



台大周遭**YouBike**站點之使用分析



Outline

專案發想




實作流程



執行與分析



資料呈現





專案發想

因應校園地大，校內及校園周遭廣設YouBike站點，
卻總是有某些站點常常無車可借、或是剩車過多需要移動。
明明特定時段車數充足、過了十分鐘卻空出整排還車架...
究竟校內外站點是否有明顯熱門時段或常被閒置的情形？



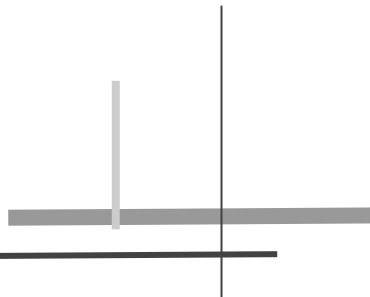
流程簡介

1. 用requests等套件將政府資料開放平台上的即時json檔下載
2. 爬下為期兩周的站點資料，並將資料匯入SQLite
3. 將所需資料從資料庫擷取出來，透過matplotlib進行資訊視覺化
4. 分析校內外各站點的借還車高峰

發送請求
至介接網站



下載每分鐘更新之json檔
使用 Python 轉為 Excel or DB 檔



爬蟲資料來源

YouBike2.0臺北市公共自行車即時資訊

為提供更優質之資料服務，YouBike臺北市公共自行車即時資訊介接網址為https://tcgbusfs.blob.core.windows.net/dotapp/youbike/v2/youbike_immediate.json，檔案格式為json檔。

評分此資料集：

平均 3.87 (15 人次投票)

瀏覽次數: 5314 下載次數: 1 意見數: 2

列印

主要欄位說明

*粗體欄位為資料標準欄位

sno(站點編號)、sna(場站中文名稱)、tot(場站總停車格)、sbi(場站目前車輛數量)、sarea(場站區域)、mday(資料更新時間)、lat(緯度)、lng(經度)、ar(地點)、sareaen(場站區域英文)、snaen(場站名稱英文)、aren(地址英文)、bemp(空位數量)、act(全站禁用狀態)、srcUpdateTime、updateTime、infoTime、infoDate

資料資源下載網址

JSON

檢視資料

YouBike2.0臺北市公共自行車即時資訊

提供機關

臺北市府交通局



爬蟲程式

```
1  import requests
2  import time
3  import os
4  from json import loads
5  from xlswriter import Workbook
6  import logging
7  FORMAT = '%(asctime)s %(levelname)s: %(message)s'
8  # ERROR(40) / INFO(30) / WARNING(20) / DEBUG(10)
9  logging.basicConfig(level=logging.ERROR, filename='youbike.log', filemode='w', format=FORMAT)
10
11  workdir = os.path.dirname(__file__)
12  print('-- workdir = ' + workdir + '---')
13  os.chdir(workdir)
14  SOURCE = "https://tcgbusfs.blob.core.windows.net/dotapp/youbike/v2/youbike_immediate.json"
15  DATA_DIR = workdir + "/data/json/"
16  CONCERNED_PATH = workdir + "/data/concerned.txt"
17  EXCEL_DIR = workdir + "/data/excel/"
18
```

資料介接網址




載入各站點資料

```
concerned = open(CONCERNED_PATH, "r", encoding="utf-8")
stations = {}
ordered_list = ['sna', 'sbi', 'bemp', 'tot', 'ar', 'act']
ordered_dict = {'sna': 'Name', 'sbi': 'Bike', 'bemp': 'Vacancy', 'tot': 'Total', 'ar': 'Location', 'act': 'Available'}
```

```
def read_stations():
    lines = concerned.readlines()
    for line in lines:
        a = line.split()
        stations.setdefault(a[0], a[1])
```

記錄各站的欄位資料



```
def parse_file_name(filename:str):
    word = filename.split('-')
    if len(word[0]) == 1:
        word[0] = '0' + word[0]
    if len(word[1]) == 1:
        word[1] = '0' + word[1]
    new_name = word[0] + '-' + word[1]
    return new_name
```

```
def main():
    startTime = time.localtime()
    startHour = startTime[3] #Hour
    read_stations()
    while True:
```

```
    date = time.localtime()
    if startHour != date[3]:
        break
```

儲存爬下來的json檔案

```
    try:
        rq = requests.get(SOURCE)
        # yyyy-mm-dd
        date_str = str(date[0]) + "-" + str(date[1]) + "-" + str(date[2])
        # hh-mm
        filename = parse_file_name(str(date[3]) + "-" + str(date[4]))
        json_path = DATA_DIR + date_str + '/' + filename[:2]
        if not os.path.exists(json_path):
            os.makedirs(json_path)
        f = open(json_path + "/" + filename + ".json", "w", encoding="utf-8")
        f.write(rq.text)
        f.close()

        print("Successfully download data: " + filename)
```

```
    except:
        logging.error("Catch an exception.", exc_info=True)
    if '59' == date[4]:
        break
    time.sleep(60)
```


將Json檔的資料匯入資料庫

```
import mariadb
import sys
import logging
import os
from os import path, walk
import json

FORMAT = '%(asctime)s %(levelname)s: %(message)s'
logging.basicConfig(level=logging.ERROR, filename='importJsonToDb.log', filemode='w', format=FORMAT)
workdir = os.path.dirname(__file__)
print('-- workdir = ' + workdir + '---')
DIRSep = os.sep
MaxDbCatchTime = ''

DATA_DIR = path.join(workdir, 'data', 'json')
DbConfig = {
    "user": "root",
    "password": "123456",
    "host": "localhost",
    "port": 3306,
    "database": "test"
}
```



登入資料庫帳密

insertDataToDb (srcList) 函式

```
#Adding Data
# Disable Auto-Commit
if not (MaxDbCatchTime):
    cur.execute(["SELECT Max(catchTime) as maxCatchTime FROM historyData"])
    for (maxCatchTime,) in cur:
        if not maxCatchTime is None:
            dt=maxCatchTime.strftime('%Y-%m-%d %H:%M:%S')
            MaxDbCatchTime = dt
        break
    cur.close()
    cur = conn.cursor()
conn.autocommit = False
for rowJson in srcList:
    # cur.execute(
    #     "INSERT INTO `historyData` (`catchTime`, `sno`, `sna`, `tot`, `sbi`, `sarea`, `mday`, `lat`, `lng`, `ar`, `sarea`
    #     (rowJson["catchTime"], rowJson["sno"], rowJson["sna"], rowJson["tot"], rowJson["sbi"], rowJson["sarea"], rowJson
    if( rowJson["catchTime"] > MaxDbCatchTime) and (rowJson["sno"] in NtuSites):
        cur.execute(
            "INSERT INTO `historyData` (`catchTime`, `sno`, `tot`, `sbi`, `mday`, `bemp`, `act`, `srcUpdateTime`, `update
            (rowJson["catchTime"], rowJson["sno"], rowJson["tot"], rowJson["sbi"], rowJson["mday"], rowJson["bemp"], rowJ
conn.commit()
# Close Connection
conn.close()
```

避免蓋掉已取得的資料

將最新一筆的資料寫入db檔中

取得校內所有站點的最新一筆資料

```
NtuSites = getNtuSites()
for (fullpath , datetimeStr) in getFileList(DATA_DIR):
    srcList = []
    with open(fullpath, encoding='utf-8') as f:
        srcList = json.load(f)
        for row in srcList:
            row["catchTime"] = datetimeStr

insertDataToDb (srcList)
```

資料表結構為

```
CREATE TABLE `historyData` (
  `catchTime` datetime NOT NULL,
  `sno` varchar(10) NOT NULL,
  `sna` varchar(50) NOT NULL,
  `tot` int(11) NOT NULL,
  `sbi` int(11) NOT NULL,
  `sarea` varchar(50) NOT NULL,
  `mday` datetime NOT NULL,
  `lat` decimal(10,6) NOT NULL,
  `lng` decimal(10,6) NOT NULL,
  `ar` varchar(50) NOT NULL,
  `sareaen` varchar(50) NOT NULL,
  `snaen` varchar(100) NOT NULL,
  `aren` varchar(100) NOT NULL,
  `bemp` int(11) NOT NULL,
  `act` int(11) NOT NULL,
  `srcUpdateTime` datetime NOT NULL,
  `updateTime` datetime NOT NULL,
  `infoTime` datetime NOT NULL,
  `infoDate` date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

蒐集資料介紹

Data file (MariaDB)

✓ 顯示第 0 - 24 列 (總計 1377432 筆, 查詢用了 0.0003 秒。)

SELECT * FROM `historyData`

☐ 效能分析 [行內編輯] [編輯] [SQL 語句分析] [建立 PHP 程式碼]

1

































> >>

資料列數: 25

篩選資料列: 搜尋此資料表

依主鍵排序: 無

+ 選項

		catchTime	sno	tot	sbi	mday	bemp	act	srcUpdateTime	updateTime	infoTime	infoDate		
<input type="checkbox"/>	 編輯	 複製	 刪除	2021-12-19 10:17:00	500101022	60	1	2021-12-19 10:14:11	59	1	2021-12-19 10:16:14	2021-12-19 10:16:50	2021-12-19 10:14:11	2021-12-19
<input type="checkbox"/>	 編輯	 複製	 刪除	2021-12-19 10:17:00	500101026	15	0	2021-12-19 09:54:03	15	1	2021-12-19 10:16:14	2021-12-19 10:16:50	2021-12-19 09:54:03	2021-12-19
<input type="checkbox"/>	 編輯	 複製	 刪除	2021-12-19 10:17:00	500101028	20	3	2021-12-19 10:11:11	17	1	2021-12-19 10:16:14	2021-12-19 10:16:50	2021-12-19 10:11:11	2021-12-19
<input type="checkbox"/>	 編輯	 複製	 刪除	2021-12-19 10:17:00	500101029	15	1	2021-12-19 10:11:03	14	1	2021-12-19 10:16:14	2021-12-19 10:16:50	2021-12-19 10:11:03	2021-12-19
<input type="checkbox"/>	 編輯	 複製	 刪除	2021-12-19 10:17:00	500101030	34	2	2021-12-19 09:23:11	32	1	2021-12-19 10:16:14	2021-12-19 10:16:50	2021-12-19 09:23:11	2021-12-19
<input type="checkbox"/>	 編輯	 複製	 刪除	2021-12-19 10:17:00	500101031	20	2	2021-12-19 10:11:11	17	1	2021-12-19 10:16:14	2021-12-19 10:16:50	2021-12-19 10:11:11	2021-12-19
<input type="checkbox"/>	 編輯	 複製	 刪除	2021-12-19 10:17:00	500101032	61	1	2021-12-19 10:12:10	60	1	2021-12-19 10:16:14	2021-12-19 10:16:50	2021-12-19 10:12:10	2021-12-19
<input type="checkbox"/>	 編輯	 複製	 刪除	2021-12-19 10:17:00	500101033	46	0	2021-12-19 09:58:03	46	1	2021-12-19 10:16:14	2021-12-19 10:16:50	2021-12-19 09:58:03	2021-12-19
<input type="checkbox"/>	 編輯	 複製	 刪除	2021-12-19 10:17:00	500101034	18	2	2021-12-19 10:06:11	16	1	2021-12-19 10:16:14	2021-12-19 10:16:50	2021-12-19 10:06:11	2021-12-19
<input type="checkbox"/>	 編輯	 複製	 刪除	2021-12-19 10:17:00	500101035	11	5	2021-12-19 10:14:11	6	1	2021-12-19 10:16:14	2021-12-19 10:16:50	2021-12-19 10:14:11	2021-12-19

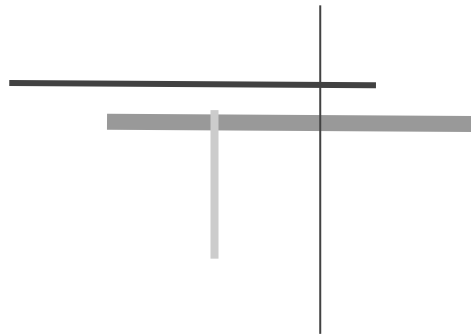
Json 檔資料欄位

欄位名稱	說明	欄位名稱	說明
sno [PK]	站點代號	ar	地點
sna	場站中文名稱	sareaen	場站區域英文
tot	場站總停車格	snaen	場站名稱英文
sbi	場站目前車輛數量	aren	地址英文
sarea	場站區域	bemp	空位數量
mday	資料更新時間	act	全站禁用狀態
lat	緯度	srcUpdateTime、updateTime、 infoTime、infoDate	
lng	經度	catchTime	



預期分析內容

- 各個站點一周內的借還車流量
- 平日與假日曲線的差異
- 教學建築與宿舍周遭的曲線差異



資料蒐集時間: 2021 / 12 / 20 00:00:00 - 2022 / 1 / 3 00:00:00



e.g. 單一站點於單週內的借還車流量

```
SELECT catchTime ,sbi  
FROM historyData  
WHERE sno='{sno}' and catchTime >= '2021-12-20 00:00:00' and catchTime <=  
'2021-12-27 00:30:00'  
and substr(catchTime,15,5)='30:00'  
Order by catchTime
```

分析頻率: 每60分鐘一次

比較日期: 2021 / 12 / 20 - 2021 / 12 / 26



e.g.平日宿舍周遭與教學館周遭站點之的車流變化

```
SELECT catchTime ,sbi ,sno  
FROM historyData  
WHERE (sno='{sno1}' Or sno='{sno2}') and catchTime >= '2021-12-20 00:00:00'  
and catchTime < '2021-12-21 00:00:00'  
and (substr(catchTime,16,1)='0' or substr(catchTime,16,1)='5') Order by  
sno,catchTime
```

分析頻率: 每5分鐘一次

比較日期: 2021 / 12 / 20 - 2021 / 12 / 23 V.S. 2021 / 12 / 25



e.g. 單一站點於平日與假日的車流量比較

```
SELECT catchTime ,sbi  
FROM historyData  
WHERE sno='{sno}' and catchTime >= '2021-12-20 00:00:00' and catchTime <=  
'2021-12-27 00:30:00'  
and (substr(catchTime,16,1)='0' or substr(catchTime,16,1)='5') Order by  
catchTime
```

分析頻率: 每5分鐘一次

比較日期: 2021 / 12 / 20 - 2021 / 12 / 26

單週次曲線的視覺化呈現

```
Db_File = path.join(workdir, 'youbikeData.db')

try:
    sqliteConn = sqlite3.connect(Db_File)
    sqliteCursor = sqliteConn.cursor()
    # MariaDB SQL
    #SELECT hd.*,ns.sna FROM historyData hd Inner Join ntuSiteData ns On hd.sno=ns.sno WHERE ns.sno='500119048'
    #SELECT mday ,sbi, catchTime FROM historyData WHERE sno='500119048' and mday > '2021-12-20 00:00:00' and mday < '2021-12-21 00:00:00'

    sqliteCursor.execute(f"SELECT catchTime ,sbi FROM historyData WHERE sno='500119048' and catchTime >= '2021-12-20 00:00:00'")
    allRow = sqliteCursor.fetchall()
    sqliteCursor.close()
    xList = []
    yList = []
    for r in allRow:
        xList.append(r[0])
        yList.append(r[1])
    plt.plot(xList, yList)

    plt.show()
except Exception as e:
    print(f"-- exception --{e}")
    logging.error("Catch an exception.", exc_info=True)
    sys.exit(1)
```

使用套件執行SQL語法



可能限制與可行性評估

1. 所蒐集資料僅有一至兩周，無法確定是否為長期現象。
2. 不能確定站點車輛數目下降是真的沒車或是youbike公司為節省空間將車輛取出。
3. 當資料收集量夠大，未來可能可以發展出較「以經驗預測」更為精準的時間-可用車輛預測模型。

