

CHAT

```
# 印出資料
print(df)
```

這段代碼首先讀取CSV文件並將其轉換為一個pandas DataFrame。然後，它將第二欄資料轉換為日期格式，並只保留4月3日之後的資料。最後，它印出結果。

請注意，這段代碼假設時間列是CSV文件的第二欄，並且時間的格式可以被pandas的 `to_datetime` 函數正確解析。如果實際的情況與此不同，則可能需要修改這段代碼。

abc970399

single positional indexer is out-of-bounds File "C:\Users\User\Desktop\cycu\_ai2024\20240409\try2.py", line 7, in df.iloc[:, 1] = pd.to\_datetime(df.iloc[:, 1]) ~~~~~^ IndexError: single positional indexer is out-of-bounds

GitHub Copilot

Used 1 reference >

這個錯誤表明您的DataFrame可能沒有足夠的列。在這種情況下，您可能需要檢查您的CSV文件以確定它的結構。

如果您確定第二欄存在並且應該被解析為日期，那麼您可能需要確保在讀取CSV文件時包含了所有必要的列。例如，如果您的CSV文件使用了某種分隔符，那麼您需要在 `read_csv` 函數中指定它。

以下是一個修改過的代碼示例，它使用逗號作為分隔符：

```
import pandas as pd

# 讀取CSV文件，使用'big5'編碼
df = pd.read_csv('C:\\Users\\User\\Desktop\\cycu_ai2024\\20240409\\地震活動彙整_638482849776798722.csv',

# 將第二欄資料轉換為日期格式
df.iloc[:, 1] = pd.to_datetime(df.iloc[:, 1])

# 只保留4月3日後的資料
df = df[df.iloc[:, 1] > '2024-04-03']

# 印出資料
print(df)
```

如果這仍然不能解決問題，那麼您可能需要檢查您的CSV文件以確定其結構。

如何處理

如何使用Pandas將CSV文件中的日期時間列轉換為Python的datetime對象？

詢問 Copilot 或輸入 / 以取得命令

try2.py

try3.py

20240312.py

20240409 > try3.py > ...

```
20 m = folium.Map(location=[latitude.mean(), longitude.mean()], zoom_start=6)
21
22 # 創建一個用於存儲地點的列表
23 locations = []
24
25 # 將地點添加到列表中
26 for index, row in filtered_df.iterrows():
27     locations.append({
28         'coordinates': [row['緯度'], row['經度']],
29         'time': row['地震時間'].isoformat(),
30         'properties': {
31             'popup': f'Time: {row['地震時間']}<br>Latitude: {row['緯度']}<br>Longitude: {row
32             'icon': 'circle',
33             'iconstyle': {
34                 'fillColor': 'red' if row['規模'] >= 6 else 'orange' if row['規模'] == 5 el
35                 'fillOpacity': 0.6,
36                 'stroke': 'false',
37                 'radius': row['規模']
38             }
39         }
40     })
41
42 # 將地點添加到地圖上
43 TimestampedGeoJson(locations, period='PT1H', add_last_point=True).add_to(m)
44
45 # 將地圖保存為HTML，並將其保存到指定的路徑
46 m.save('C:\\Users\\User\\Desktop\\cycu_ai2024\\20240409\\earthquake_locations.html')
47 else:
48     print("DataFrame中沒有'地震時間'這一列")
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

Python Deb...

Python Deb...

PS C:\Users\User\Desktop\cycu\_ai2024>
PS C:\Users\User\Desktop\cycu\_ai2024> c;; cd 'C:\Users\User\Desktop\cycu\_ai2024'; & 'c:\Users\User\AppData\Local\Programs\Python\Python311\python.exe' 'c:\Users\User\.vscode\extensions\ms-python.debugpy-2024.4.0-w
in32-x64\bundled\libs\debugpy\adapter\..\..\debugpy\launcher' '55070' '--' 'c:\Users\User\Desktop\cycu\_ai20
24\20240409\try2.py'
PS C:\Users\User\Desktop\cycu\_ai2024> ^C
PS C:\Users\User\Desktop\cycu\_ai2024>
PS C:\Users\User\Desktop\cycu\_ai2024> c;; cd 'C:\Users\User\Desktop\cycu\_ai2024'; & 'c:\Users\User\AppData\Local\Programs\Python\Python311\python.exe' 'c:\Users\User\.vscode\extensions\ms-python.debugpy-2024.4.0-w
in32-x64\bundled\libs\debugpy\adapter\..\..\debugpy\launcher' '55164' '--' 'c:\Users\User\Desktop\cycu\_ai20
24\20240409\try3.py'
PS C:\Users\User\Desktop\cycu\_ai2024> []

Ln 46, Col 83

Spaces: 4

UTF-8

CRLF

Python

3.11.5 64-bit

下午 08:15

2024/4/9

CHAT

GitHub Copilot

Used 1 reference >

要根據時間將點添加到地圖上並創建動態地圖，我們可以使用 `folium` 庫的 `TimestampedGeoJson` 插件。以下是如何實現這一點的代碼：

```
import pandas as pd
import folium
from folium.plugins import TimestampedGeoJson

# 讀取CSV文件，使用'big5'編碼，跳過第一行
df = pd.read_csv('C:\\Users\\User\\Desktop\\cycu_ai2024\\20240409\\地震活動彙整_63848284977698722.csv',

# 檢查'地震時間'是否在列名中
if '地震時間' in df.columns:
    # 將'地震時間'轉換為datetime格式
    df['地震時間'] = pd.to_datetime(df['地震時間'])
    # 過濾數據，只保留4/3 7:58之後的記錄
    filtered_df = df[df['地震時間'] > '4/3 7:58']

# 讀取'經度'和'緯度'列
longitude = df['經度']
latitude = df['緯度']

# 創建一個地圖對象
m = folium.Map(location=[latitude.mean(), longitude.mean()], zoom_start=6)

# 創建一個用於存儲地點的列表
locations = []

# 將地點添加到列表中
for index, row in filtered_df.iterrows():
    locations.append({
        'coordinates': [row['緯度'], row['經度']],
        'time': row['地震時間'].isoformat(),
        'properties': {
            'popup': f"Time: {row['地震時間']}<br>Latitude: {row['緯度']}<br>Longitude: {row['經度']}",
            'icon': 'circle',
            'iconstyle': {
                'fillColor': 'red' if row['規模'] >= 6 else 'orange' if row['規模'] == 5 else 'yellow',
                'fillOpacity': 0.6,
                'stroke': 'false',
                'radius': row['規模']
            }
        }
    })

# 將地點添加到地圖上
TimestampedGeoJson(locations, period='PT1H', add_last_point=True).add_to(m)

# 將地圖保存為HTML，並將其保存到指定的路徑
m.save('C:\\Users\\User\\Desktop\\cycu_ai2024\\20240409\\earthquake_locations.html')
else:
    print("DataFrame中沒有'地震時間'這一列")
```

如何使用Pandas將CSV文件中的日期時間列轉換為Python的datetime對象？

詢問 Copilot 或輸入 / 以取得命令

try2.py

try3.py

20240312.py

20240409 > try3.py > ...

```
20 m = folium.Map(location=[latitude.mean(), longitude.mean()], zoom_start=6)
21
22 # 創建一個用於存儲地點的列表
23 locations = []
24
25 # 將地點添加到列表中
26 for index, row in filtered_df.iterrows():
27     locations.append({
28         'coordinates': [row['緯度'], row['經度']],
29         'time': row['地震時間'].isoformat(),
30         'properties': {
31             'popup': f"Time: {row['地震時間']}<br>Latitude: {row['緯度']}<br>Longitude: {row
32             'icon': 'circle',
33             'iconstyle': {
34                 'fillColor': 'red' if row['規模'] >= 6 else 'orange' if row['規模'] == 5 el
35                 'fillOpacity': 0.6,
36                 'stroke': 'false',
37                 'radius': row['規模']
38             }
39         }
40     })
41
42 # 將地點添加到地圖上
43 TimestampedGeoJson(locations, period='PT1H', add_last_point=True).add_to(m)
44
45 # 將地圖保存為HTML，並將其保存到指定的路徑
46 m.save('C:\\Users\\User\\Desktop\\cycu_ai2024\\20240409\\earthquake_locations.html')
47 else:
48     print("DataFrame中沒有'地震時間'這一列")
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

Python Deb...

Python Deb...

PS C:\Users\User\Desktop\cycu\_ai2024>
PS C:\Users\User\Desktop\cycu\_ai2024> c:: cd 'c:\Users\User\Desktop\cycu\_ai2024'; & 'c:\Users\User\AppData\Local\Programs\Python\Python311\python.exe' 'c:\Users\User\.vscode\extensions\ms-python.debugpy-2024.4.0-w
in32-x64\bundled\debugpy\adapter\...\debugpy\launcher' '55070' '--' 'c:\Users\User\Desktop\cycu\_ai20
24\20240409\try2.py'
PS C:\Users\User\Desktop\cycu\_ai2024> C
PS C:\Users\User\Desktop\cycu\_ai2024>
PS C:\Users\User\Desktop\cycu\_ai2024> c:: cd 'c:\Users\User\Desktop\cycu\_ai2024'; & 'c:\Users\User\AppData\Local\Programs\Python\Python311\python.exe' 'c:\Users\User\.vscode\extensions\ms-python.debugpy-2024.4.0-w
in32-x64\bundled\debugpy\adapter\...\debugpy\launcher' '55164' '--' 'c:\Users\User\Desktop\cycu\_ai20
24\20240409\try3.py'
PS C:\Users\User\Desktop\cycu\_ai2024>

Ln 46, Col 83 Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit 下午 08:13 2024/4/9

The image shows a Windows desktop with a VS Code editor window open. The editor has a dark theme and displays a Python script named 'try3.py'. The script uses the 'folium' library to create a map and the 'pandas' library to read a CSV file. The CSV file is located at 'C:\\Users\\User\\Desktop\\cycu\_ai2024\\20240409\\地震活動彙整\_638482849776798722.csv'. The script reads the CSV file, filters the data to only include earthquakes that occurred after April 3, 2024, and then creates a map showing the locations of these earthquakes. The map is titled '20240409' and is saved as 'earthquake\_locations.html'. The script also prints the number of earthquakes found. The terminal window at the bottom shows the command prompt output, which includes the file path and the command to run the script. The output shows that the script successfully ran and printed the number of earthquakes found. The Windows taskbar at the bottom shows the Start button, a search bar, and several open applications including VS Code, a file explorer, and a web browser. The system clock in the bottom right corner shows the date as 2024/4/9 and the time as 08:14.