結合 Pandas 與 Matploglib 進行進階資料視覺化練習

簡報閱讀 範例與作業 問題討論 學習心得(完成)

重要知識點

- 了解如何使用Pandas 處理資料集，並加
  視覺化效果

資料集輸入與處理

先行導入相關的套件

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib as mpl
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

瞭解有關資料集屬性

- 我們可以使用 info()或是 descript() 方法瞭解有關資料集屬性的更多資訊。特別是行和列的數量、列名稱、它們的數據類型和空值數。

資料集的處理

- 有時候無法從資料集明確的看出資料的屬性與因子的相互關係，要針對資料做處理

瞭解數據集

要瞭解數據集的統計摘要，即記錄數、平均值、標準差、最小值和最大值，我們使用 describe()

- df. describe()

可以使用 info() 方法瞭解有關資料集屬性的更多資訊。

- df.info()

處理缺失值

- df = pd.get_dummies

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6497 entries, 0 to 4897
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed_acidity         6497 non-null   float64
 1   volatile_acidity      6497 non-null   float64
 2   citric_acid           6497 non-null   float64
 3   residual_sugar        6497 non-null   float64
 4   chlorides             6497 non-null   float64
 5   free_sulfur_dioxide   6497 non-null   float64
 6   total_sulfur_dioxide  6497 non-null   float64
 7   density               6497 non-null   float64
 8   pH                    6497 non-null   float64
 9   sulphates             6497 non-null   float64
 10  alcohol               6497 non-null   float64
 11  quality               6497 non-null   int64  
 12  color                 6497 non-null   object 
dtypes: float64(11), int64(1), object(1)
memory usage: 710.6+ KB
```

- 數據分析的一個固有部分，看到數據集的資料，十分複雜凌亂，數值大小不一，難以明確識別

	fixed_acidity	volatile_acidity	citric_acid	residual_sugar	chlorides	free_sulfur_dioxide	total_sulfur_dioxide	density	pH	sulphates
count	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000
mean	7.215307	0.339666	0.318633	5.443235	0.056034	30.525319	115.744574	0.994697	3.218501	0.531268
std	1.296434	0.164636	0.145318	4.757804	0.035034	17.749400	56.521855	0.002999	0.160787	0.148806
min	3.800000	0.080000	0.000000	0.600000	0.009000	1.000000	6.000000	0.987110	2.720000	0.220000
25%	6.400000	0.230000	0.250000	1.800000	0.038000	17.000000	77.000000	0.992340	3.110000	0.430000
50%	7.000000	0.290000	0.310000	3.000000	0.047000	29.000000	118.000000	0.994890	3.210000	0.510000
75%	7.700000	0.400000	0.390000	8.100000	0.065000	41.000000	156.000000	0.996990	3.320000	0.600000
max	15.900000	1.580000	1.660000	65.800000	0.611000	289.000000	440.000000	1.038980	4.010000	2.000000

- 所以, 我們將使用 hist 函數通過將所有屬性繪製在一起使操作變得簡單。

直方圖

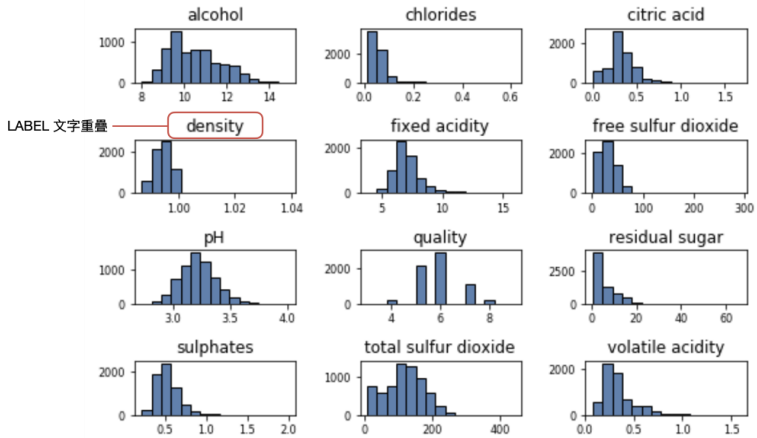
- 直方圖：在垂直軸上計數，在水平軸上使用值範圍。hist 函數通過將所有屬性繪製在一起使操作變得簡單。

對於 label 的驗證最有利的特徵

- 可以搭配 df.head() 對比資料集特徵

#劃出直方圖, 設定10組, bin=10 (直條的數目)
df_all.hist(bins=10,
color='lightblue',edgecolor='blue',linewidth=1.0
,xlabelsize=8, ylabelsize=8, grid=False)

#edgecolor: 條柱的邊線顏色
#linewidth: 線的寬度
#xlabel/ylabel: x/y 軸字體大小
#grid:背景是否有網格



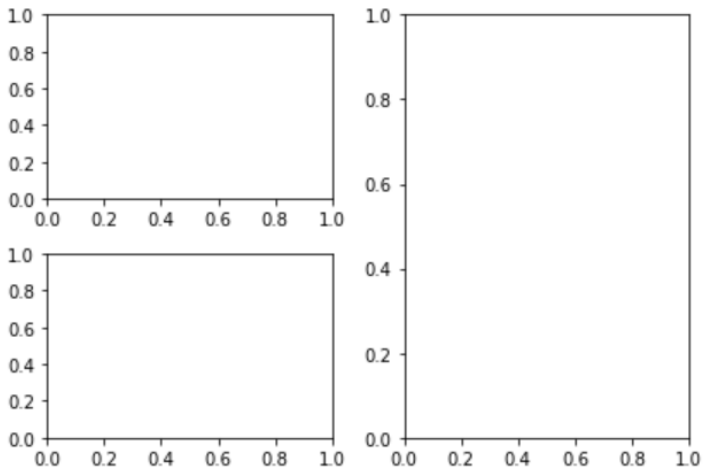
直方圖：tight_layout

避免多個圖重疊，使用 tight_layout 分開，可以節省新增 Figure 的軸的動作
plt.tight_layout()

範例：
#建立一個Figure（空的顯示區）
fig = plt.figure()

```
ax2 = plt.subplot(223)
ax3 = plt.subplot(122)
plt.tight_layout()
```

說明：
.tight_layout 提供 rect 參數，表示一個外界的框框，默認是(0, 0, 1, 1)
(x1, y1, x2, y2)
(x1, y1)矩形限制框左下角點
(x2, y2)矩形限制框右上角點

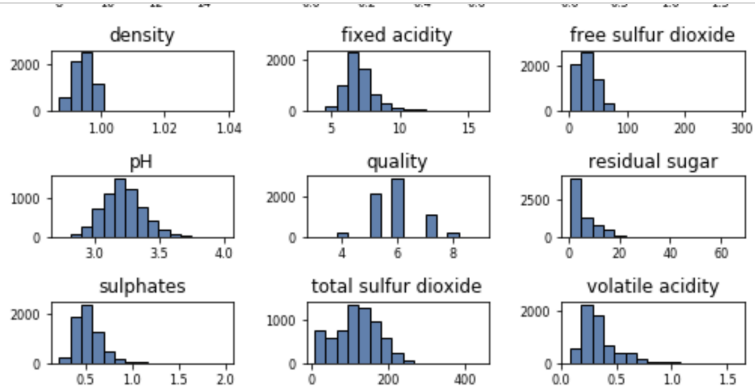


```
#劃出直方圖，設定10組，bin=10 (直條的數目)
df_all.hist(bins=10,
color='lightblue',edgecolor='blue',linewidth=1.0
,xlabelsize=8, ylabelsize=8, grid=False)

#edgecolor：條柱的邊線顏色
#linewidth：線的寬度
#xlabel/ylabel：x/y 軸字體大小
#grid：背景是否有網格

plt.tight_layout(rect=(0, 0, 1.2, 1.2))
```

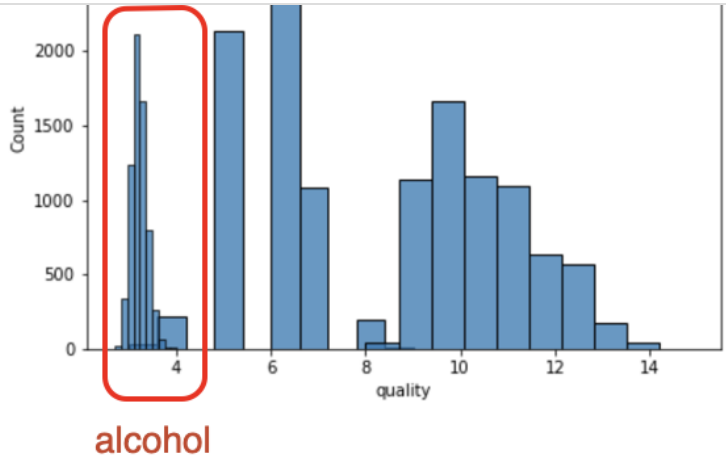
修正後圖片的顯示：



資料觀察

從上圖來看，我們可以發現，數據多的幾個特徵：alcohol, pH, total_sulfur_dioxide, fixedacidity, citric_acid, volatile_acidity
可以挑選幾個特徵來看彼此的關係

```
plt.figure(figsize=(7,5))
sns.histplot(df_all["quality"], bins=10, label="quality")
sns.histplot(df_all["pH"], bins=10, label="pH")
sns.histplot(df_all["alcohol"], bins=10, label = "alcohol")
sns.histplot(df_all["total_sulfur_dioxide"], bins=10, label = "total_sulfur_dioxide")
```



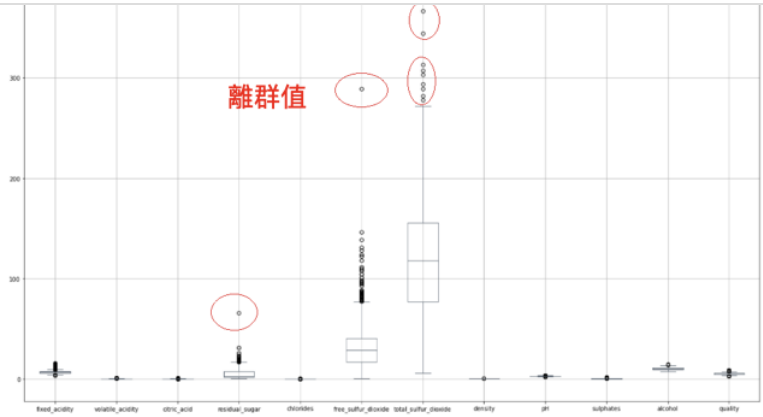
上圖的解讀：

- 比對資料整理前，Quality 的分布，我們可以發現 pH 跟 Quality 呈現高度正相關
- alcohol 對 Quality 的影響被侷促

觀察離群值

- 箱形圖（Box plot），又稱為盒鬚圖、盒式圖、盒狀圖或箱線圖，是一種用作顯示一組數據分散情況資料的統計圖
- 用來觀察每個特徵的離群值

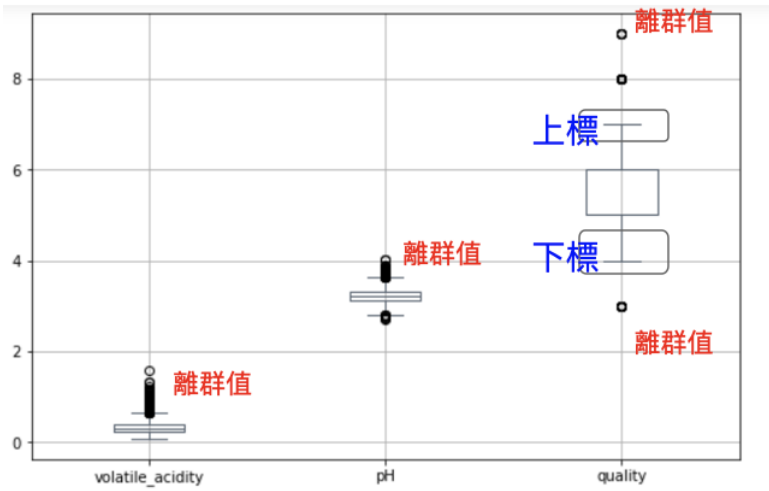
```
df_all.boxplot(color='#556270')
plt.tight_layout(rect=(0, 0, 3, 3))
```



觀察離群值 - 針對特定特徵

df_all.boxplot(column=['volatile_acidity','pH','quality'], color='#556270')

#plt.tight_layout(rect=(0, 0, 3, 3))



熱力圖

- 熱力圖是一個以顏色變化來顯示數據的矩陣。簡單來說，就是用依據數字的不同，使用不同的顏色來呈現數據
- 利用熱力圖可以看資料表裡多個特徵兩兩的相似度。

vmax=None,cmap=None, center=None, robust=False, annot=None, fmt='.2g', annot_kws=None,linewidths=0, linecolor='white', cbar=True, cbar_kws=None, cbar_ax=None,square=False, xticklabels='auto', yticklabels='auto', mask=None, ax=None,**kwargs)

- 熱力圖輸入資料引數：
  - data：矩陣資料集，可以是numpy的陣列（array），也可以是 pandas 的 DataFrame。
- 熱力圖矩陣塊顏色引數：
  - vmax，vmin：分別是熱力圖的顏色取值最大和最小範圍，預設是根據data資料
  - cmap：取值是 matplotlib 包裡的 colormap 名稱或顏色物件，
  - center：設定熱力圖的色彩中心對齊值；通過設定 center 值，可以調整生成的影象顏色的整體深淺；

熱力圖矩陣塊註釋引數：

- annot(annotate的縮寫)：預設取值 False；如果是 True，在熱力圖每個方格寫入資料；如果是矩陣，在熱力圖每個方格寫入該矩陣對應位置資料
- fmt：矩陣上標識數字的資料格式，比如保留小數點後幾位數字
- annot_kws：預設取值 False；如果是 True，設定熱力圖矩陣上數字的大小顏色字型

- linewidths：定義熱力圖裡“表示兩兩特徵關係的矩陣小塊”之間的間隔大小
- linecolor：切分熱力圖上每個矩陣小塊的線的顏色，預設值是’white’

熱力圖顏色刻度條引數：

- cbar：是否在熱力圖側邊繪製顏色刻度條，預設值是 True
- cbar_kws：熱力圖側邊繪製顏色刻度條時，相關字型設定，預設值是 None
- cbar_ax：熱力圖側邊繪製顏色刻度條時，刻度條位置設定，預設值是 None

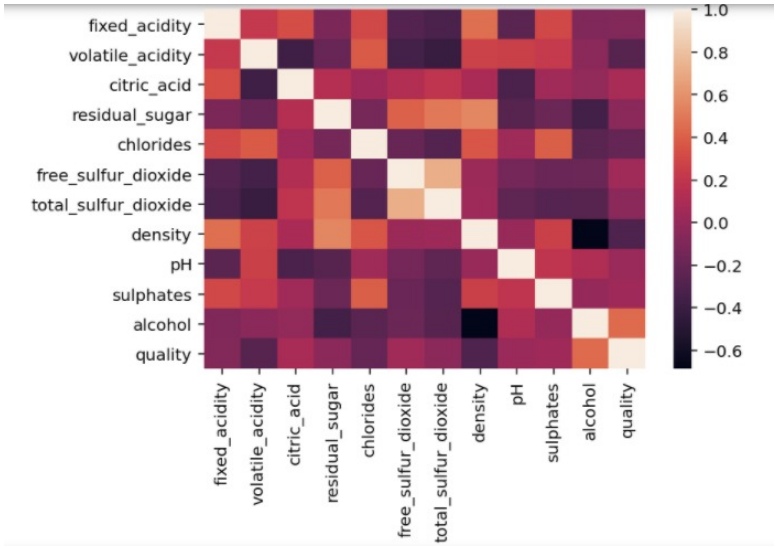
其他：

- square：設定熱力圖矩陣小塊形狀，預設值是 False
- xticklabels，yticklabels：xticklabels 控制每列標籤名的輸出；yticklabels 控制每行標籤名的輸出。預設值是 auto。
- mask：控制某個矩陣是否顯示出來。預設值是 None。
- ax：設定作圖的座標軸，一般畫多個子圖時需要修改不同的子圖的該值。

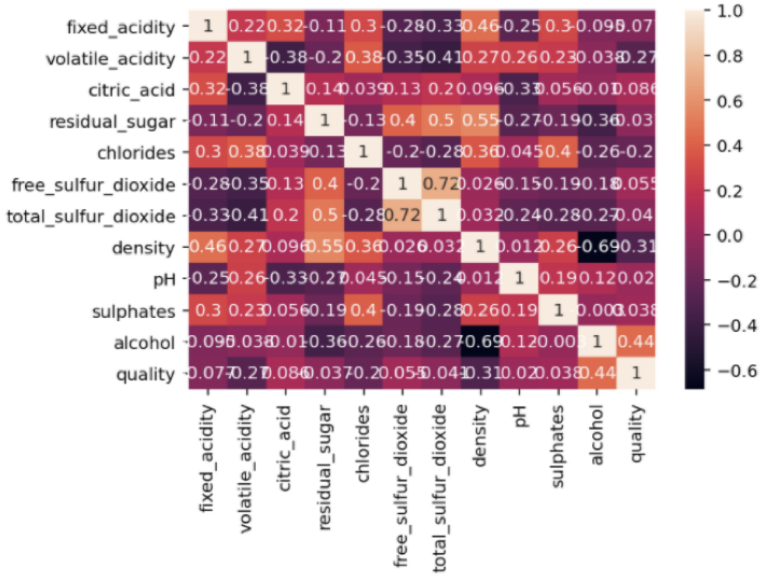
實際應用

```
# 最直接的做法是把資料集丟給heatmap函數
sns.heatmap(df_all.corr())

# 可以看到熱力圖主要展示的是二維數據的數據關係
# 不同大小的值對應不同的顏色深淺
```



# 加上標記data
sns.heatmap(df_all.corr(), annot=True)



知識點回顧

PYTHON 數據處理

Pandas - Python 讀取 csv 檔、excel 檔及文字檔 txt 的工具套件。

（rug）用來觀測分布，同時也可以使用 fit 參數去擬合分配圖形。

我們可以使用 Matplotlib, Seaborn, Pandas 處理龐大的數據集

- 長條圖（Bar plot)：長條圖也可稱為柱狀圖，通常用在數值的顯示或者比較
- 直方圖(Hist plot)：用於頻率分佈，y 軸表示頻率分佈（數值或者比率），hist 函數柱體個數預設 bins=10，且預設圖中會有網格線。
- 散佈圖能夠顯示 2 個維度上每組數據的值。可以顯示觀察數據分布情形，描述數據的相關性
- 熱力圖(heatmap)是一個以顏色變化來顯示數據的矩陣。簡單來說，就是用依據數字的不同，使用不同的顏色來呈現數據。

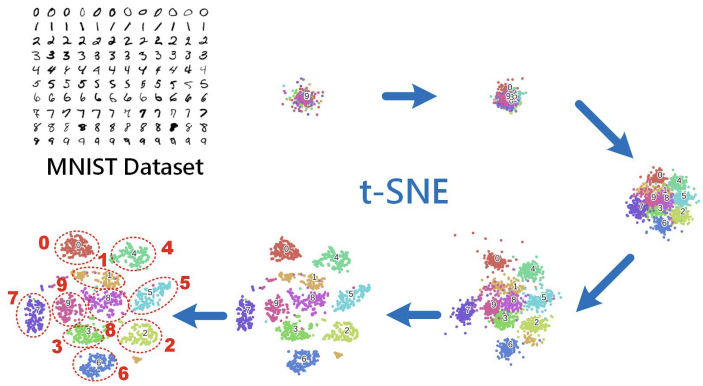
延伸閱讀[SEP]

資料降維與視覺化

目的：
為了讓人們容易理解高維資料的分佈情況及降低後續特徵提取演算量，最常用的方式就是將資料「**降維(Dimensionality Reduction)**」到二維或三維空間再進行觀察，亦可看做是將資料從高維度重新投影(Projection)至低維度空間

作用：
易於觀察資料集的內容，尤其在經過降維之後有沒有更好，以手寫辨識資料庫為例

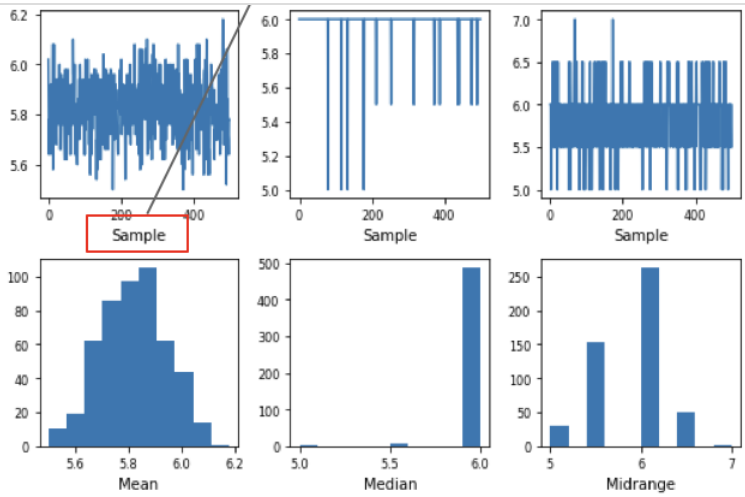
[HUB專欄】如何應用高維資料可視化一眼看穿你的資料集]



中位數與平均值

#這個資料集包括了平均數，標準差，一定區間的數值
當樣本母數差距過大時，我們可以考慮使用 bootstrap 幫忙繪製圖表包含 mean，median and mid-range statistics

```
s = pd.Series(df_all['quality'])
pd.plotting.bootstrap_plot(s, samples=500)
plt.tight_layout(rect=(0, 0, 1.2, 1.2))
```

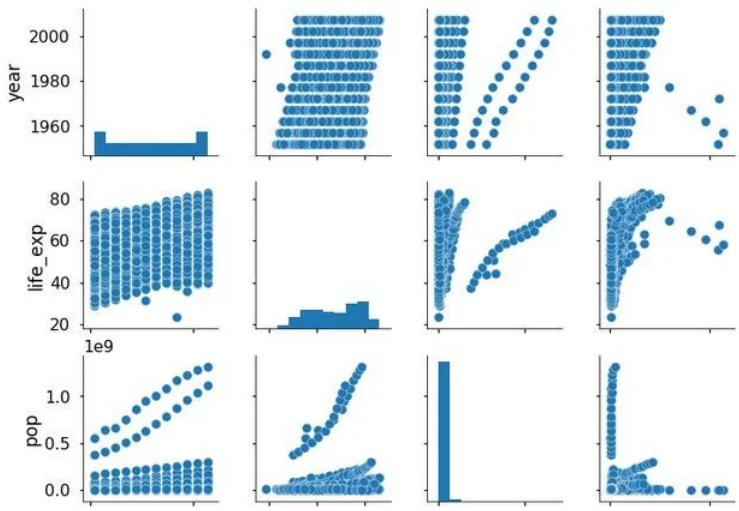


Python 如何快速創建強大的探索性數據分析

網站：[Python如何快速創建強大的探索性數據分析可視化]

如何創建默認配對圖以快速檢查我們的數據，以及如何自定義視覺化以獲取更深入的洞察力

- 針對 sns.pairplot() 有深入的分析
- 針對使用 PairGrid 進行自定義
- 內容包含 散點圖，內核密度，箱型圖 ..... 等等
- 利用配色與關鍵字，進一步分析 data



下一步：閱讀範例與完成作業