

## 1 目標

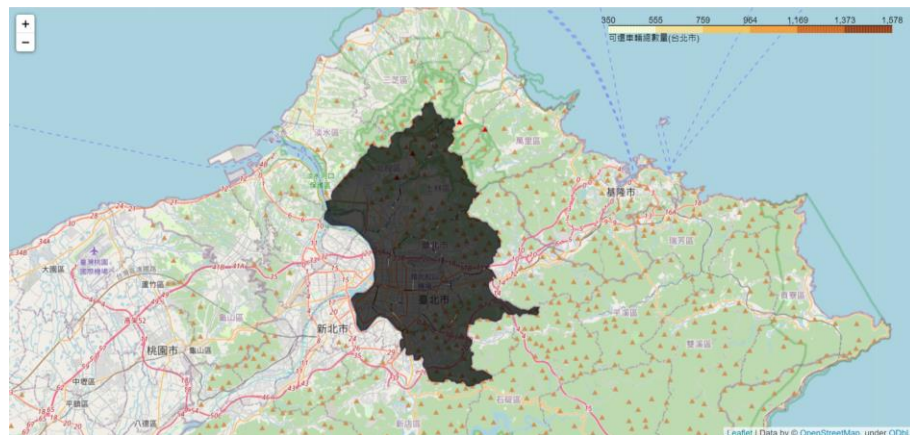
- 1.1 台北市分區可還車輛面量圖
- 1.2 新北市分區可還車輛面量圖
- 1.3 台灣各縣市腳踏車總共數量面量圖
- 1.4 其他種類的地圖

## 2 完成項目

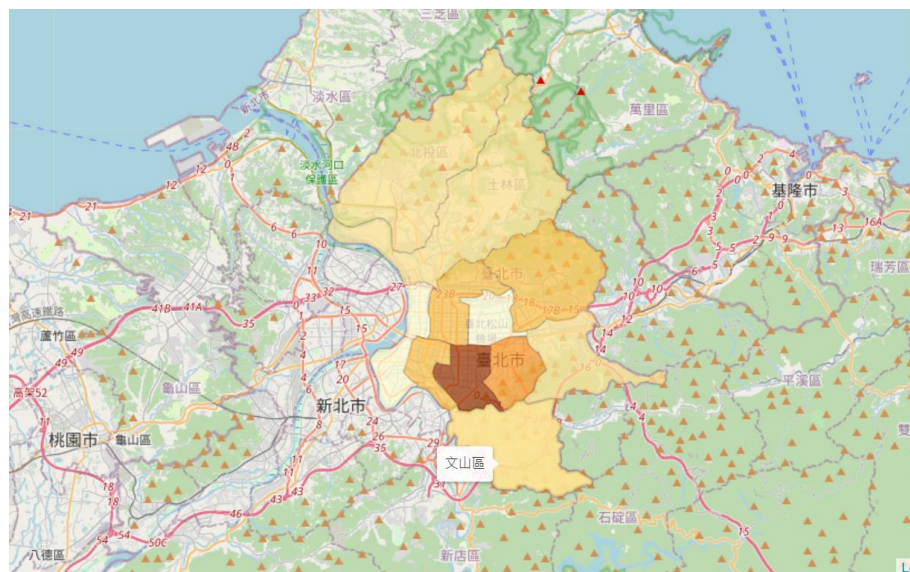
- 2.1 台北市分區可還車輛面量圖
- 2.2 新北市分區可還車輛面量圖

## 3 遭遇困難

- 3.1 使用 Choropleth 函數時，跑出來面量圖的顏色是黑色，而不是設定好的漸層顏色，而且連右上方「可還車總數量(台北市)」的量表都沒有出現



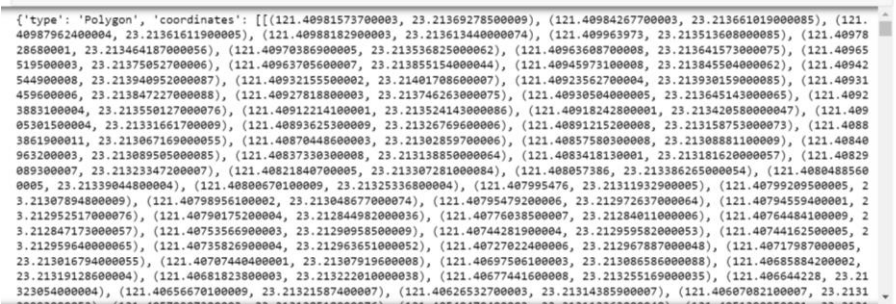
- 3.2 當滑鼠停留在台北市某個分區時，除了顯示分區名之外，想要顯示某分區可還車量總數量



下方其中一段比較重要的程式碼，能夠把上方的圖畫出來，大概稍微

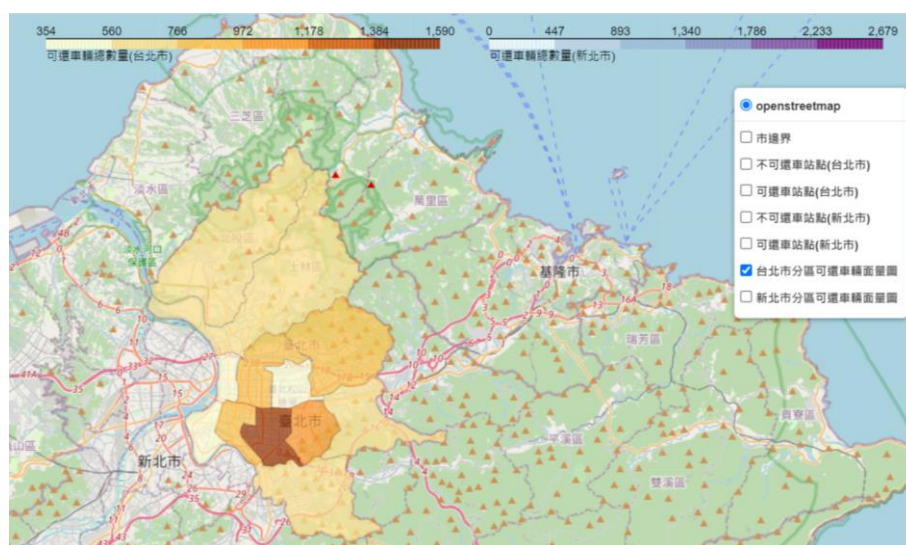
講解一下這一段程式碼，geo\_data 儲存台北市每個分區的邊界所有的座標，如下方的圖所示。data 儲存每個分區的名字和每個分區總共的數量，如下方的圖所示。整個 Choropleth 函數負責把整個面量圖和量表畫出來，而 add\_child 函數做的事情就是，當滑鼠停留在某一個分區時，他會告訴我這個分區的名字。除了告訴分區的名字，我希望可以知道每個分區的可還車輛總數量，但這個部份仍然不知道要加在程式碼的哪一段會比較好，或是修改原有的程式碼並且融合我想要的功能

```
choropleth = folium.Choropleth(  
    geo_data = geoData,  
    name = "台北市分區可還車輛面量圖",  
    data = data,  
    columns = ["sarea", "bemp"],  
    key_on = "feature.properties.TOWNNAME",  
    fill_color = "YlOrBr",  
    fill_opacity = 0.7,  
    line_opacity = 0.1,  
    legend_name = "可還車輛總數量(台北市)",  
    highlight = True  
)  
.add_to(m)  
  
choropleth.geojson.add_child(  
    folium.features.GeoJsonTooltip(['TOWNNAME'], labels = False)  
)
```



	sarea	bemp
0	中山區	788
1	中正區	849
2	信義區	887
3	內湖區	753
4	北投區	675
5	南港區	632
6	士林區	814
7	大同區	401
8	大安區	1421
9	文山區	538
10	松山區	342
11	萬華區	435

- 3.3 我想要點選 openstreetmap 中其中一個選項時，只出現該點選的地圖內容。但現在遇到的問題是，當我取消「新北市分區可還車輛面量圖」的勾選時，該地圖的量表仍然出現在右上方，如下方的圖所示



#### 4 如何解決

- 4.1 問題: 使用 Choropleth 函數時，跑出來結果跟預想的不太一樣  
解決過程:

先看一下下方的程式碼，Choropleth 函數有幾個參數，我認為問題應該就是出在參數沒有代入正確，才會造成跑出來的結果跟自己預想的不太一樣，光這個問題就卡蠻久的，那時候，我覺得問題可能出在 geo\_data、data、columns 和 key\_on 這幾個參數上，因為剩下來的參數只是設定顏色、名字和簡單的功能而已，不太會影響整個跑出來的結果。

於是我就去研究這幾個可能會發生問題的參數，到網路上尋找跟 Choropleth 相關的資料，跟自己寫的程式碼差在哪邊，然後，自己也有去研究 geoData 和 data 的內容，以及資料的格式是否符合參數代入的形式，最後內容和形式沒問題。那問題，就可能出在 columns 和 key\_on 參數身上，columns 參數我研究到最後，他是跟 data 有關係，如果 columns 標的欄位名稱沒有在 data 中，就會發生錯誤。所以，問題就出在 key\_on 的參數代入錯誤，我原本以為，key\_on 跟 data 是比較有關係的，根據我瀏覽過許多人分享與 Choropleth 相關的例子，結果不是，他是跟 geo\_data 比較有關係，而且我那時候還有花時間去了解什麼是 geojson，因為 key\_on 參數會用到 geojson 概念，另外我還得去了解 shapefile 內部的構造，因為 geoData 我是讀取 shp 檔得到的，又跟 key\_on 參數也有關係，所以我要先知道我讀取的 shp 檔的架構長怎樣，下方的圖是讀取 shp 檔的內部結構。



最後，data 資料裝的是每個分區的名字和每個分區可還車輛總數量，columns 代入 data 中的兩個欄位的名字，geoData 除了裝每個分區邊界所有的座標之外，其實他也有記錄每個分區的名字，只不過欄位是 TOWNNAME，對應到 data 中的 sarea 欄位，key\_on 採用 geoData 形式讀取 geoData 的 TOWNNAME 欄位，又跟 data 的 sarea 欄位做對應，然後 data 的每一個 sarea 對應到那個分區可還車輛總數量，所以才能畫出每個分區的面量圖，以及劃出每個分區的邊界。

```
choropleth = folium.Choropleth(
    geo_data = geoData,
    name = "台北市分區可還車輛面量圖",
    data = data,
    columns = ["sarea", "bemp"],
    key_on = "feature.properties.TOWNNAME",
    fill_color = "YlOrBr",
    fill_opacity = 0.7,
    line_opacity = 0.1,
    legend_name = "可還車輛總數量(台北市)",
    highlight = True
).add_to(m)
```

```
{'type': 'FeatureCollection',
 'features': [{ 'id': '150',
  'type': 'Feature',
  'properties': { 'COUNTYCODE': '63000',
   'COUNTYID': 'A',
   'COUNTYNAME': '臺北市',
   'TOWNCODE': '63000030',
   'TOWNENG': "Da'an District",
   'TOWNID': 'A02',
   'TOWNNAME': '大安區'},
  'geometry': { 'type': 'Polygon',
   'coordinates': (((121.543841724, 25.044906697000044),
    (121.54407412800003, 25.044928091000088),
    (121.54489838300003, 25.045039620000007),
    (121.54534485800002, 25.045100473000048),
    (121.54588290700008, 25.045171390000064),
    (121.54634082400003, 25.045232181000074),
    (121.54653536600006, 25.045241884000063),
    (121.54674307800008, 25.045216703000005),
```

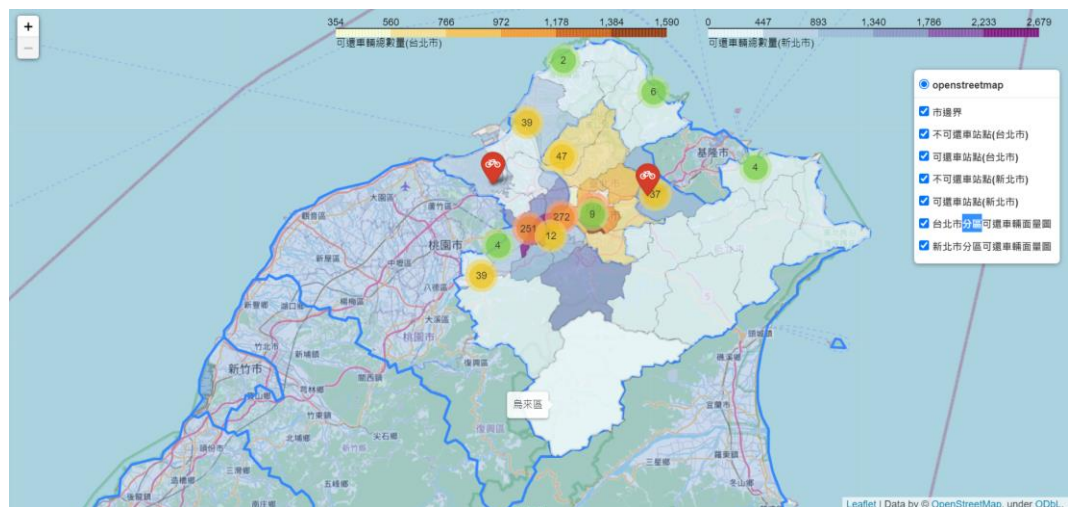
	TOWNID	TOWNCODE	COUNTYNAME	TOWNNAME	TOWNENG	COUNTYID	COUNTYCODE	geometry
150	A02	63000030	臺北市	大安區	Da'an District	A	63000	POLYGON (((121.54384 25.04491, 121.54407 25.044...
151	A11	63000080	臺北市	文山區	Wenshan District	A	63000	POLYGON (((121.59725 25.01439, 121.59724 25.014...
331	A17	63000020	臺北市	信義區	Xinyi District	A	63000	POLYGON (((121.57995 25.04950, 121.57996 25.049...
332	A05	63000070	臺北市	萬華區	Wanhua District	A	63000	POLYGON (((121.50521 25.04949, 121.50554 25.049...
333	A03	63000050	臺北市	中正區	Zhongzheng District	A	63000	POLYGON (((121.51341 25.04940, 121.51375 25.049...
334	A13	63000090	臺北市	南港區	Nangang District	A	63000	POLYGON (((121.61425 25.06451, 121.61425 25.064...
335	A01	63000010	臺北市	松山區	Songshan District	A	63000	POLYGON (((121.56971 25.07629, 121.56981 25.076...
336	A09	63000060	臺北市	大同區	Datong District	A	63000	POLYGON (((121.51410 25.07942, 121.51488 25.079...
337	A10	63000040	臺北市	中山區	Zhongshan District	A	63000	POLYGON (((121.54147 25.09287, 121.54155 25.092...
338	A14	63000100	臺北市	內湖區	Neihu District	A	63000	POLYGON (((121.59359 25.11515, 121.59368 25.115...
339	A15	63000110	臺北市	士林區	Shilin District	A	63000	POLYGON (((121.58088 25.19535, 121.58101 25.195...
340	A16	63000120	臺北市	北投區	Beitou District	A	63000	POLYGON (((121.55990 25.21014, 121.56010 25.210...

## 5 未來計畫

- 5.1 解決「當滑鼠停留在台北市某個分區時，除了顯示分區名之外，想要顯示某分區可還車量總數量」的問題，因為三個面量圖都會用到它
- 5.2 尋找台灣所有縣市關於腳踏車的資料
- 5.3 畫出「台灣各縣市腳踏車總共數量面量圖」
- 5.4 畫出其他種類的地圖，大概兩到三個左右
- 5.5 解決「量表沒有消失」的問題

## 6 程式碼

### 6.1 最終產品



### 6.2 台北市分區可還車輛面量圖

```
def AddTaipeiChoroplethToMap(m):
    url = 'https://tcgbusfs.blob.core.windows.net/dotapp/youbike/v2/youbike_immediate.json'
    data = requests.get(url).text
    data = json.loads(data)
    data = pd.DataFrame(data)
    data = data[['sarea', 'bemp']]
    data = data[data['sarea'] != '臺大專區']
    data = data.groupby('sarea').sum()
    data.reset_index(inplace=True)

    geoData = gpd.read_file(r'./TOWN_MOI_1100415.shp', encoding = 'utf-8')
    geoData = geoData[geoData['COUNTYNAME'] == '臺北市']

    choropleth = folium.Choropleth(
        geo_data = geoData,
        name = "台北市分區可還車輛面量圖",
        data = data,
        columns = ["sarea", "bemp"],
        key_on = "feature.properties.TOWNNAME",
        fill_color = "YlOrBr",
        fill_opacity = 0.7,
        line_opacity = 0.1,
        legend_name = "可還車輛總數量(台北市)",
        highlight = True
    ).add_to(m)

    choropleth.geojson.add_child(
        folium.features.GeoJsonTooltip(['TOWNNAME'], labels = False)
    )
```

### 6.3 新北市分區可還車輛面量圖

說明: 由於新北市自行車資料集只提供歷史資料, 所以我下載 csv 檔, 再透過 read\_csv 函數讀取資料, 不像台北市自行車資料集可以提供即時性資料, 就直接讀取網址取得資料。另外, 新北市自行車資料集沒有坪林區、雙溪區、烏來區、石碇區、平溪區和貢寮區資料, 所以 bemp 預設為 0, 這麼做的理由是, 這些分區沒有設 youbike 站點, 沒有站點就不會有可以還車的數量, 因此可還車數量設為 0。

```
def AddXinbeiChoroplethToMap(m):
    data = pd.read_csv('./新北市公共自行車租賃系統(YouBike).csv')
    data = data[['sarea', 'bemp']]
    data = data[data['sarea'] != '臺大專區']
    data = data.groupby('sarea').sum()
    data.reset_index(inplace=True)
    sarealist = ["坪林區", "雙溪區", "烏來區", "石碇區", "平溪區", "貢寮區"]
    for element in sarealist:
        data = data.append({'sarea': element, 'bemp': 0}, ignore_index = True)

    geoData = gpd.read_file(r'./TOWN_MOI_1100415.shp', encoding = 'utf-8')
    geoData = geoData[geoData['COUNTYNAME'] == '新北市']

    choropleth = folium.Choropleth(
        geo_data = geoData,
        name = "新北市分區可還車輛面量圖",
        data = data,
        columns = ["sarea", "bemp"],
        key_on = "feature.properties.TOWNSNAME",
        fill_color = "BuPu",
        fill_opacity = 0.7,
        line_opacity = 0.1,
        legend_name = "可還車輛總數量(新北市)",
        highlight = True
    ).add_to(m)

    choropleth.geojson.add_child(
        folium.features.GeoJsonTooltip(['TOWNSNAME'], labels = False)
    )
```

## 6.4 主要程式碼

說明: 透過呼叫函數把每一種地圖加入到地圖中

```
import requests
import json
import folium
import geopandas as gpd
from folium.plugins import MarkerCluster
import pandas as pd
```

```
m = folium.Map((25.0133904, 121.52245), zoom_start = 11, min_zoom = 10)
```

```
m = AddBoundaryCityToMap(m)
m = AddTaipeiBikeToMap(m)
m = AddXinbeiBikeToMap(m)
m = AddTaipeiChoroplethToMap(m)
m = AddXinbeiChoroplethToMap(m)
```

```
folium.LayerControl(collapsed = False).add_to(m)
```

## 7 參考來源

### 7.1 [geojson.io](http://geojson.io)

- 7.2 [透過 Google Map API 處理 GeoJSON 資料](#)
- 7.3 [geopandas.GeoDataFrame. \\_geo\\_ interface](#)
- 7.4 [folium 各函式參數介紹](#)
- 7.5 [folium choropleth not showing when called](#)
- 7.6 [How to Build choropleth map in Python | Streamlit Tutorial #3 | Data Driven Maps With Python Folium](#)
- 7.7 [python-shapefile 庫讀取 shapefile 文件信息](#)
- 7.8 [Using Folium to Generate Choropleth Map with Customised Tooltips\(Python\)](#)