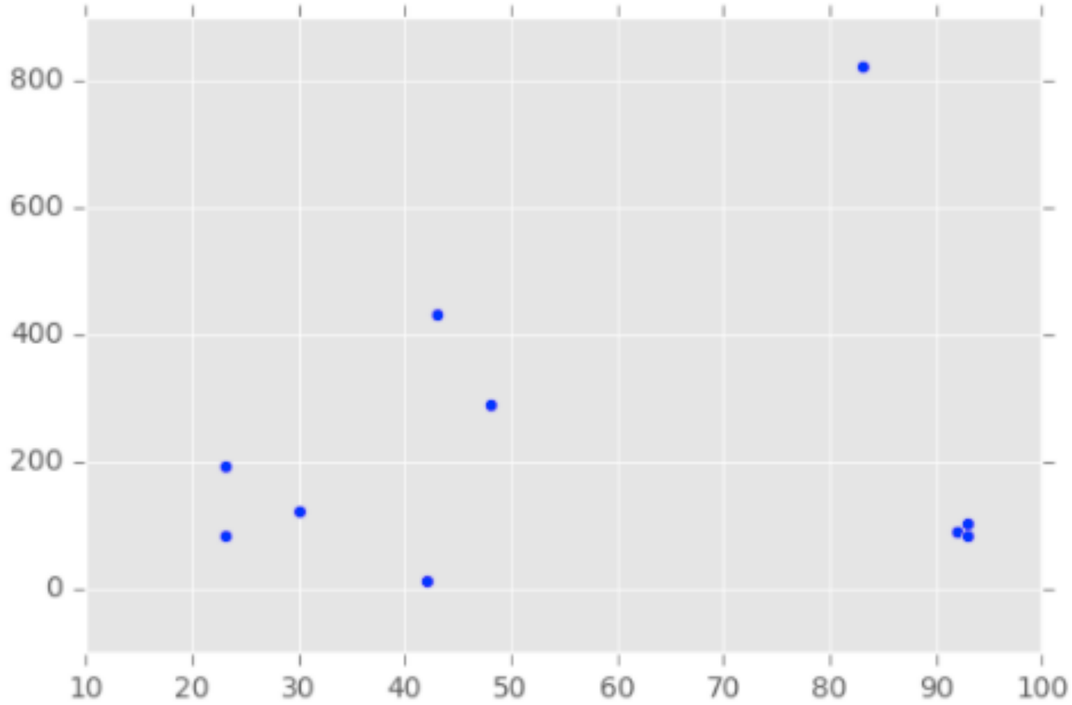
折線圖 (Line Chart)

- 適用於呈現數據變化的趨勢
  - matplotlib.pyplot.plot (x, y)



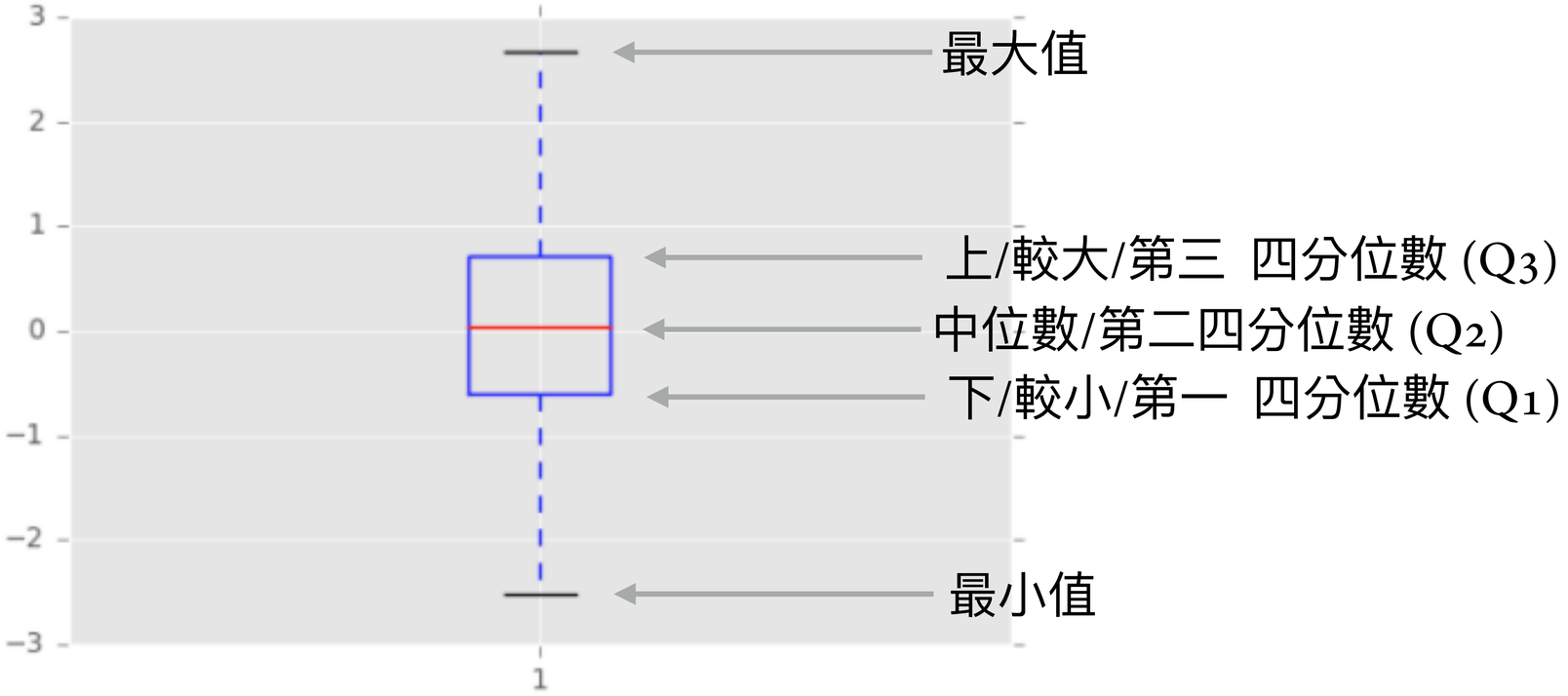
散佈圖（Scatter Diagram）

- 適合呈現變項（X, Y）間的相關性
  - matplotlib.pyplot.scatter (x, y)



箱形圖（Box Plot）

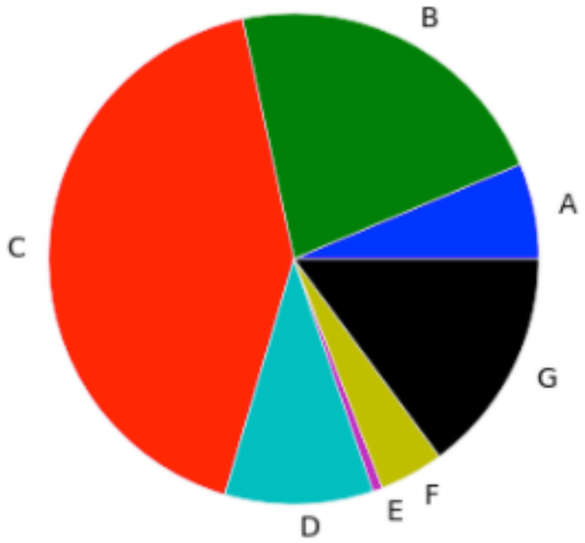
- 又稱盒鬚圖、箱線圖、盒式圖等，適合呈現數據分散的情形
  - matplotlib.pyplot.boxplot(x, showfliers=False)



註：有些數據被視為應忽略的異常值而不納入作圖，
後續介紹「異常值偵測」時會再詳細說明

圓餅圖

- 呈現不同事物量的佔總量的百分比，如：全球母語人口比例
- 適合用於呈現其中一項占比特別高的情況，但不建議用於呈現比例接近的情況，因為圓餅圖的角度難以比較數值的高低
  - matplotlib.pyplot.pie(y,labels)

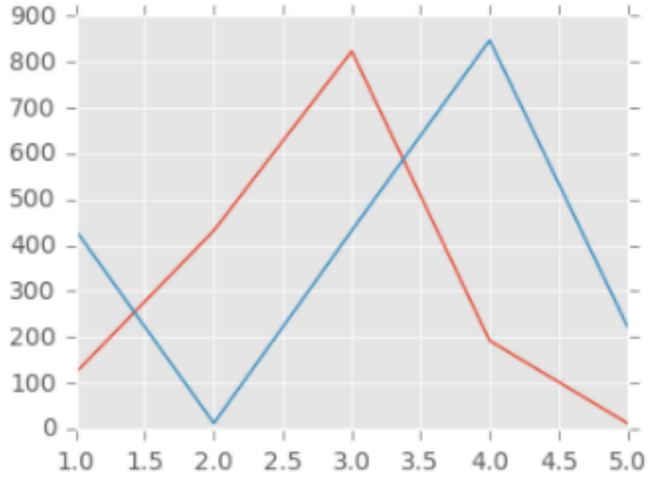


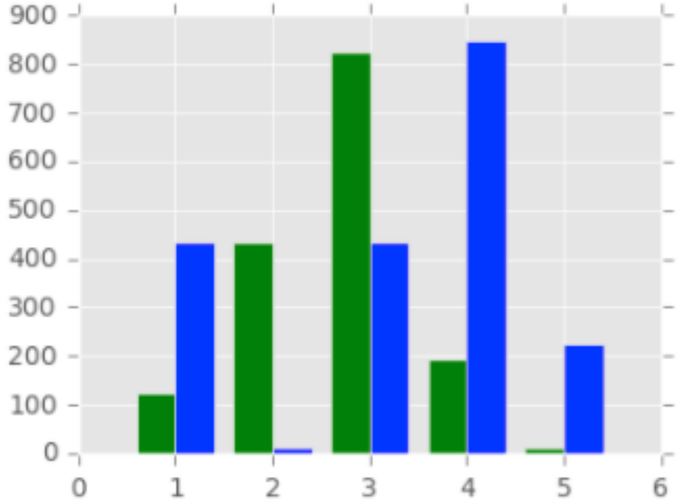
Matplotlib 視覺化功能

作圖及線條風格、顏色、標題、中文處理、多圖合一等

X軸雙變項作圖

- 折線圖：
  - matplotlib.pyplot.plot (x1, y1)
  - matplotlib.pyplot.plot (x2, y2)

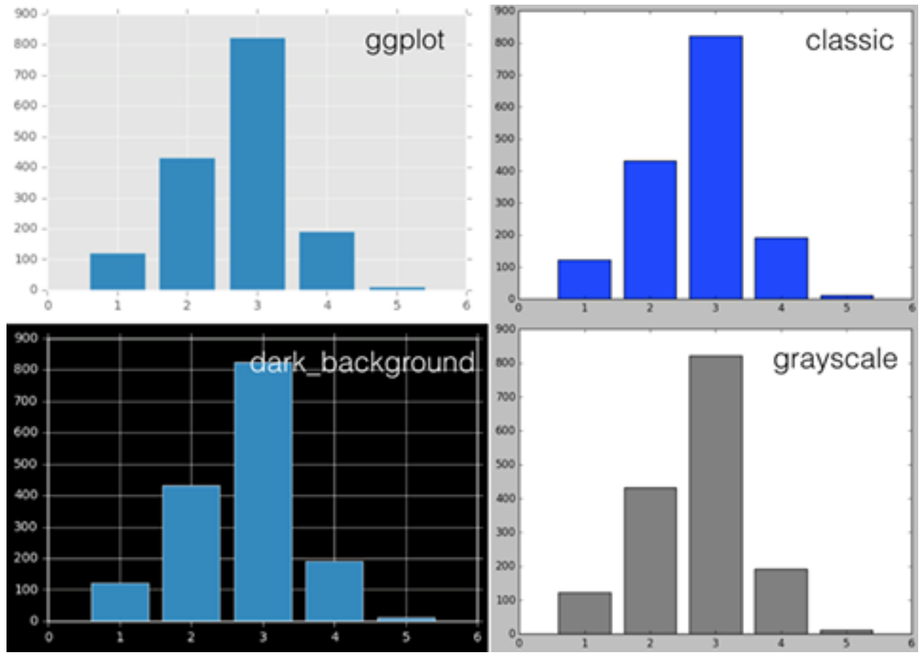




- 長條圖：
  - 調整 X 位置、顏色（color）、寬度（width）

作圖風格

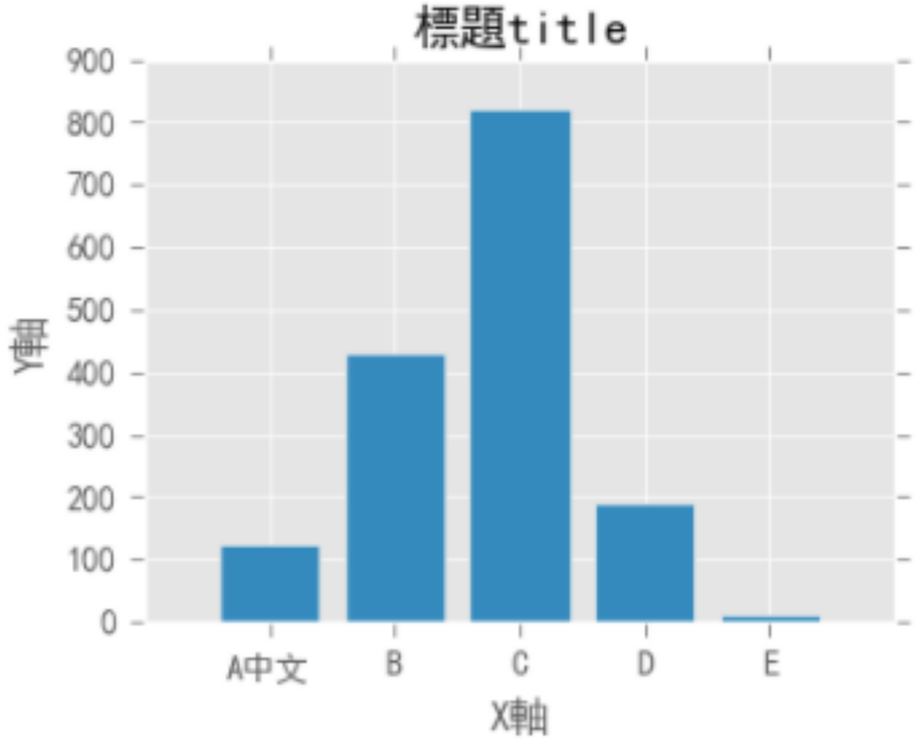
- 內建21種作圖風格：
  - 查看所有的風格選擇：print(plt.style.available)
  - 使用作圖風格：plt.style.use('風格名稱')
  - 預設：classic、建議：ggplot



- 所有作圖風格呈現效果：https://tonysyu.github.io/raw_content/matplotlib-style-gallery/gallery.html

標題、XY軸、X軸刻度名稱

- 標題名稱
  - plt.title('標題title')
- X軸名稱
  - plt.xlabel('X軸名稱')
- Y軸名稱
  - plt.ylabel('Y軸名稱')
- X軸tick名稱
  - plt.xticks(x, ['A','B','C','D','E'])



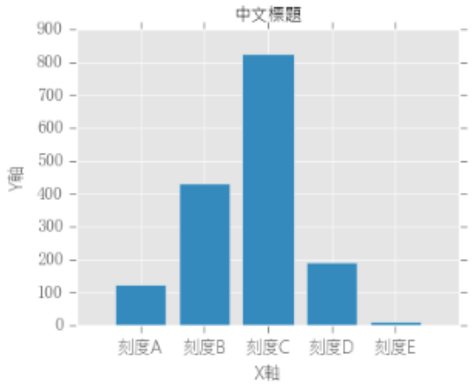
顯示中文

- 更改全部作圖字體
  - plt.rcParams[‘font.family']='SimHei' #黑體
- 每次指定特定字體
  - e.g. font = {'fontname':'Times New Roman’}
  - e.g. plt.title(‘Title',**font)
- 列出所有可用的字體
  - [f.name for f in matplotlib.font_manager.fontManager.ttflist]

顯示中文 (cont.)

- Windows可用的中文字體：Microsoft YaHei (Win7)、Microsoft MHei (Windows)、DFKai-SB(Win10)
- 指定字體路徑

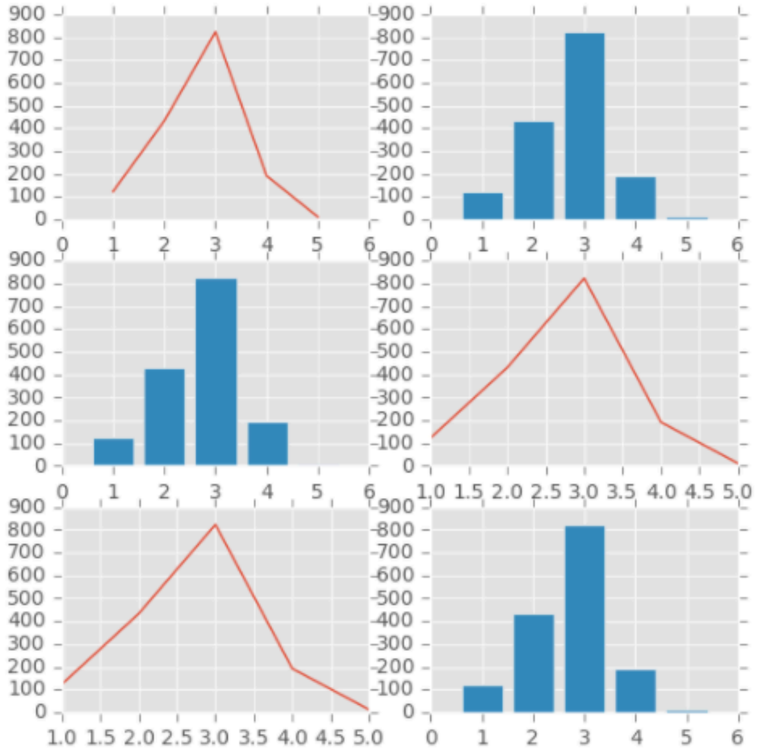
In [19]:
```
import matplotlib.pyplot as plt
from matplotlib.font_manager import FontProperties
plt.style.use('ggplot')

fig = plt.figure(figsize=(4,3))
x = [1,2,3,4,5]
y = [123,432,823,192,12]

#plt.rcParams['font.family']=''
font = FontProperties(fname=r'C:\Windows\Fonts\msjhl.ttc')

plt.bar(x,y,align='center')
plt.title('中文標題', fontproperties=font) #標題名稱
plt.xlabel('X軸' ,fontproperties=font) #X軸名稱
plt.ylabel('Y軸',fontproperties=font) #Y軸名稱
plt.xticks(x, ['刻度A','刻度B','刻度C','刻度D','刻度E'],fontproperties=font) #X軸刻度名稱

plt.show()
```



多圖合一 - 作圖位置

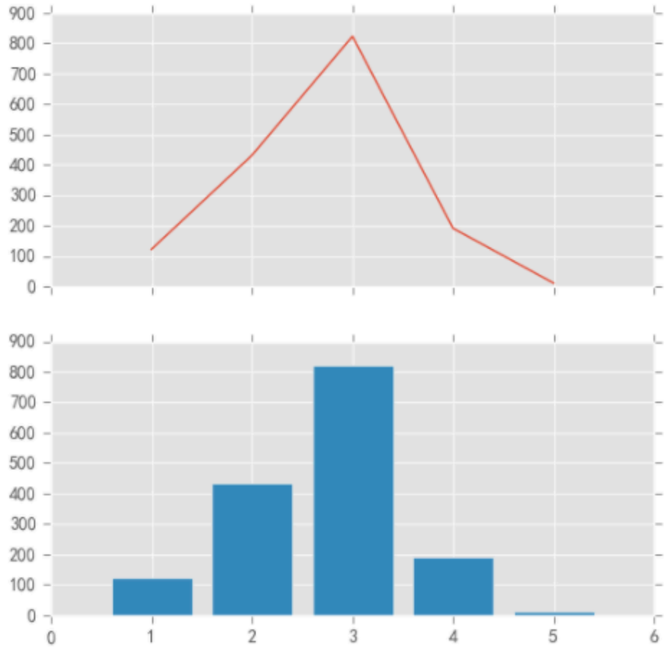
- plt.subplot()
  - plt.subplot(321) # 3 x 2的圖位置1
  - plt.bar(x,y) # 作圖
- 3 x 2

3 \ 2		
	1	2
	3	4
	5	6



多圖合一 - 共用X軸

- 共用X軸
  - 回傳子圖1的x軸：e.g. ax1 = plt.subplot(211) #2x1的第1張圖
  - 設定作圖property裡子圖1的x軸不顯示：e.g. plt.setp(ax1.get_xticklabels(), visible=False)
  - plt.subplot(212,sharex=ax1) #2x1的第2張圖（共用ax1）



顏色

- 快速使用：

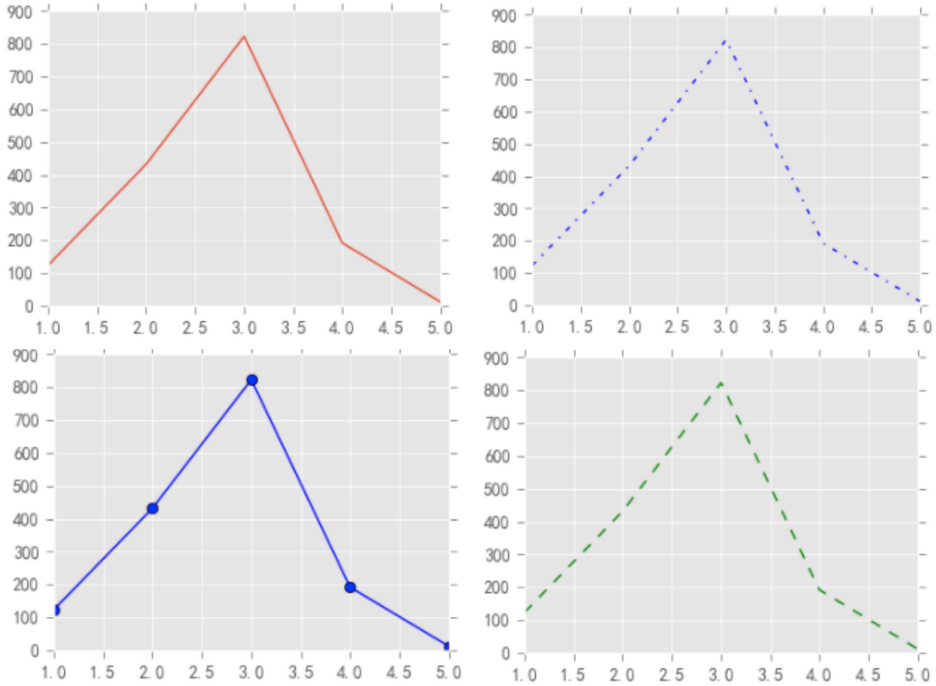
代碼	全名	顏色	效果
b	blue	籃	
g	green	綠	
r	red	紅	
c	cray	青	
m	magenta	洋紅	
y	yellow	黃	
k	black	黑	
w	white	白	

- RGB色碼：
  - e.g. plt.plot(x,y,color='#6ACC65')

折線圖線條種類

- plt.plot(*x, y, color, linestyle*)

linestyle	description
'-' or 'solid'	solid line
'--' or 'dashed'	dashed line
'-.' or 'dashdot'	dash-dotted line
':' or 'dotted'	dotted line
'None'	draw nothing
' '	draw nothing
''	draw nothing



（表格來源：matplotlib官方文件）

所有折線圖線條種類(補充)

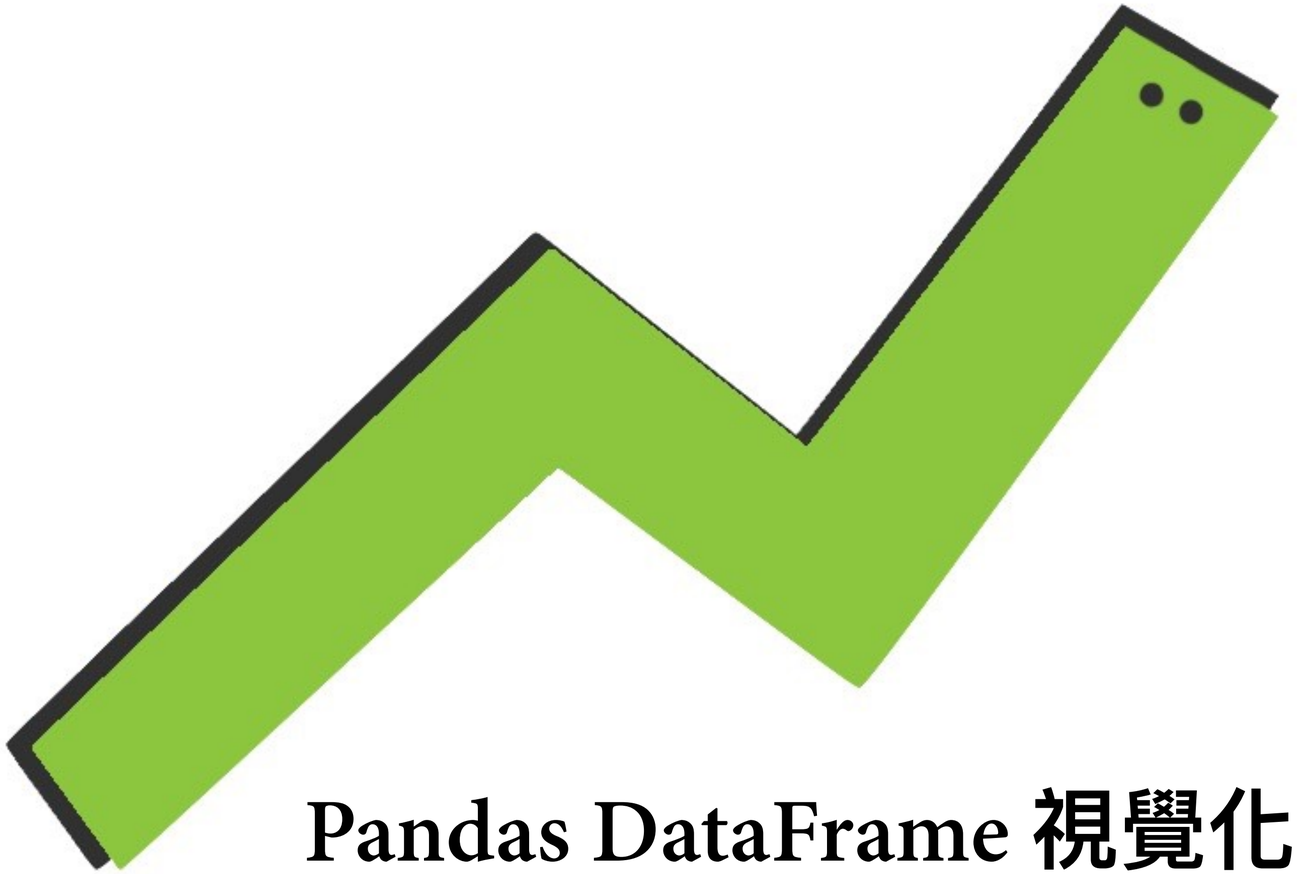
character	description
'-'	solid line style
'--'	dashed line style
'-.'	dash-dot line style
':'	dotted line style
'.'	point marker
','	pixel marker
'o'	circle marker
'v'	triangle_down marker
'^'	triangle_up marker
'<'	triangle_left marker
'>'	triangle_right marker
'1'	tri_down marker
'2'	tri_up marker
'3'	tri_left marker
'4'	tri_right marker

's'	square marker
'p'	pentagon marker
'*'	star marker
'h'	hexagon1 marker
'H'	hexagon2 marker
'+'	plus marker
'x'	x marker
'D'	diamond marker
'd'	thin_diamond marker
'|'	vline marker
'_'	hline marker

(表格來源：matplotlib官方文件)

References

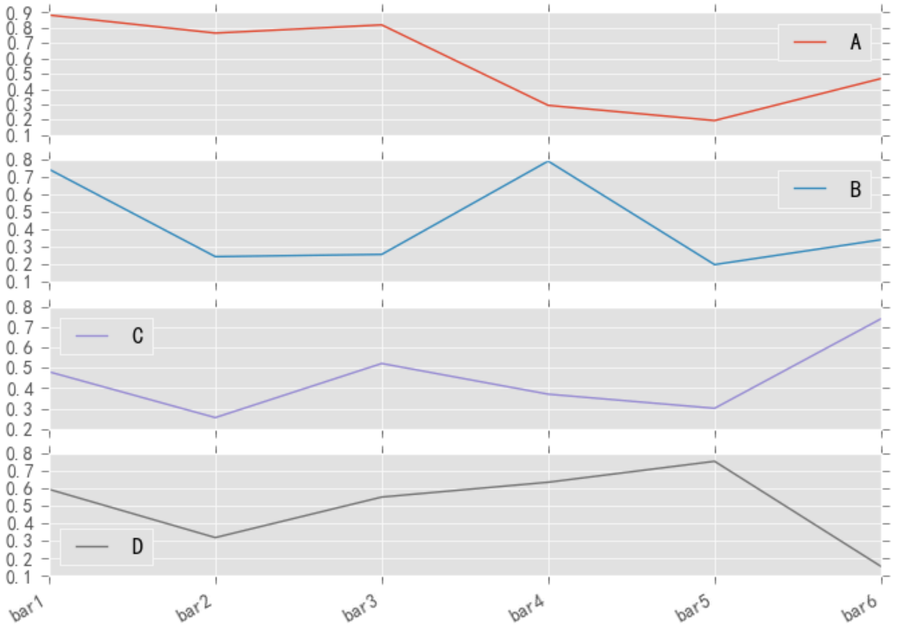
- Matplotlib官方pyplot API文件：
- http://matplotlib.org/api/pyplot_api.html

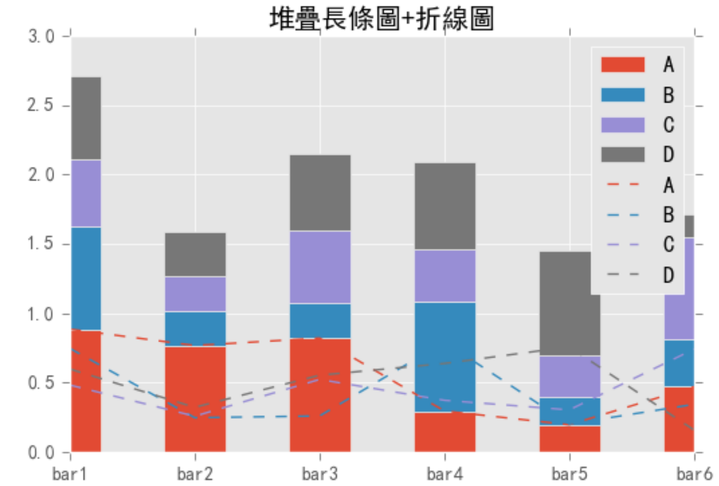


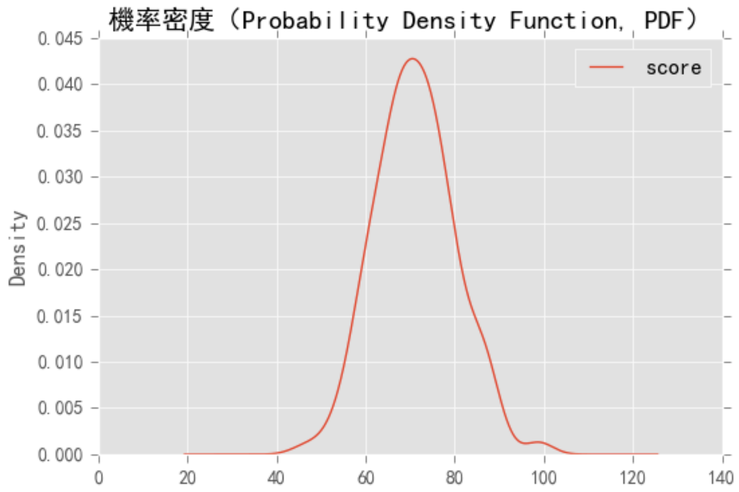
Pandas DataFrame 視覺化

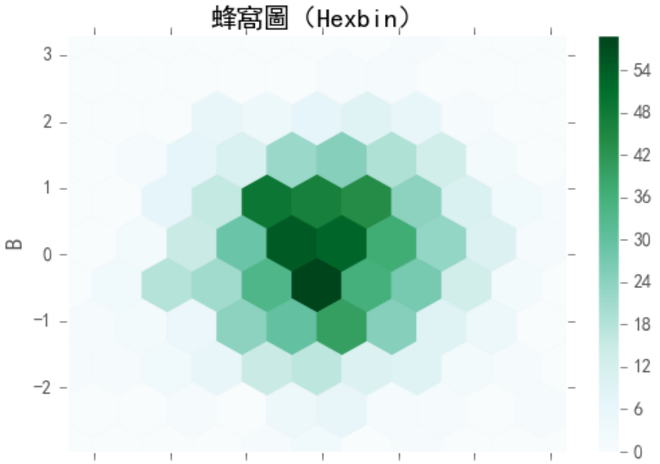
Pandas 視覺化

- 豐富的視覺化種類
- 為DataFrame快速作圖









作圖函式與種類

- DataFrame.plot(*x=None, y=None, kind='line', ax=None, subplots=False, sharex=None, sharey=False, layout=None, figsize=None, use_index=True, title=None, grid=None, legend=True, style=None, xticks=None, yticks=None, fontsize=None, sort_columns=False*) （0.19.1版）
- kind（作圖種類）
  - ‘line’ : line plot (default) （折線圖）
  - ‘bar’ : vertical bar plot （垂直長條圖）
  - ‘barh’ : horizontal bar plot （水平長條圖）
  - ‘hist’ : histogram （直方圖）
  - ‘box’ : boxplot（箱型圖）
  - ‘kde’ : Kernel Density Estimation plot （機率密度圖）
  - ‘density’ : same as ‘kde’ （機率密度圖）
  - ‘area’ : area plot （面積圖）
  - ‘pie’ : pie plot （圓餅圖）
  - ‘scatter’ : scatter plot（散佈圖）
  - ‘hexbin’ : hexbin plot（蜂窩圖）

作圖參數說明與功能

- subplot（每一欄都作出一張子圖）：True/False
- sharex（共用x軸）：True/False
- sharey（共用y軸）：True/False
- figsize（作圖尺寸）：(width, height)
- title（標題）：string
- grid（格線）：True/False
- legend（圖例）：True/False
- style（風格）：list or dict（每一欄的作圖風格）
- xticks（x刻度）：list
- yticks（y刻度）：list
- fontsize（字體大小）：int
- stacked（堆疊）：True/False
- sort_columns（依照column名稱排序作圖）：True/False

References

- Pandas官方網站：http://pandas.pydata.org/
- Pandas官方文件：http://pandas.pydata.org/pandas-docs/stable/
- Pandas 0.19.0版手冊pdf：http://pandas.pydata.org/pandas-docs/version/0.19.0/pandas.pdf
- Pandas DataFrame.plot函式文件：http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.plot.html

註：Pandas的功能非常豐富強大，課程時間有限僅介紹Pandas重要常用的函式功能，建議大家再自行查閱以上資源。

補充：df.plot()及Axes的使用

- df.plot()是DataFrame快速作圖的函式，作完圖後若需要plot()裡沒有的進一步細部作圖項目（如：設定X、Y軸標題、共用XY軸等、XY軸刻度範圍等），要使用df.plot()回傳Axes物件中的函式，例如：

```
In [10]: ax = df.plot(kind='bar',title='長條圖',figsize=(6,4))
         ax.set_xlabel('X軸標題')
         ax.set_ylabel('Y軸標題')

Out[10]: <matplotlib.text.Text at 0x111627c18>
```