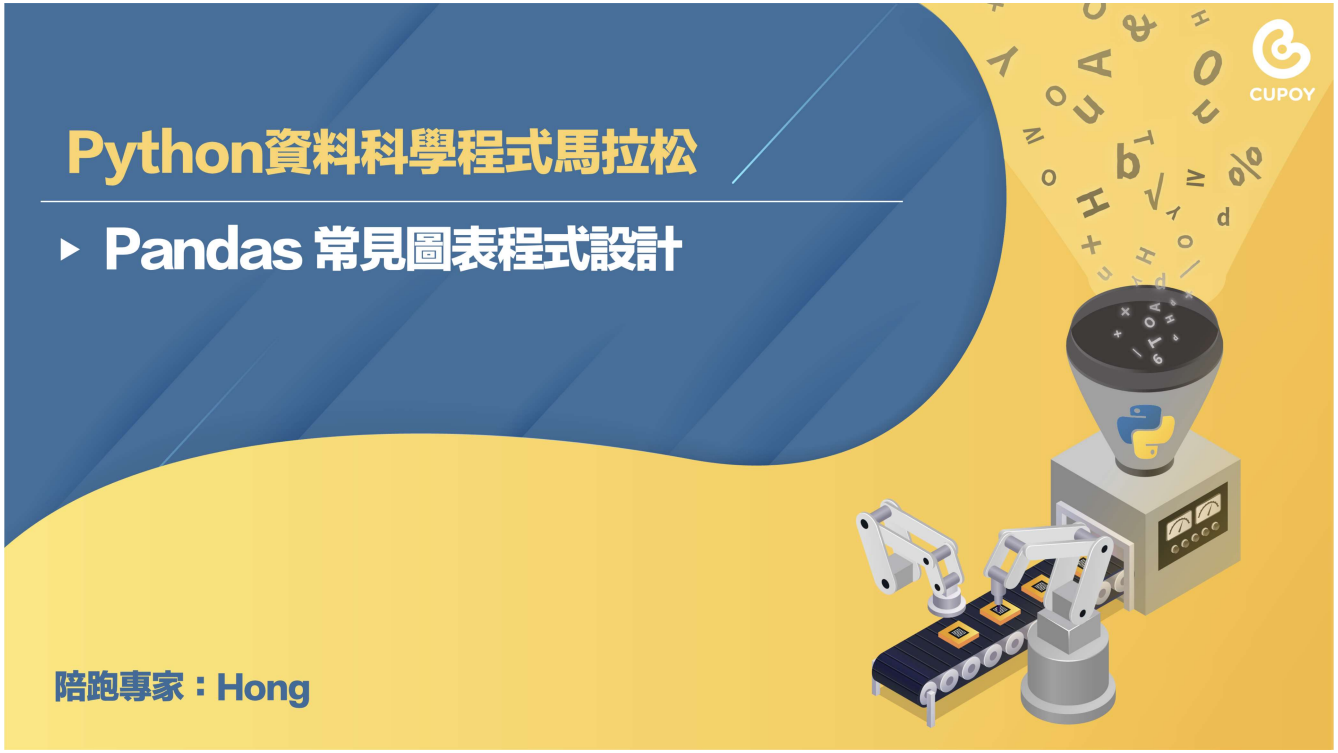
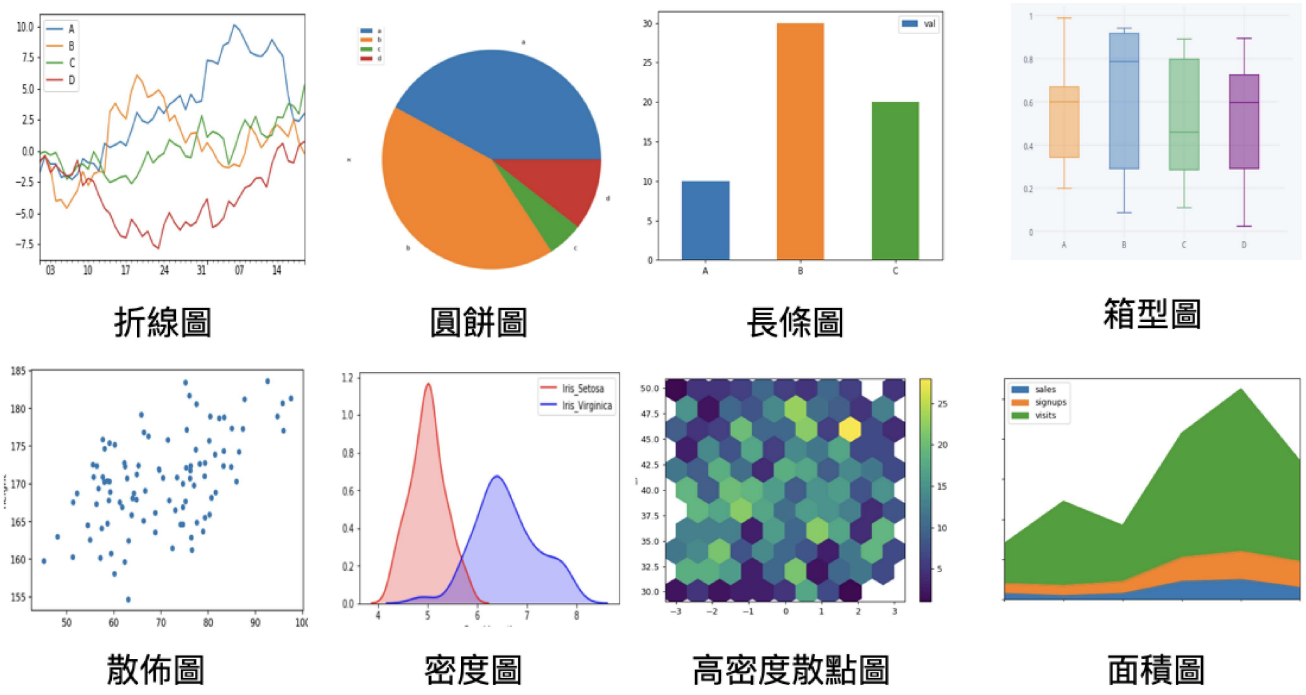


重要知識點

- 介紹常見的圖表種類與對應使用情境
- 介紹如何在 pandas 中繪製圖表

認識常見圖表



折線圖

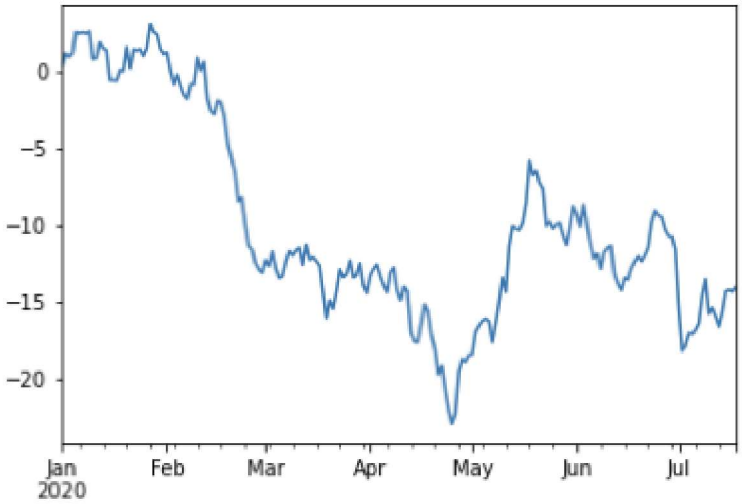
- 適用：會隨時間變動的值，呈現趨勢變化
- 實際應用：匯率
- 由匯率折線圖可看出，美金從九月中旬至今仍持續貶值中

折線圖 in pandas plot

- 折線圖可以簡單的在 pandas 資料使用 .plot() 就可以畫出來
- 其中 x 軸部分為資料集的 index，y 軸為資料集欄位值，如下圖紅框資料。

```
ts = pd.Series(np.random.randn(200), index=pd.date_range('1/1/2020', periods=200))
ts = ts.cumsum()
print(ts)
ts.plot()
```

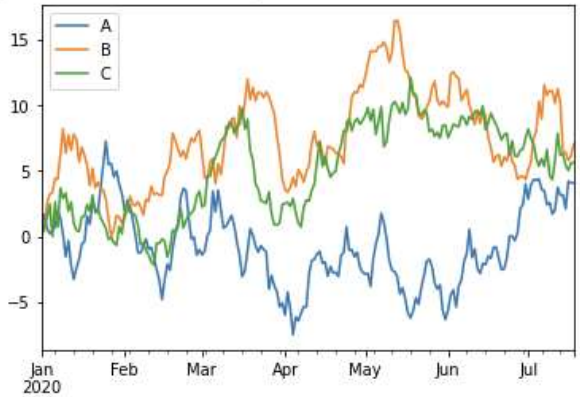
```
2020-01-01     0.296190
2020-01-02     1.122342
2020-01-03     0.887218
2020-01-04     1.105221
2020-01-05     2.523833
                 ...
2020-07-14   -15.682662
2020-07-15   -14.335729
2020-07-16   -14.244317
2020-07-17   -14.353459
2020-07-18   -14.051700
Freq: D, Length: 200, dtype: float64
<matplotlib.axes._subplots.AxesSubplot at 0x7f7878a50f60>
```



如果資料有多個欄位時，也可以同時話出多筆資料在同一張圖上。

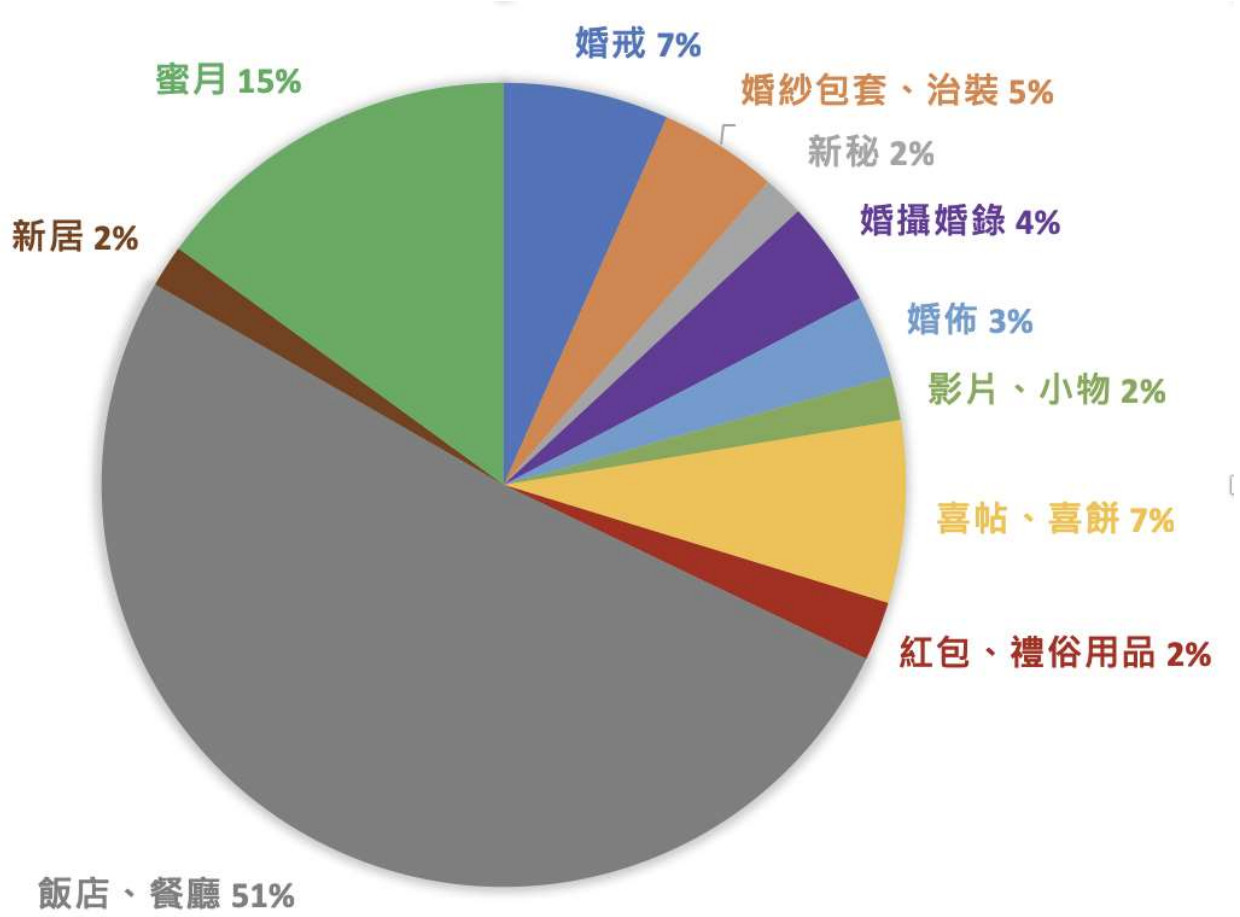
```
df = pd.DataFrame(np.random.randn(200, 3), index=pd.date_range('1/1/2020', periods=200), columns=["A","B","C"])
df = df.cumsum()
print(df)
df.plot();
```

```
                   A         B         C
2020-01-01  0.719841  0.536558  0.046504
2020-01-02  1.716844  0.419193  0.788143
2020-01-03  0.598808  2.551977  1.421477
2020-01-04  0.232542  3.229406  2.481343
2020-01-05  0.404765  3.357299  0.003117
...              ...       ...       ...
2020-07-14  3.164508  6.438139  6.293626
2020-07-15  2.119847  6.433417  5.412571
2020-07-16  4.215448  5.808347  5.042872
2020-07-17  4.099448  6.174222  5.627170
2020-07-18  4.071364  7.091973  5.615444

[200 rows x 3 columns]
```



圓餅圖

- 適用：呈現不同種類資料，在整體資料所佔比例
- 實際應用：記帳
- 由記帳園餅圖可看出，舉辦婚宴時一半以上的開銷都花在飯店、餐廳

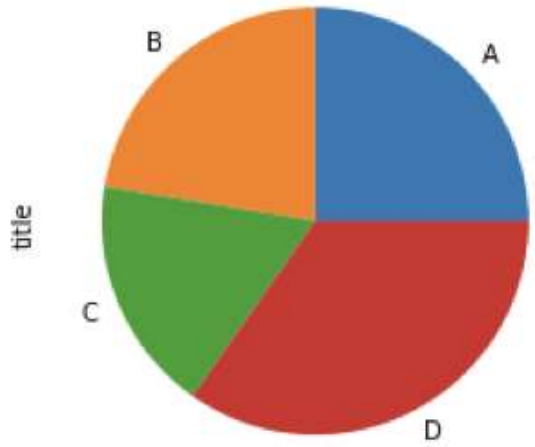


圓餅圖 in pandas plot

- 圓餅圖，其中資料 index 為圓餅圖類別 ABCD，欄位部份為 ABCD 所對應數值， .plot.pie() 會依據所對的數值計算比例畫出圓餅圖。
- 參數 name 為圓餅圖的名稱

```
#圓餅圖
df  =  pd.Series(np.random.rand(4),  index=["A","B","C","D"],name="title")
print(df)
df.plot.pie()
```

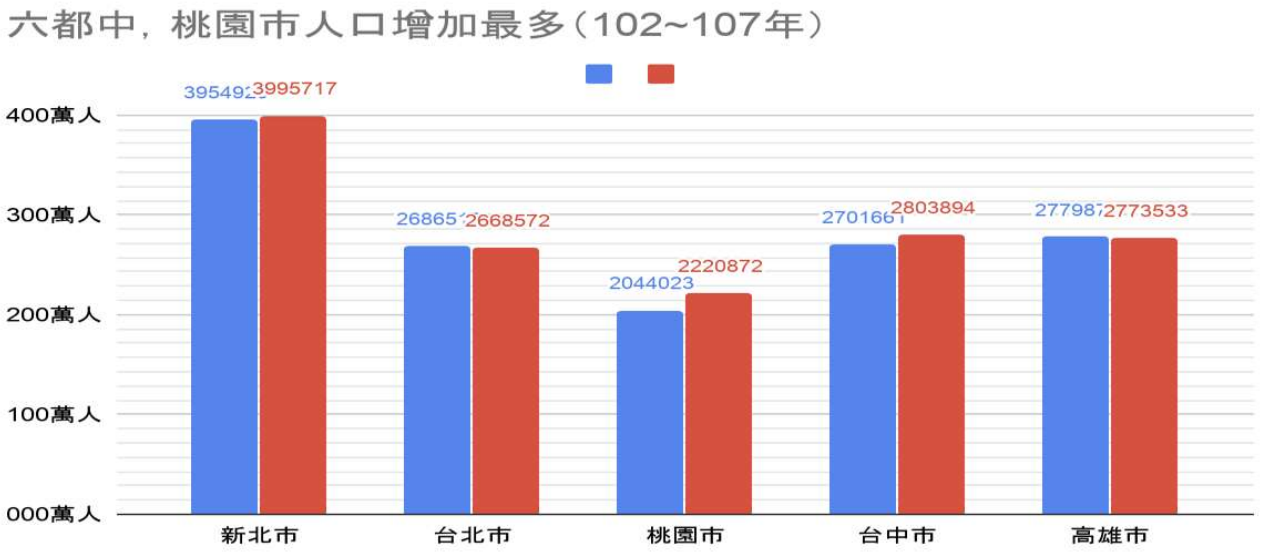
```
A    0.671026
B    0.601189
C    0.478511
D    0.932186
Name: title, dtype: float64
<matplotlib.axes._subplots.AxesSubplot at 0x7f7871060240>
```



長條圖

- 適用：不同種類資料，在不同時間點的變化

- 實際應用：人口成長變化
- 由人口長條圖可看出，從 102 年到 107 年新北市人口都是六都之冠；桃園市是六都中人口成長最快速的城市

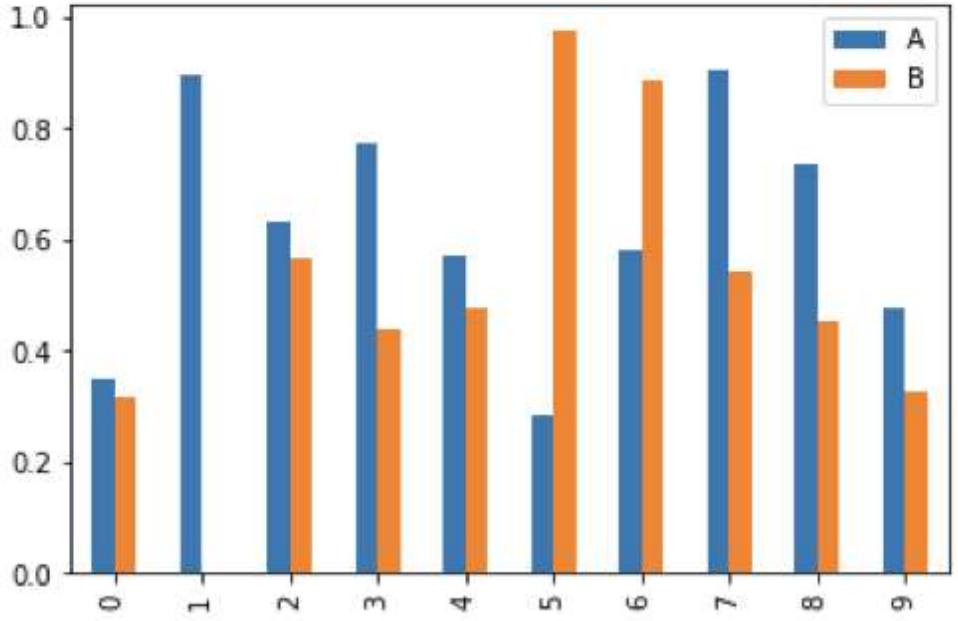


長條圖 in pandas plot

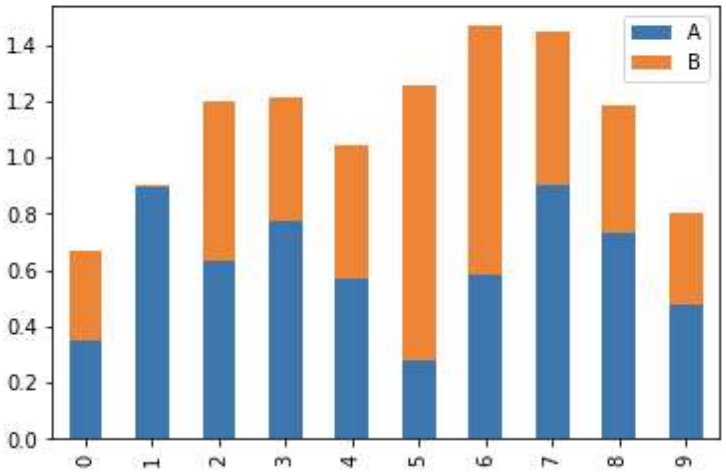
- 長條圖，其中資料 index 為 x 軸，欄位部份為 AB 所對應數值，.plot.bar() 會依據所對的數值畫出長條圖。
- 參數 stacked 可以將長條圖疊加

```
#長條圖
df = pd.DataFrame(np.random.rand(10, 2), columns=["A","B"])
print(df)
df.plot.bar();
```

```
          A         B
0  0.351182  0.318318
1  0.897806  0.003340
2  0.632099  0.567870
3  0.772415  0.441826
4  0.570837  0.476311
5  0.282301  0.974770
6  0.582213  0.884616
7  0.904212  0.543904
8  0.734159  0.452428
9  0.475335  0.328955
```

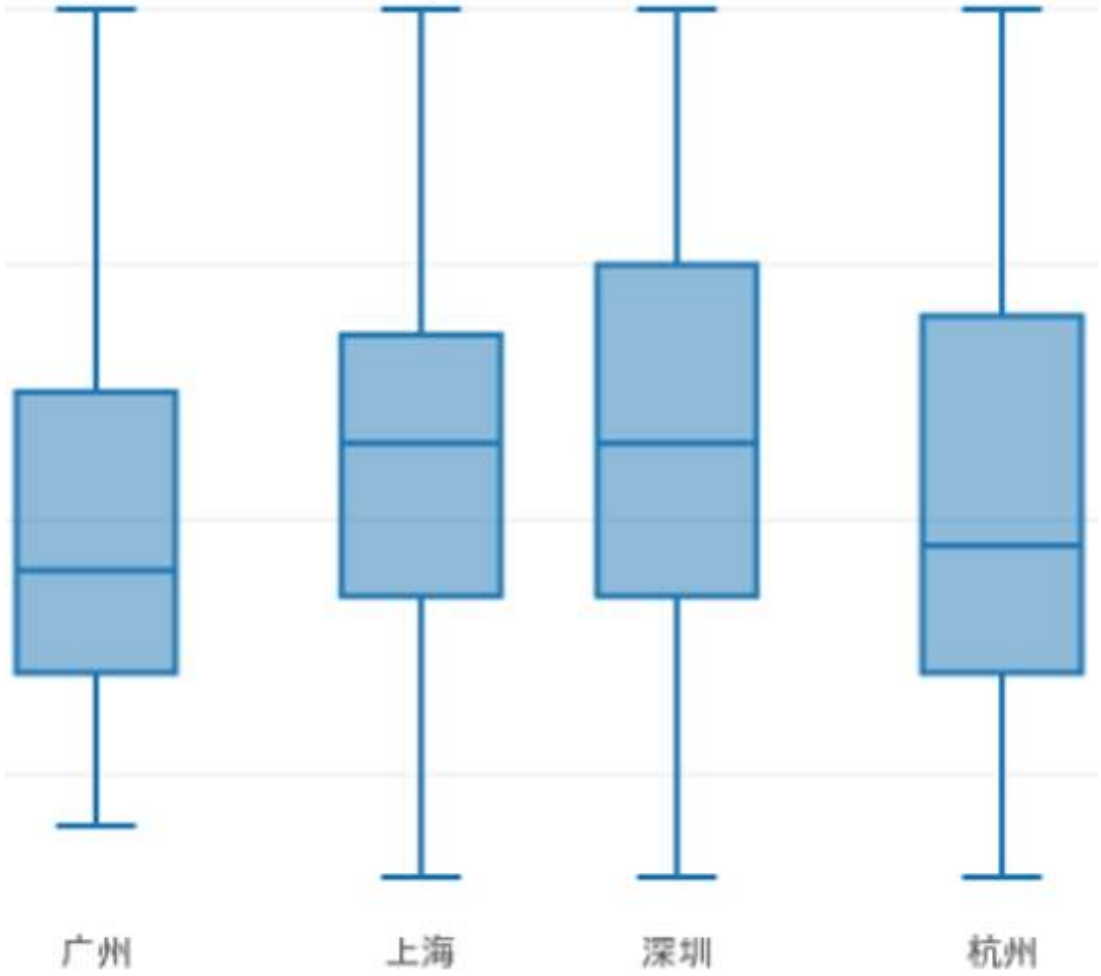


```
df.plot.bar(stacked=True);
```



箱型圖

- 適用：完整呈現數值分布的統計圖表
- 實際應用：薪資水平
- 由薪資箱型圖可看出，廣州的最低薪資是四個城市中最高的，而深圳是薪資水平最高的城市

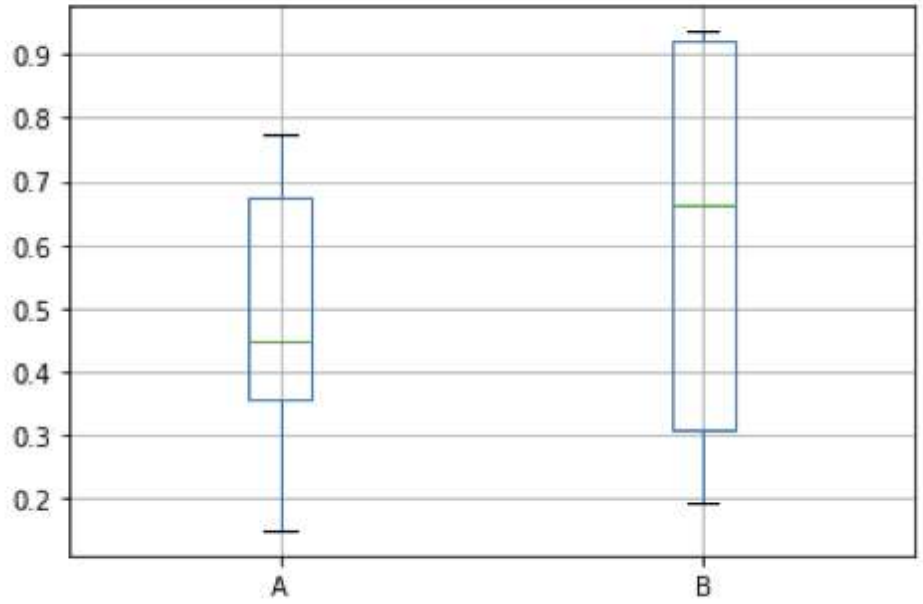


箱型圖 in pandas plot

- 箱型圖，其中資料 index 為 x 軸，欄位部份為 AB 所對應數值，.boxplot() 會依據所對的數值畫出箱型圖。

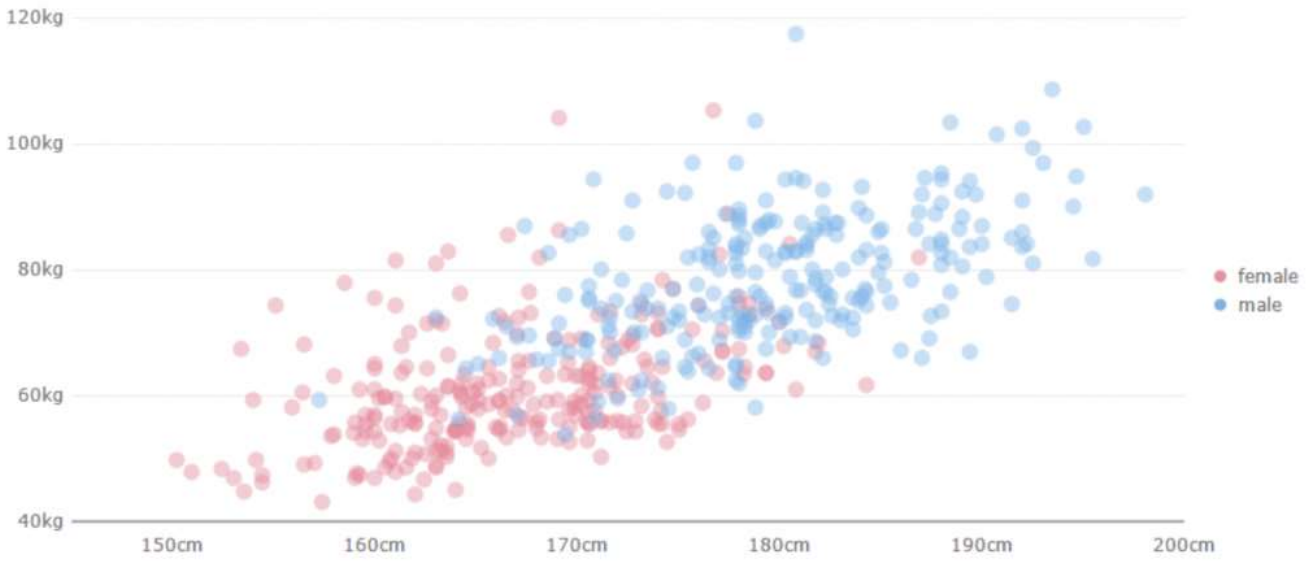
```python
#箱型圖
df = pd.DataFrame(np.random.rand(10, 2), columns=["A","B"])
print(df)
df.boxplot()
```

```
          A         B
0  0.772122  0.297662
1  0.150642  0.919099
2  0.761859  0.444911
3  0.461002  0.876932
4  0.374707  0.214824
5  0.674743  0.192699
6  0.667188  0.921153
7  0.353126  0.343032
8  0.436113  0.929506
9  0.327474  0.935201
<matplotlib.axes._subplots.AxesSubplot at 0x7f7870db3160>
```



散佈圖

- 適用：呈現相關數值間的關係
- 實際應用：性別與體重的關係
- 由散佈圖可看出，大多數的人，身高與體重成正比關係，而男性的身高體重趨勢，都高於女性

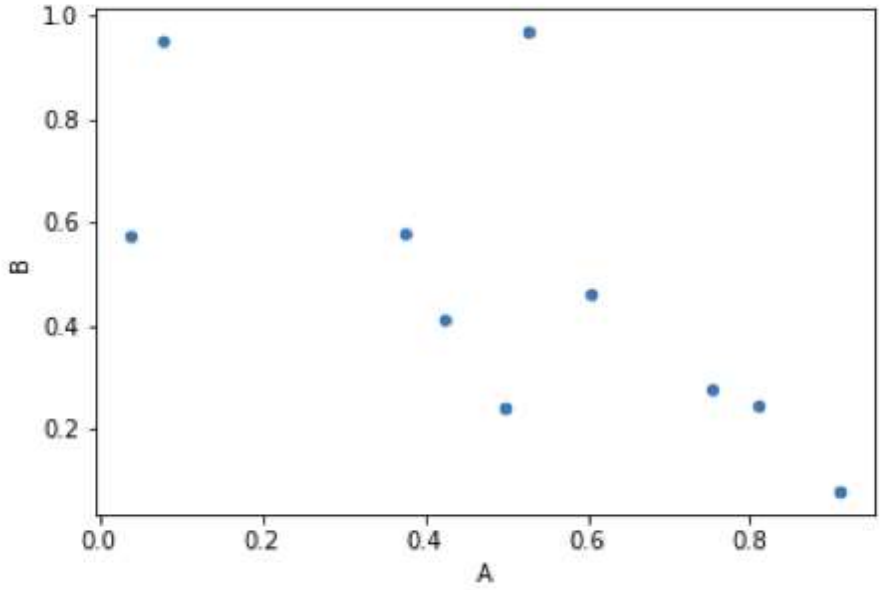


散佈圖 in panda plot

散佈圖，將資料欄位以 x 軸為 A，y 軸為 B 畫出散佈圖

```
#散佈圖
df = pd.DataFrame(np.random.rand(10, 2), columns=["A","B"])
print(df)
df.plot.scatter(x='A', y='B')
```

```
          A         B
0  0.810470  0.246334
1  0.752608  0.276965
2  0.424421  0.409977
3  0.909217  0.079332
4  0.376237  0.578638
5  0.603542  0.460691
6  0.039021  0.572284
7  0.499532  0.239554
8  0.526853  0.967803
9  0.077156  0.949759
<matplotlib.axes._subplots.AxesSubplot at 0x7f78706bff60>
```



知識點回顧

折線圖

- 適用：會隨時間變動的值
- 函數：.plot()

長條圖

- 適用：不同種類資料，在不同時間點的變化
- 函數：.plot.bar()

箱型圖

- 適用：完整呈現數值分布的統計圖表
- 函數：.boxplot()

散佈圖

- 適用：呈現相關數值間的關係
- 函數：.plot.scatter()

參考資料

手把手教你用 Pandas 生成可視化圖表：

[Python數據分析：手把手教你用Pandas生成可視化圖表]

一、線型圖

對於pandas的內置數據類型，Series 和 DataFrame 都有一個用於生成各類 圖表 的 plot 方法。 默認情況下， 它們所生成的是線型圖。其實Series和DataFrame上的這個功能只是使用matplotlib庫的plot()方法的簡單包裝實現。

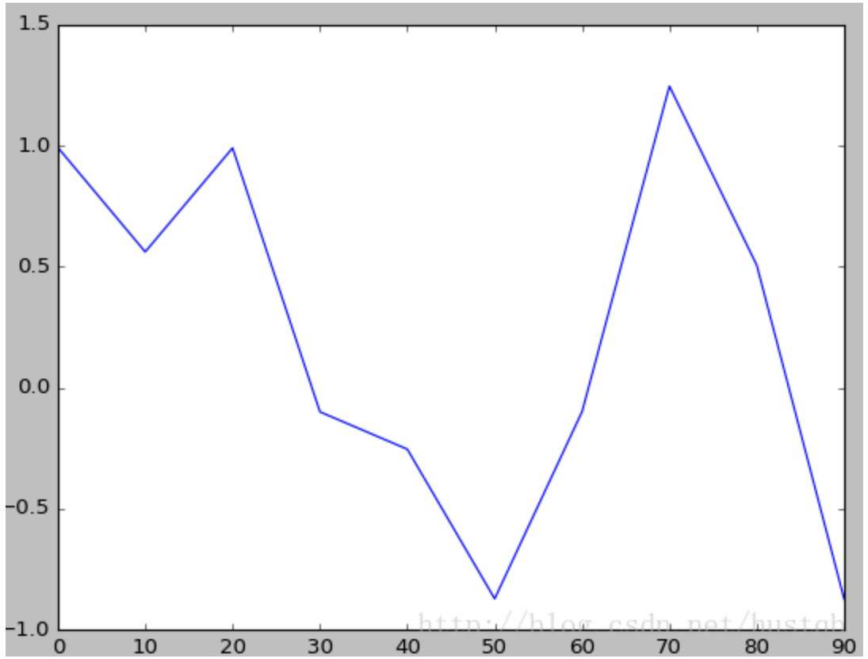
參考以下示例代碼 ：

```
import pandas as pd
import numpy as np

df =
pd.DataFrame(np.random.randn(10,4),index=pd.date_range('2018
/12/18',
   periods=10), columns=list('ABCD'))

df.plot()
```

[pandas 中的繪圖函數]

線型圖

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
s = pd.Series(np.random.randn(10).cumsum(), index=np.arange(0, 100, 10))
s.plot()
plt.show()
```



延伸閱讀

從 pandas 開始 Python 與資料科學之旅

網站：[medium]

數據交點 進擊的資料科學 如何成為資料分析師 輕鬆學習 PYTHON 輕鬆學習 R 語言 PYTHON 與網頁資料擷取 | 數據交點網站

從 pandas 開始 Python 與資料科學之旅

仿效 Tidyverse 的學習模式

Yao-Jen Kuo Follow

Dec 12, 2017 · 21 min read

我們介紹過從 Tidyverse 中的 dplyr 與 ggplot2 套件開始學習 R 語言而非傳統 Base R First 的方式；這樣的學習模式假使套用在 Python 中的話，不從變數類型、資料結構、流程控制、迴圈、自訂函數與類別談起，又該如何開始學習 Python 與資料科學？

Python 基礎資料視覺化—Matplotlib

網站：[medium]

Python 基礎資料視覺化— Matplotlib

對資料進行視覺化，讓你更加了解它

 Yu-Hsuan Chou Nov 20, 2019 · 20 min read

為什麼要做資料視覺化？

> *有效的視覺化可以幫助用戶分析和推理資料和證據。它使複雜的資料更容易理解、理解和使用。—— Wikipedia*

現在的資訊傳達大多講求淺顯易懂以及良好的圖像化，而資料視覺化的目的就在於讓現在這些大量的數據資料能夠更好地被梳理，得以表現出資料的規律、趨勢、分布等等現象，然後再據此做更好的討論以及策略研擬。

如何選擇適當的視覺效果？

我們經常看到各式各樣的視覺化效果，像是基本的長條圖、折線圖、圓餅圖等等，可以依據需求來選擇。舉例來說，如果需要做趨勢的比較，大多會選擇折線圖、如果要做分布的呈現，則是可能會選擇密度圖或是直方

下一步：閱讀範例與完成作業