利用 Python 爬蟲 104 人力銀行的資料

抓取 104 人力行上的公司資料，彙整成 excel

```
import time
import random
import requests
import pandas as pd


class Job104Spider():
    def search(self, keyword, max_mun=10, filter_params=None,
sort_type='符合度', is_sort_asc=False):
        """搜尋職缺"""
        jobs = []
        total_count = 0

        url = 'https://www.104.com.tw/jobs/search/list'
        query =
f'ro=0&kwop=7&keyword={keyword}&expansionType=area,spec,com,job,wf,wkt
m&mode=s&jobsource=2018indexpoc'
        if filter_params:
            # 加上篩選參數，要先轉換為URL 參數字串格式
            query += ''.join([f'&{key}={value}' for key, value, in
filter_params.items()])

        headers = {
            'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.92
Safari/537.36',
            'Referer': 'https://www.104.com.tw/jobs/search/',
        }

        # 加上排序條件
        sort_dict = {
            '符合度': '1',
            '日期': '2',
            '經歷': '3',
            '學歷': '4',
            '應徵人數': '7',
            '待遇': '13',
        }
        sort_params = f"&order={sort_dict.get(sort_type, '1')}"
        sort_params += '&asc=1' if is_sort_asc else '&asc=0'
        query += sort_params

        page = 1
        while len(jobs) < max_mun:
            params = f'{query}&page={page}'
            r = requests.get(url, params=params, headers=headers)
            if r.status_code != requests.codes.ok:
```

```
                print('請求失敗', r.status_code)
                data = r.json()
                print(data['status'], data['statusMsg'],
data['errorMsg'])
                break

            data = r.json()
            total_count = data['data']['totalCount']
            jobs.extend(data['data']['list'])

            if (page == data['data']['totalPage']) or (data['data']
['totalPage'] == 0):
                break
            page += 1
            time.sleep(random.uniform(3, 5))

        return total_count, jobs[:max_mun]

    def get_job(self, job_id):
        """取得職缺詳細資料"""
        url = f'https://www.104.com.tw/job/ajax/content/{job_id}'

        headers = {
            'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.92
Safari/537.36',
            'Referer': f'https://www.104.com.tw/job/{job_id}'
        }

        r = requests.get(url, headers=headers)
        if r.status_code != requests.codes.ok:
            print('請求失敗', r.status_code)
            return

        data = r.json()
        return data['data']

    def search_job_transform(self, job_data):
        """將職缺資料轉換格式、補齊資料"""
        appear_date = job_data['appearDate']
        apply_num = int(job_data['applyCnt'])
        company_addr = f"{job_data['jobAddrNoDesc']}
{job_data['jobAddress']}"

        job_url = f"https:{job_data['link']['job']}"
        job_company_url = f"https:{job_data['link']['cust']}"
        job_analyze_url = f"https:{job_data['link']['applyAnalyze']}"

        job_id = job_url.split('/job/')[-1]
        if '?' in job_id:
```

```
            job_id = job_id.split('?')[0]

        salary_high = int(job_data['salaryLow'])
        salary_low = int(job_data['salaryHigh'])

        job = {
            'job_id': job_id,
            'type': job_data['jobType'],
            'name': job_data['jobName'],  # 職缺名稱
            # 'desc': job_data['descSnippet'],  # 描述
            'appear_date': appear_date,  # 更新日期
            'apply_num': apply_num,
            'apply_text': job_data['applyDesc'],  # 應徵人數描述
            'company_name': job_data['custName'],  # 公司名稱
            'company_addr': company_addr,  # 工作地址
            'job_url': job_url,  # 職缺網頁
            'job_analyze_url': job_analyze_url,  # 應徵分析網頁
            'job_company_url': job_company_url,  # 公司介紹網頁
            'lon': job_data['lon'],  # 經度
            'lat': job_data['lat'],  # 緯度
            'education': job_data['optionEdu'],  # 學歷
            'period': job_data['periodDesc'],  # 經驗年份
            'salary': job_data['salaryDesc'],  # 薪資描述
            'salary_high': salary_high,  # 薪資最高
            'salary_low': salary_low,  # 薪資最低
            'tags': job_data['tags'],  # 標籤
        }
        return job


if __name__ == "__main__":
    job104_spider = Job104Spider()

    filter_params = {
        'area': '6001001000' # (地區) 台北市
    }
    total_count, jobs = job104_spider.search('python', max_mun=300,
filter_params=filter_params)

    print('搜尋結果職缺總數：', total_count)
    jobs = [job104_spider.search_job_transform(job) for job in jobs]

    # 將資料轉換為DataFrame
    df = pd.DataFrame(jobs)

    # 將DataFrame 寫入Excel
    df.to_excel('jobs.xlsx', index=False)
```

搜尋結果職缺總數： 3094

Excel 表:

job_id	name	ppear_dat	pply_num	pply_text	mpany_na	mpany_ad	education	period	salary
8b7fv	【2024 Ca	20240515	16	11~30人應	DELOITT	台北市信	大學	經歷不拘	待遇面議
6odpu	兒童運算	20240514	4	0~5人應徵	總公司_長	台北市中	大學	3年以上	待遇面議
80bs8	Python工程	20240513	6	6~10人應	云智資訊	台北市松	專科	3年以上	待遇面議
7f25b	python工程	20240515	7	6~10人應	迪倫蓮恩	台北市信	碩士	經歷不拘	時薪300~
8az61	Senior Pyt	20240418	4	0~5人應徵	康翔科技	台北市士	專科	經歷不拘	待遇面議
8524o	Python工程	20240503	3	0~5人應徵	皓博科技	台北市信	專科	經歷不拘	待遇面議
82dtp	Python工程	20240513	5	0~5人應徵	創昇資訊	台北市松	專科	2年以上	待遇面議
89rhw	Python 軟	20240422	8	6~10人應	核桃運算	台北市中	大學	3年以上	待遇面議
7in95	Python工程	20240426	22	11~30人應	春水堂科	台北市內	大學	經歷不拘	待遇面議
7sf7z	Python自動	20240510	4	0~5人應徵	云智資訊	台北市松	專科	2年以上	待遇面議
8aabb	數據應用	20240329	13	11~30人應	中國石油	台北市松	大學	經歷不拘	待遇面議
8b3rf	<可暑其	20240513	17	11~30人應	艾思程式	台北市中	大學	經歷不拘	時薪600~
7x79r	Python 軟	20240516	15	11~30人應	烏龜移動	台北市松	高中	1年以上	月薪40,00
7oujv	Go/Python	20240429	11	11~30人應	幣銖有限	台北市大	大學	5年以上	年薪1,500
7uito	python工程	20240516	7	6~10人應	兆徠科技	台北市內	大學	1年以上	待遇面議
89a38	Python 後	20240513	28	11~30人應	沐恩生醫	台北市中	大學	經歷不拘	待遇面議
7olfm	後端工程	20240223	14	11~30人應	海易科技	台北市南	專科	經歷不拘	月薪45,00
8awjd	[擴編] Jr./	20240508	17	11~30人應	香港商易	台北市大	專科	3年以上	月薪70,00
8avpg	<可暑其	20240513	4	0~5人應徵	艾思程式	台北市中	大學	經歷不拘	時薪600~
7izdv	[台北] 後	20240506	13	11~30人應	Linker Vis	台北市大	大學	經歷不拘	月薪50,00
7ji9s	PYTHON	20240401	7	6~10人應	美商網碩	台北市松	專科	3年以上	待遇面議
71j4l	Python 後	20240513	28	11~30人應	光禾感知	台北市大	專科	經歷不拘	待遇面議

簡單分析，計算要求教育程度和申請者數量相關性(要求教育程度越高分數越高)

```
import pandas as pd

# 讀取Excel 文件
df = pd.read_excel('jobs.xlsx')

# 將'education' 欄位轉換為數值型態
education_mapping = {'高中': 1, '大學': 2, '碩士': 3, '博士': 4}
df['education'] = df['education'].map(education_mapping)

# 計算'apply_num' 和'education' 之間的相關性
correlation = df[['apply_num', 'education']].corr()

print(correlation)
```

```
           apply_num  education
apply_num   1.000000   0.050372
education   0.050372   1.000000
```

結果: 要求教育程度和申請者數量兩者的相關性為 0.050372，幾乎沒有關連性。