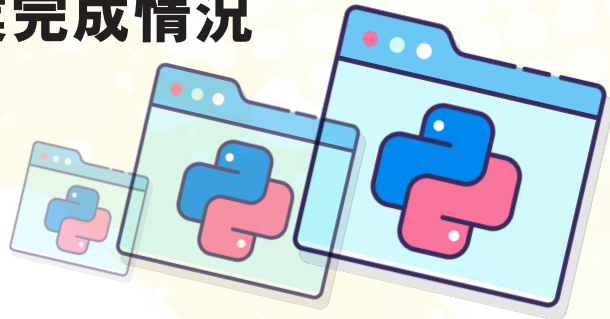




# 使用Python實現自動檢查及跟進網上作業 ——自動分析作業完成情況

文·圖 | 蘇華傑



## 前言

在教師日常工作中，如果能夠用編程知識將重複的工作實現自動化，那麼工作將會變得高效，也能減低自身的工作壓力。在《使用Python實現自動檢查及跟進網上作業——自動從eClass下載登分表》中，筆者展示了如何透過Python自動化操作網頁下載eClass登分表，接下來將繼續講解如何使用Python整理未完成作業的學生名單，分析作業情況，並通過數據可視化的方式呈現出來。

## 一、Pandas

Pandas是一種可提供數據分析功能的Python開源軟件庫，它具備靈活且高效地處理數據的數據結構，能支持對各種不同格式的數據進行讀與寫操作、數據子集化、數據轉換、數據清洗、數據聚合與分組運算等功能（增田秀人，2021）。在本文中，筆者使用Pandas的DataFrame進行數據處理與分析，使用M2晶片的Macbook Air運行全部程序僅用時1.2秒，與其他自動化操作Excel的開源庫（如xlwings、openpyxl）相比，Pandas的整體效能明顯更勝一籌。

## 二、數據預處理

我們可以觀察已經下載的Excel檔案資料（見圖1），對數據進行預處理以便後續的分析工作。首先，從圖1可以看到原始數據檔案是以“--”來表示作業未完成，因此在使用Pandas讀取資料時，可以將“--”設為遺漏值；然後，第1行只是任務的標題，與第2行資料重疊，用處不大，可以設定header=1（即從第2行開始讀取）；接着，可以看到每名學生對應的班別與學號存放在同一列，這不利於分組分析，因此可以使用str.split的expand功能將其分拆；最後，對於不需要的列，可以使用drop指令予以刪除。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1		不讀取此行	23數學節節練	24數學節節練	25	26	27	28	29	30	31	32	33	34數學節節練	
2	學號	學生	課後:節節練23	課後:節節練24	課後:節節練25	課後:節節練26	課後:節節練27	課後:節節練28	課後:節節練29	課後:節節練30	課後:節節練31	課後:節節練32	課後:節節練33	課後:節節練34	總分
3	SC1A-1	何	10		7	7	8	9	9	6	9	9			65
4	SC1A-2	何	10		10	9	7	9	9	9	6	8	7	7	96
5	SC1A-3	何	10		9	9	#	#	9	9	6	7	8	9	105
6	SC1A-4	何	10		10	8	9	#	#	9	9	6	8	8	104
7	SC1A-5	何	10		9	9	9	#	#	9	6	7	8	9	96
8	SC1A-6	吳	10		10	3	8	9	9						75,67
9	SC1A-7	李	10		10	9	1	#	#	9	6	8	8		89,5
10	SC1A-8	李	10		10	1	2	3	6	8	1	3	2	3	58,83
11	SC1A-9	李	10		10	3	2	9	7	8	6	3	8		63
12	SC1A-10	林	10		10	4	2	9	9	6	8	3	4		81
13	SC1A-11	林	10		10	9	9	#	#	9	9	6	8	8	98
14	SC1A-12	洪	10		10	9	8	#	#	9	9	6	5	2	85
15	SC1A-13	張	10		10	3	3	9	7	8	6	7	9	6	88

圖 1：原始數據資料

具體的程式碼如下（# 號後面的內容是對程式碼的註釋）：

```
import pandas as pd

filename=f'/Users/keithsou/Downloads/0_置頂 2122 第三段 節節練.xlsx' #檔案路徑
def grouping(classname): #建立按 “班別” 分組的函式
    df=pd.read_excel(filename,header=1,na_values='--') #以 DataFrame 讀取檔案,並設定遺漏值
    df2=df['學號'].str.split('-',expand=True) #將 “班別-學號” 這一系列拆分為兩列
    df2.columns=['班別','學號'] #為新拆分兩列之標籤命名
    df=df.drop(columns=['總分','學號']) #刪除不需要的列
    df=pd.concat([df2,df],axis=1) #將 df 與 df2 合併
    df.set_index(['班別','學號','學生'], inplace = True) #設定索引的名稱
    group=df.groupby('班別') #建立按 “班別” 分組的 DataFrame
    group_class=group.get_group(classname) #選取某個班別的 DataFrame
    return group_class #返回已選取班別的 DataFrame
```

在建立函式後，不妨輸入 `display (group_class ( 'SC1A' ))` 檢視函式的實際效果，輸出結果是 SC1A 班已完成預處理的數據（見圖 2）。

班別	學號	學生	課後:節節練 23	課後:節節練 24	課後:節節練 25	課後:節節練 26	課後:節節練 27	課後:節節練 28	課後:節節練 29	課後:節節練 30	課後:節節練 31	課後:節節練 32	課後:節節練 33	課後:節節練 34
SC1A	1	何	10.0	7.0	7.0	8.0	NaN	9.0	9.0	6.0	NaN	NaN	9.0	NaN
	2	何	10.0	10.0	9.0	7.0	9.0	9.0	9.0	6.0	8.00	7.0	7.0	5.0
	3	何	10.0	9.0	9.0	10.0	10.0	9.0	9.0	6.0	7.00	8.0	9.0	9.0
	4	何	10.0	10.0	8.0	9.0	10.0	9.0	9.0	6.0	8.00	8.0	10.0	7.0

圖 2：完成數據預處理的 DataFrame

### 三、整理未完成作業的學生名單

對於“未完成作業的學生名單”之佈局，筆者採取按照學生名字分隔的方式進行設計。這樣設計的好處是既可以讓教師快速地查看哪名學生未完成作業，及時對這些學生作出跟進；也能讓學生清楚看到未完成的作業名稱，避免補做時仍然出現做漏的情況。

具體的程式碼如下（# 號後面的內容是對程式碼的註釋）：

```
def missing_homework(classname): #建立查找指定班別未完成作業學生名單的函式
    missing=grouping(classname)[grouping(classname).isnull().T.any()] #選取指定班別的遺漏值
    message="" #建立一個名為“message”的字符串，用以儲存全部訊息
    msg_title1=f'{taskname}:\n' # 標題 1
    msg_title2=f'{classname}未完成 Powerlesson2 作業名單：' # 標題 2
    msg_title3=f'\n-----\n' # 標題 3
    msg_content="" #用以儲存欠交訊息內容的字符串
    for index,row in missing.iterrows(): #遍歷含遺漏值 DataFrame 的每一行
        msg_content+=f'{index[1][1:]}號 {index[2]} \n' # msg_content 加上：8 號 陳大明
        for i in row[row.isnull()].index: #遍歷陳大明的每一筆資料
            msg_content+=f'未完成: {i[4:]} \n' #寫出 未完成任務的名稱
            msg_content+=f'未完成總數目: {row.isnull().sum()} \n' #寫出未完成任務的總數
            msg_content+=f'-----\n' #增加一行虛線

    if msg_content!="":
        message+=f'{msg_title1}{msg_title2}{msg_title3}{msg_content}'
    else:
        message+=f'{msg_title1}{classname}已完成作業{msg_title3}'
    return message #返回 message
```

在建立函式後，不妨輸入 print (missing\_homework ('SC1A')) 檢視函式的實際效果，可以從輸出結果看到 SC1A班未完成作業的學生名單（見圖3）。

```
0_置頂 2122 第三段 節節練：
SC1A 未完成Powerlesson2作業名單：
-----
1號 何曉彤
未完成：節節練27
未完成：節節練31
未完成：節節練32
未完成：節節練34
未完成總數目：4
-----
5號 何曉彤
未完成：節節練34
未完成總數目：1
-----
6號 吳曉彤
未完成：節節練34
未完成總數目：1
-----
```

圖 3：未完成作業的學生名單

#### 四、分析作業的表現

由於在第二部分已經使用Pandas的DataFrame對數據進行預處理，包括去除不需要的資料和對數據進行分組，因此接下來的分析過程會相對簡便，使用agg函數從不同維度進行聚合操作，可獲得描述性統計資料。對於二維數據，當axis=0時，表示沿着垂直方向進行分析，即按任務名稱劃分；當axis=1時，表示沿着水平方向進行分析，即按學生名稱劃分；而round(1)則表示保留小數點後一位。

具體的程式碼如下（#號後面的內容是對程式碼的註釋）：

```
def descriptive_statistics(classname):
    #取得指定班別整體在各項作業表現的平均值、中位數、最值、標準差等描述性統計資料
    result_task=grouping(classname).agg(['mean','median','max','min','std'],axis=0).round(1).T

    #取得指定班別每名學生作業的完成數、平均值、中位數、最值等描述性統計資料
    result_students=grouping(classname).agg(['count','mean','median','max','min'],axis=1).round(1)

    #展示結果
    display(result_task,result_students)

    #返回結果
    return(result_task,result_students)
```

在建立函式後，不妨輸入descriptive\_statistics（‘SC1B’）檢視實際效果，可以從輸出結果看到SC1B班的作業表現。其中，圖4是result\_task的結果，展示了每項任務的表現；圖5是result\_students的結果，展示了每名學生的作業表現。

	mean	median	max	min	std
課後:節節練23	9.2	10.0	10.0	2.0	1.7
課後:節節練24	8.0	9.0	10.0	0.0	2.6
課後:節節練25	8.2	9.0	10.0	3.0	1.9
課後:節節練26	7.3	8.0	10.0	1.0	2.0
課後:節節練27	8.4	9.0	10.0	1.0	2.2
課後:節節練28	7.3	8.0	10.0	3.0	2.0
課後:節節練29	7.7	8.0	10.0	2.0	2.0
課後:節節練30	3.9	4.0	6.0	0.0	1.5
課後:節節練31	5.2	5.3	7.2	2.0	1.3
課後:節節練32	8.1	8.0	10.0	5.0	1.4
課後:節節練33	9.0	10.0	10.0	3.0	1.7
課後:節節練34	5.5	6.5	9.0	1.0	2.6

圖 4：每項任務的表現

			count	mean	median	max	min
SC1B	1	王	12.0	6.1	6.5	10.0	1.0
	2	何	12.0	7.9	8.0	10.0	5.3
	3	何	12.0	4.5	4.0	8.0	1.0
	4	吳	12.0	5.6	4.5	9.0	2.0
	5	李	12.0	8.9	9.0	10.0	6.0
	6	林	12.0	8.3	8.5	10.0	4.0
	7	林	9.0	8.1	9.0	10.0	6.0
	8	施	12.0	8.3	8.5	10.0	4.0
	9	洪	12.0	3.4	3.0	7.0	0.0
	10	張	12.0	8.3	8.5	10.0	4.0
	11	梁	12.0	4.3	4.5	7.0	1.0
	12	梁	12.0	8.2	8.5	10.0	5.0

圖 5：每名學生的表現

Pandas支援將DataFrame直接寫入Excel檔案，讀者如果有需要將上述結果儲存為Excel檔案以方便後續計分，只需在函式中加入下列代碼：

```
result_students.to_excel(f'/Users/keithsou/Downloads/{classname}.xlsx')
```

## 五、對所選班別進行分析

假如，讀者需要分析的班別多於一個，可以建立列表來存放各個班別的名稱，然後使用For迴圈來編歷列表即可輕鬆實現。

```
classname=['SC1A','SC1B','SC1C','SC1D'] #輸入所選的班別
for classname in classname: #編歷列表, 執行下列函式
    missing_homework(classname) #整理未完成作業學生名單
    descriptive_statistics(classname) #分析作業表現
```

## 六、數據可視化

最後，讀者也可以透過Python第三方庫Matplotlib將分析結果以數據可視化的方式呈現出來（見圖6），詳情可參閱Matplotlib的官方文檔。

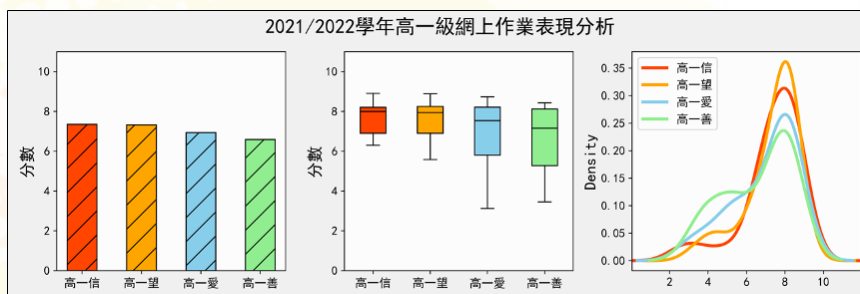


圖 6：使用 Matplotlib 實現數據可視化

## 結語

目前為止，筆者已介紹如何透過Python從eClass下載網上作業的登分表，然後整理未完成作業的學生名單，分析網上作業的表現，並以數據可視化的方式呈現出來。後續的文章將介紹如何設定在特定時間點自動運程序，並自動發送分析報告，真正地實現“自動化”。📖

## 參考文獻

陳歡（譯）（2021）。Pandas數據預處理詳解（原作者：增田秀人）。北京：中國水利水電出版社。（原著出版年：2020）

■ 蘇華傑  
■ 澳門浸信中學數學科教師