

合併表格化的文字檔

- 利用別人已經發展成熟的程式庫來實現檔案合併
 - 參考 Introduction to Python for Science and Engineering by David J. Pine (Chapter_10 Data Manipulation and Analysis: Pandas)
 - 只要先將各式檔案(txt, csv, json, xlsx,...)轉成Pandas的資料結構，後續對資料的處理行為都會一致
 - 合併表格化的文字檔只是Pandas模組的一項「小」功能
- 和主程式分開，另外使用yaml格式的檔案來簡化檔案路徑的存取
 - 參考 Network Programmability and Automation by Jason Edelman, Scott S. Lowe & Matt Oswalt (ch_5 Data Formats and Data Models)
 - 將檔案的路徑存放在另一個檔案內，yaml格式可以方便轉成python容易處理的資料結構
- 測試：
 - 中央氣象局海象監測資料(劉奕的版本)：txt檔案的讀和寫
 - 合併04/09 ~ 05/06與04/23 ~ 05/21兩項不同日期範圍的資料，最後產生04/09 ~ /05/21新資料
 - 港灣環境資訊網的定點歷線圖：json檔案的讀和寫。xlsx檔案的寫。
 - 使用Pandas前：114行程式碼。使用Pandas後：50行。使用Pandas模組後的新版程式減少64行，最後程式為原先總行數的44%。

合併兩個 txt 檔

頭

date s)	time	waterlevel(m)	Hs(m)	wave_dir	Ts(s)	wind(m/s)	wind_rank	wind_dir	gust(m/s)					
0410	0000	-0.38	nan	nan	4.4	3	NNE	6.5	4	20.0	18.2	nan	nan	nan
0410	0100	-0.40	nan	nan	3.3	2	NW	5.1	3	19.9	17.7	nan	nan	nan
0410	0200	-0.31	nan	nan	3.2	2	N	5.3	3	19.9	17.5	nan	nan	nan
...														
0506	2100	-0.16	nan	nan	1.5	1	SW	2.6	2	23.4	23.0	nan	nan	nan
0506	2200	-0.25	nan	nan	1.5	1	SSW	2.5	2	23.4	22.8	nan	nan	nan
0506	2300	-0.26	nan	nan	0.9	1	SW	1.6	2	23.4	22.5	nan	nan	nan

Input 1

+

尾

date s)	time	waterlevel(m)	Hs(m)	wave_dir	Ts(s)	wind(m/s)	wind_rank	wind_dir	gust(m/s)					
0424	0000	-0.29	nan	nan	2.5	2	NNW	4.2	3	21.1	22.9	nan	nan	nan
0424	0100	-0.14	nan	nan	3.3	2	N	6.1	4	21.1	22.8	nan	nan	nan
0424	0200	0.03	nan	nan	2.7	2	NW	4.5	3	21.1	22.8	nan	nan	nan
...														
0521	2100	-0.18	nan	nan	1.7	2	SW	2.5	2	28.0	28.0	nan	nan	nan
0521	2200	-0.20	nan	nan	1.3	1	WSW	2.8	2	28.0	27.8	nan	nan	nan
0521	2300	-0.09	nan	nan	0.6	1	WSW	1.2	1	27.9	27.7	nan	nan	nan

Input 2

=

頭

date s)	time	waterlevel(m)	Hs(m)	wave_dir	Ts(s)	wind(m/s)	wind_rank	wind_dir	gust(m/s)					
0410	0000	-0.38	nan	nan	4.4	3	NNE	6.5	4	20.0	18.2	nan	nan	nan
0410	0100	-0.40	nan	nan	3.3	2	NW	5.1	3	19.9	17.7	nan	nan	nan
0410	0200	-0.31	nan	nan	3.2	2	N	5.3	3	19.9	17.5	nan	nan	nan
...														
0521	2100	-0.18	nan	nan	1.7	2	SW	2.5	2	28.0	28.0	nan	nan	nan
0521	2200	-0.20	nan	nan	1.3	1	WSW	2.8	2	28.0	27.8	nan	nan	nan
0521	2300	-0.09	nan	nan	0.6	1	WSW	1.2	1	27.9	27.7	nan	nan	nan

output

尾

簡潔的 txt 合併程式

```
1 import yaml
2 import glob
3 import os
4
5 import pandas as pd
6
7
8 with open('path.yml', 'rt') as f_r:
9     data = yaml.load(f_r, Loader=yaml.FullLoader)
10    FOLDER_TXT = data['new']
11    FOLDER_MERGE = data['old']
12
13 # merge txt files and discard additional duplicates
14
15 files_path = FOLDER_TXT + '*.txt'
16 filenames_new = glob.glob(files_path)
17
18 for filename_new in filenames_new:
19
20     filename_no_path = os.path.basename(filename_new)
21     filename_merge = f'{FOLDER_MERGE}{filename_no_path}'
22
23     df_new = pd.read_table(filename_new, sep='\t', dtype=str)
24
25     try:
26         df_old = pd.read_table(filename_merge, sep='\t', dtype=str)
27     except:
28         df_old = None
29
30     full_df = pd.concat([df_old, df_new])
31     unique_df = full_df.drop_duplicates(subset=['date', 'time'])
32     unique_df.to_csv(filename_merge, index=False, sep='\t', na_rep='nan')
33
34     #print(f'merge {filename_no_path} (complete)')
35 print(f'merge txt files(complete)')
36
```

← 讀取 yaml 檔案
來獲得路徑

收集待處理的 txt 檔案 (最新下載到的)

txt 檔被合併前 (後) 的路徑和檔名

將 txt 檔 (最新下載) 轉成 Pandas 的資料結構

將 txt 檔 (待合併的) 轉成 Pandas 的資料結構

合併檔案並根據 date 和 time 兩個欄位剔除重複的部份

將合併後的資料轉成 txt 檔 (覆蓋原先待合併的檔案)

```
pydoc@pydoc:~/slides/merge_cwb_mmc_txt/code$ tree -L 2
.
├── data
│   ├── merge
│   └── txt
├── merge_cwb_mmc_txt_0.py
└── path.yml
3 directories, 2 files
```

```
path.yml
~/slides/merge_cwb_mmc_txt/code
---
old: './data/merge/' # folder of old, merged files
new: './data/txt/' # folder of new files
```

合併兩個 json 檔

頭

```
[
  {
    "datetime": "2021-03-24 11:00:00",
    "year": "2021",
    "MONTH_": "3",
    "day": "24",
    "HOUR": "11",
    "minute": "0",
    "tpv": "23.9"
  },
  {
    "datetime": "2021-03-24 12:00:00",
    "year": "2021",
    "MONTH_": "3",
    "day": "24",
    "HOUR": "12",
    "minute": "0",
    "tpv": "24"
  },
  {
    "datetime": "2021-03-24 13:00:00",
    "tpv": "26.7"
  },
  ...
  {
    "datetime": "2021-04-24 09:00:00",
    "year": "2021",
    "MONTH_": "4",
    "day": "24",
    "HOUR": "9",
    "minute": "0",
    "tpv": "26.8"
  },
  {
    "datetime": "2021-04-24 10:00:00",
    "year": "2021",
    "MONTH_": "4",
    "day": "24",
    "HOUR": "10",
    "minute": "0",
    "tpv": "26.8"
  }
]
```

Input 1

+

尾

```
[
  {
    "datetime": "2021-04-22 11:00:00",
    "year": "2021",
    "MONTH_": "4",
    "day": "22",
    "HOUR": "11",
    "minute": "0",
    "tpv": "26.5"
  },
  {
    "datetime": "2021-04-22 12:00:00",
    "year": "2021",
    "MONTH_": "4",
    "day": "22",
    "HOUR": "12",
    "minute": "0",
    "tpv": "26.6"
  },
  {
    "datetime": "2021-04-22 13:00:00",
    "tpv": "30.1"
  },
  ...
  {
    "datetime": "2021-05-22 09:00:00",
    "year": "2021",
    "MONTH_": "5",
    "day": "22",
    "HOUR": "9",
    "minute": "0",
    "tpv": "30.2"
  },
  {
    "datetime": "2021-05-22 10:00:00",
    "year": "2021",
    "MONTH_": "5",
    "day": "22",
    "HOUR": "10",
    "minute": "0",
    "tpv": "30.2"
  }
]
```

Input 2

=

頭

```
[
  {
    "datetime": "2021-03-24 11:00:00",
    "year": "2021",
    "MONTH_": "3",
    "day": "24",
    "HOUR": "11",
    "minute": "0",
    "tpv": "23.9"
  },
  {
    "datetime": "2021-03-24 12:00:00",
    "year": "2021",
    "MONTH_": "3",
    "day": "24",
    "HOUR": "12",
    "minute": "0",
    "tpv": "24"
  },
  {
    "datetime": "2021-03-24 13:00:00",
    "tpv": "30.1"
  },
  ...
  {
    "datetime": "2021-05-22 09:00:00",
    "year": "2021",
    "MONTH_": "5",
    "day": "22",
    "HOUR": "9",
    "minute": "0",
    "tpv": "30.2"
  },
  {
    "datetime": "2021-05-22 10:00:00",
    "year": "2021",
    "MONTH_": "5",
    "day": "22",
    "HOUR": "10",
    "minute": "0",
    "tpv": "30.2"
  }
]
```

尾

output

使用 Pandas 前 (定點歷線圖程式)

```
merge_docklands_records.py - /home/pydoc/python3_scripts/merge_docklan...
File Edit Format Run Options Window Help
1 import glob
2 import json
3 import os
4
5 from json_excel_converter import Converter
6 from json_excel_converter.xlsx import Writer
7
8 harbor_names = {
9     "KL": "基隆港",
10    "TP": "臺北港",
11    "TC": "臺中港",
12    "KH": "高雄港",
13    "SA": "蘇澳港",
14    "HL": "花蓮港",
15    "AP": "安平港",
16    "BD": "布袋港",
17    "MT": "馬祖"
18 }
19
20 with open('path_inout.cfg', mode='rt', encoding='utf-8') as f_r:
21     lines = f_r.readlines()
22     #path_in1 = lines[0].strip() # line #1 is used no more
23     path_inputs = lines[1].strip()
24     path_output = lines[2].strip()
25
26 with open('places.cfg', mode='rt', encoding='utf-8') as f_r:
27     locations = json.loads(f_r.read())
28
29
30 def old_merger(path_in1, path_in2, path_out):
31     print('-'*40)
32     #print('Updating records up to', path_in2.split('\\')[2]) #suppose windows
33     print('Updating records up to', path_in2.split('/')[2]) #suppose linux
34     #quit()
35     print('-'*40)
36     places = [location for location in locations if not location.startswith('#')]
37
38     for place in places:
39
40         print('\nProcessing {}'.format(harbor_names[place]))
41
42         #quit()
43
44         var_filenames_in1 = path_in1 + '/records_{' + '/' + '.json'
45         filenames_in1 = glob.glob(var_filenames_in1.format(place))
46
47         for filename_in1 in filenames_in1:
48
49             basename_in1 = os.path.basename(filename_in1) # without file path
50
51             with open(filename_in1, mode='rt', encoding='utf-8') as f_r:
52                 record_in1 = json.loads(f_r.read())
53
54             # collect all json files in in2
55             var_filenames_in2 = path_in2 + '/records_{' + '/' + '.json'
56             filenames_in2 = glob.glob(var_filenames_in2.format(place))
57
```

Ln: 4 Col: 0

```
merge_docklands_records.py - /home/pydoc/python3_scripts/merge_docklan...
File Edit Format Run Options Window Help
50
51 # in in_2, open the file of the the same name in in_1
52 var_filename_in2 = path_in2 + '/records_{' + '/' + basename_in1
53 filename_in2 = var_filename_in2.format(place)
54
55 with open(filename_in2, mode='rt', encoding='utf-8') as f_r:
56     record_in2 = json.loads(f_r.read())
57
58
59 var_folder_out = path_out + '/records_{' + '/'
60 folder_out = var_folder_out.format(place)
61
62 os.makedirs(folder_out, exist_ok=True)
63
64 filename_out = folder_out + basename_in1
65
66 with open(filename_out, mode='wt', encoding='utf-8') as f_w:
67
68     record_out = record_in1.copy() # default record to be modified
69
70 # put all in1 "datetime" substrings in a whole string
71 datetimes_record_in1 = ''.join(i['datetime'] for i in record_in1)
72
73 #search "record_in1" for the 1st not-found item of "record_in2"
74 #by checking "datetime" of each item
75 # mismatch_index : index of the 1st not-found item
76 mismatch_index = 0
77 for item_in2 in record_in2:
78     date_in2 = item_in2['datetime']
79     if date_in2 not in datetimes_record_in1:
80         break
81     mismatch_index = mismatch_index + 1
82
83 if mismatch_index == len(record_in2):
84     print('下面in2被包含在in_1內') # record_out is not modified
85 else:
86     if mismatch_index == 0:
87         print('請注意下面in_2放的檔案日期是否比in_1的日期後面')
88     # most cases
89     record_out.extend(record_in2[mismatch_index:])
90
91 f_w.write(json.dumps(record_out, indent=2))
92
93 print('Got ' + basename_in1)
94
95 # save as excel file (converted from json)
96 name_without_extension = basename_in1.split('.')[0]
97 basename_excel = name_without_extension + '.xlsx'
98 filename_excel = '{}/{}'.format(folder_out, basename_excel)
99 conv = Converter()
100 conv.convert(record_out, Writer(file=filename_excel))
101
102 print('Got', basename_excel)
103
104 subdirs = glob.glob(path_inputs + '/' + '/')
105
106 for subdir in subdirs:
107     old_merger(path_output, subdir, path_output)
108
```

Ln: 109 Col: 0

114 行程式

使用 Pandas 後 (定點歷線圖程式)

```
1 import yaml
2 import glob
3 import os
4
5 import pandas as pd
6
7
8 with open('path_docklands.yml', 'rt') as f_r:
9     data = yaml.load(f_r, Loader=yaml.FullLoader)
10    FOLDER_JSON = data['new']
11    FOLDER_MERGE = data['old']
12
13 # merge json files and discard additional duplicates
14
15 files_path = FOLDER_JSON + '*.json'
16 # specify recursive to walk all parent folders of json files
17 filenames_new = glob.glob(files_path, recursive=True)
18
19
20 for filename_new in filenames_new:
21
22     path_file = os.path.split(filename_new)
23     # -> ('./inputs/20210522_103004/records_BD', 'BD_info_markers_temp21.json')
24
25     filename_no_path = path_file[-1] # get 'BD_info_markers_temp21.json'
26     path_only = os.path.split(path_file[0])
27     # -> ('./inputs/20210522_103004', 'records_BD')
28     last_subpath = path_only[-1] # get 'records_BD'
29
30     filename_merge = os.path.join(FOLDER_MERGE, last_subpath, filename_no_path)
31
32     filename_for_xlsx = filename_no_path.split('.')[0] + '.xlsx'
33     filename_xlsx = os.path.join(FOLDER_MERGE, last_subpath, filename_for_xlsx)
34
35     df_new = pd.read_json(filename_new, dtype=str) #all columns as strings
36
37     try:
38         df_old = pd.read_json(filename_merge, dtype=str) #all columns as strings
39     except:
40         df_old = None
41
42     full_df = pd.concat([df_old, df_new])
43     unique_df = full_df.drop_duplicates(subset='datetime')
44     unique_df.to_excel(filename_xlsx, index=False)
45     # specify indent for text editor to work well
46     unique_df.to_json(filename_merge, orient='records', indent=2)
47
48     print(f'merge {filename_merge} (complete)')
49
50 #print(f'merge json (& xlsx) files(complete)')
```

使用 Pandas 後縮減剩 50 行程式