

[src](#)

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import seaborn as sns
4
5 import plotly.express as px
6
7 import folium
8 from folium.plugins import HeatMap
9
10 import glob
11 import warnings
12 warnings.filterwarnings('ignore')
```

```
1 from google.colab import drive
2 drive.mount('/content/gdrive')
3
4 folder= '/content/gdrive/MyDrive/2023/2023-2-Python-AI/earth-data/'
5
```

⇄ Mounted at /content/gdrive

```
1 file_list = glob.glob(folder + "/*.csv")
2
```

```
1 # data in the first line, skip this line
2 dfs=[]
3 for file in file_list:
4     df=pd.read_csv(file, skiprows=1,encoding="big5")
5     dfs.append(df)
6
7 df= pd.concat(dfs)
8
9
```

```
1 #df= pd.concat(dfs)
2 #df.head(2)
3 df.head(2)
```



	編號	地震時間	經度	緯度	規模	深度	位置
0	小區域有感地震	2024-05-25 15:30:19	121.420	22.4425	4.2	21.3	臺東縣政府東南方 44.3 公里 (位於臺灣東南部海域)(臺灣東南部海域)
1	小區域有感地震	2024-05-25 13:30:50	121.674	24.2543	3.9	28.3	花蓮縣政府北北東方 29.7 公里 (位於花蓮縣秀林鄉)(花蓮縣秀林鄉)

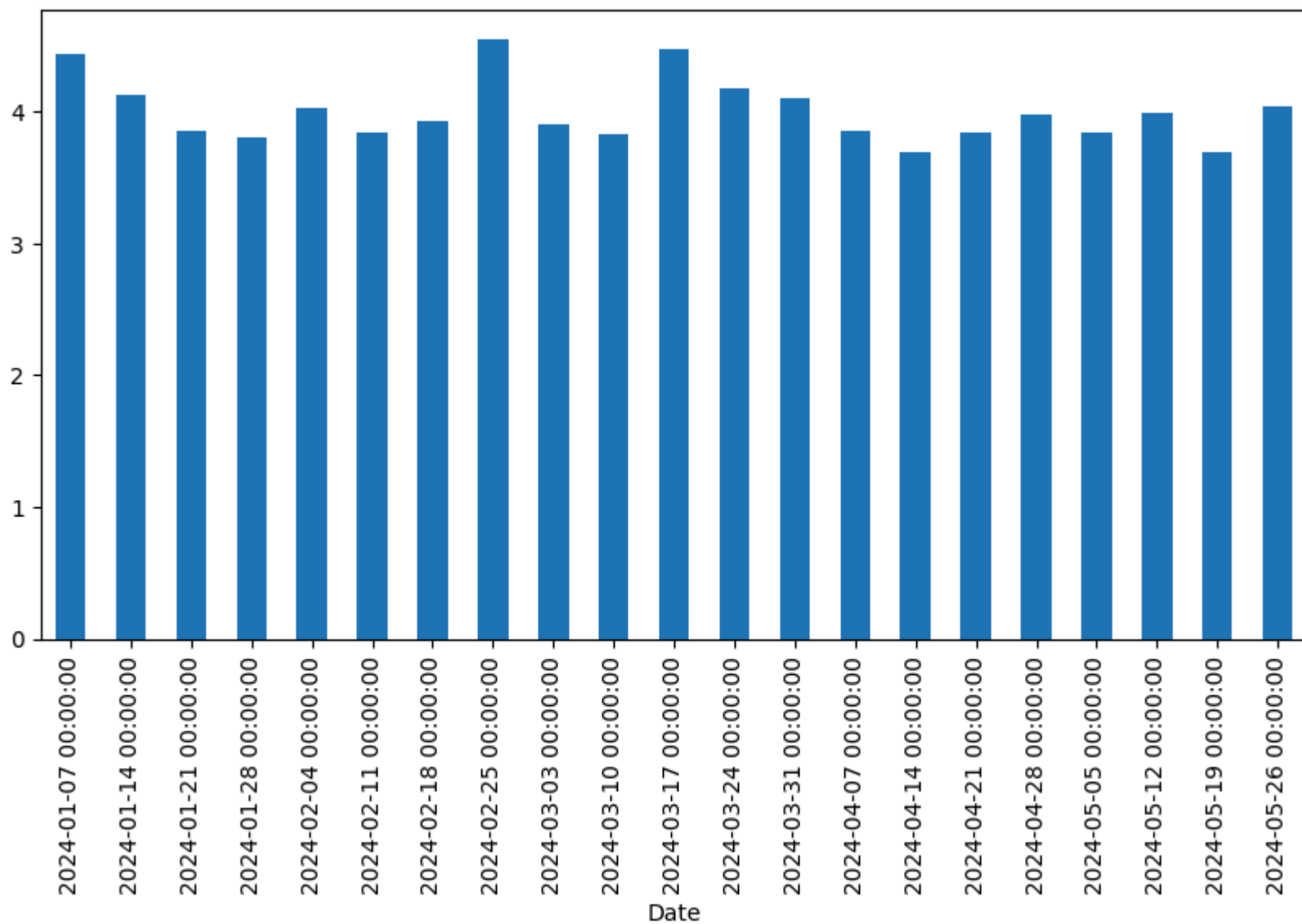
```
1 df = pd.read_csv('dataset.csv')
2
3
4 splitted = df['地震時間'].str.split(' ', n=1,
5                                     expand=True)
6 df['Date'] = splitted[0]
7 df['Time'] = splitted[1].str[:4]
8 df['Date'] = pd.to_datetime(df['Date'], format='%Y-%m-%d')
9 df.drop('地震時間',
10        axis=1,
11        inplace=True)
```

```
1 df.head(2)
```



	編號	經度	緯度	規模	深度	位置	Date	Time
0	小區域有感地震	121.420	22.4425	4.2	21.3	臺東縣政府東南方 44.3 公里 (位於臺灣東南部海域)(臺灣東南部海域)	2024-05-25	15:3
1	小區域有感地震	121.674	24.2543	3.9	28.3	花蓮縣政府北北東方 29.7 公里 (位於花蓮縣秀林鄉)(花蓮縣秀林鄉)	2024-05-25	13:3

```
1 plt.figure(figsize=(10, 5))
2 #x = df.groupby('week').mean()['規模']
3 x = df.groupby([pd.Grouper(key='Date', freq='W')])['規模'].mean()
4
5 x.plot.bar()
6 plt.show()
```

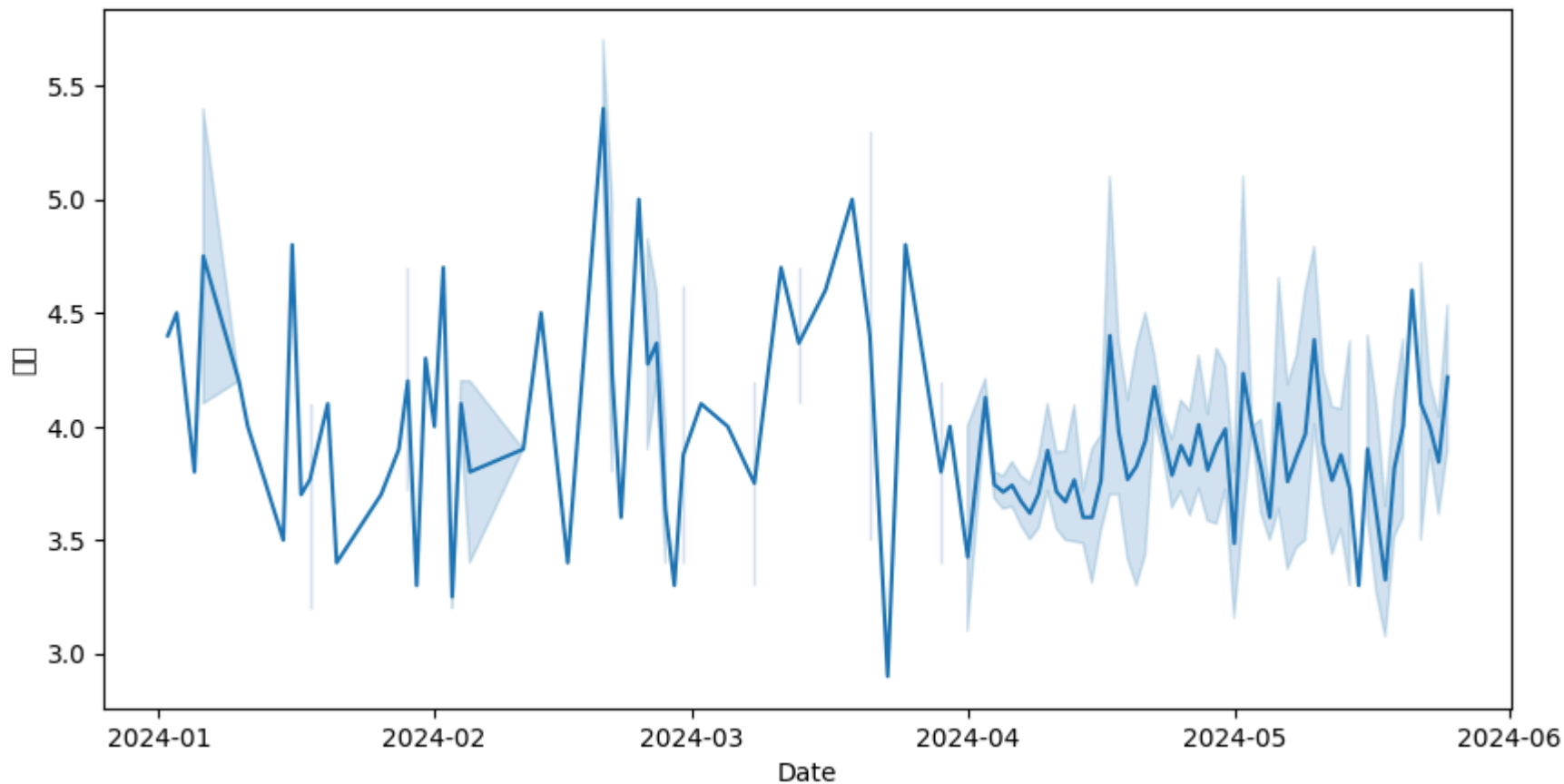


```

1 plt.figure(figsize=(10, 5))
2 sns.lineplot(data=df,
3             x='Date',
4             y='規模')

```

↔ <Axes: xlabel='Date', ylabel='規模'>

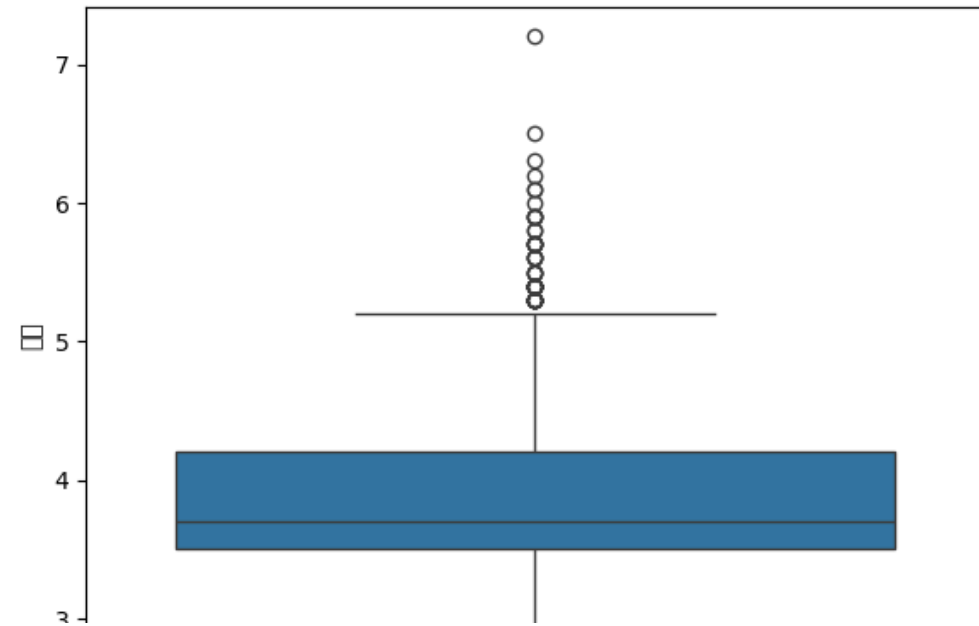
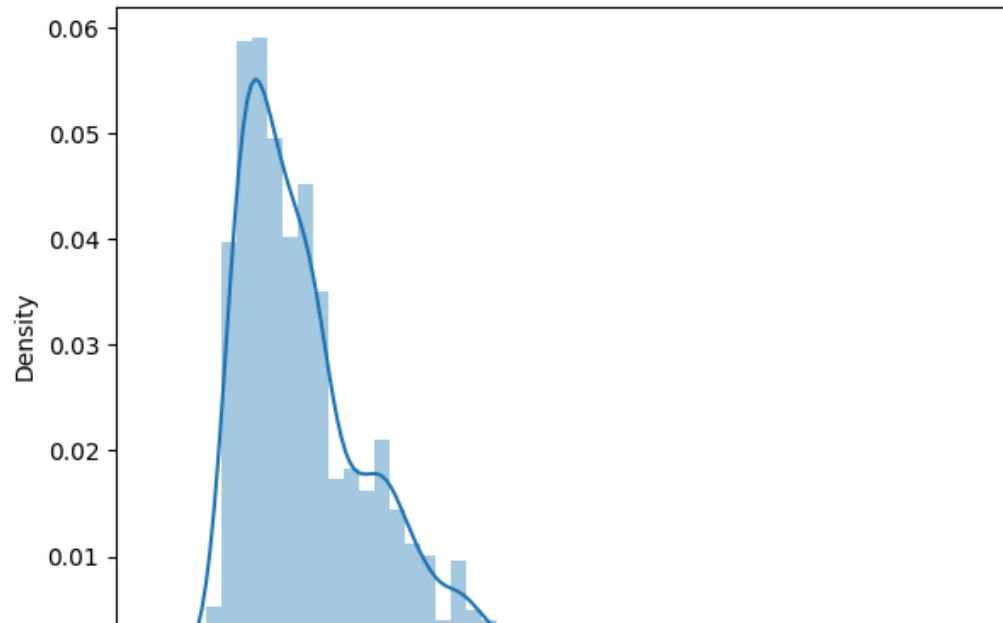


```

1
2 plt.subplots(figsize=(15, 5))
3
4 plt.subplot(1, 2, 1)
5 sns.distplot(df['深度'])
6
7 plt.subplot(1, 2, 2)
8 sns.boxplot(df['規模'])
9

```

⇒ <Axes: ylabel='規模'>

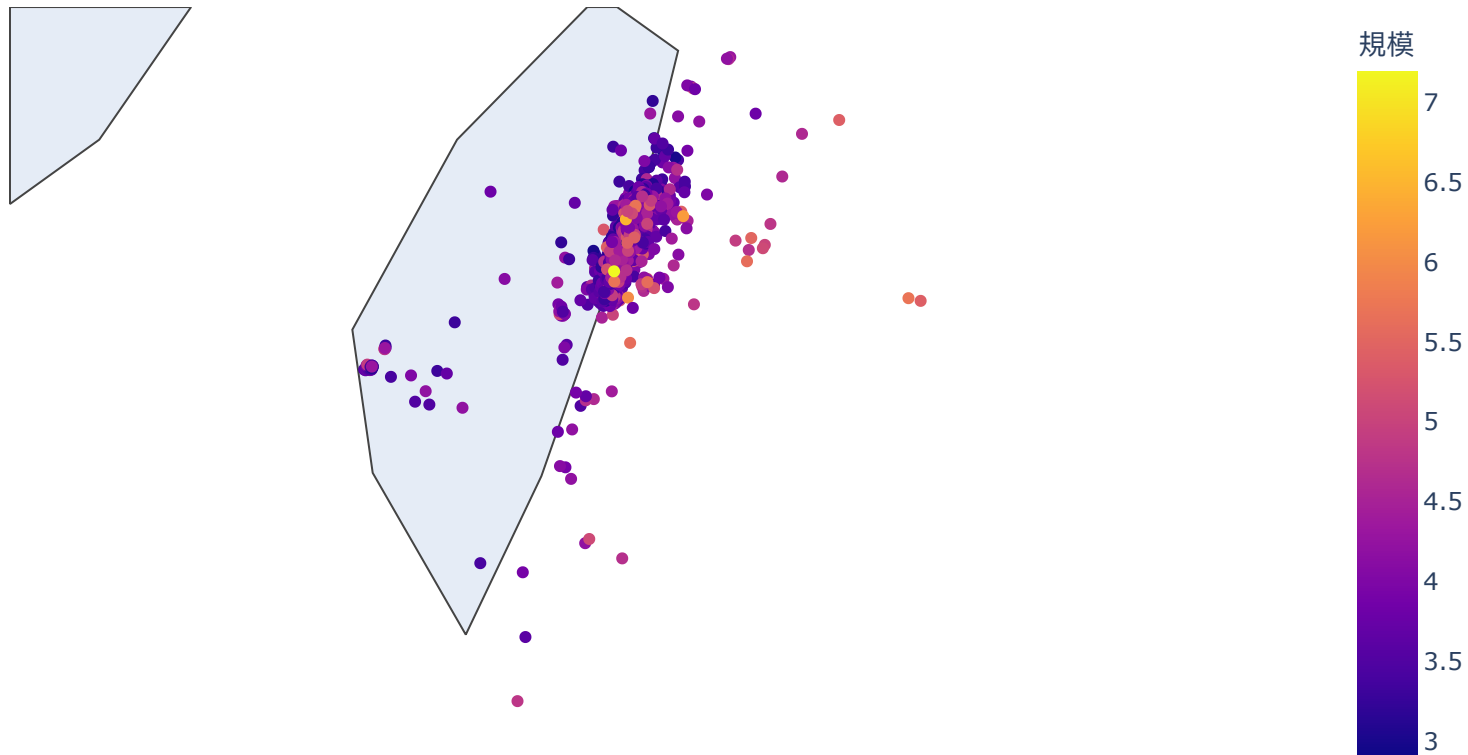


```
1
2 plt.subplots(figsize=(15, 5))
3
4 plt.subplot(1, 2, 1)
5 sb.distplot(df['Magnitude'])
6
7 plt.subplot(1, 2, 2)
8 sb.boxplot(df['Magnitude'])
```

```
1 plt.figure(figsize=(10, 8))
2 sb.scatterplot(data=df,
3               x='Latitude',
4               y='Longitude',
5               hue='Magnitude')
```

```
1 fig = px.scatter_geo(df, lat='緯度',
2                      lon='經度',
3                      color="規模",
4                      fitbounds='locations',
5                      scope='asia')
```

```
1 fig.show()
```



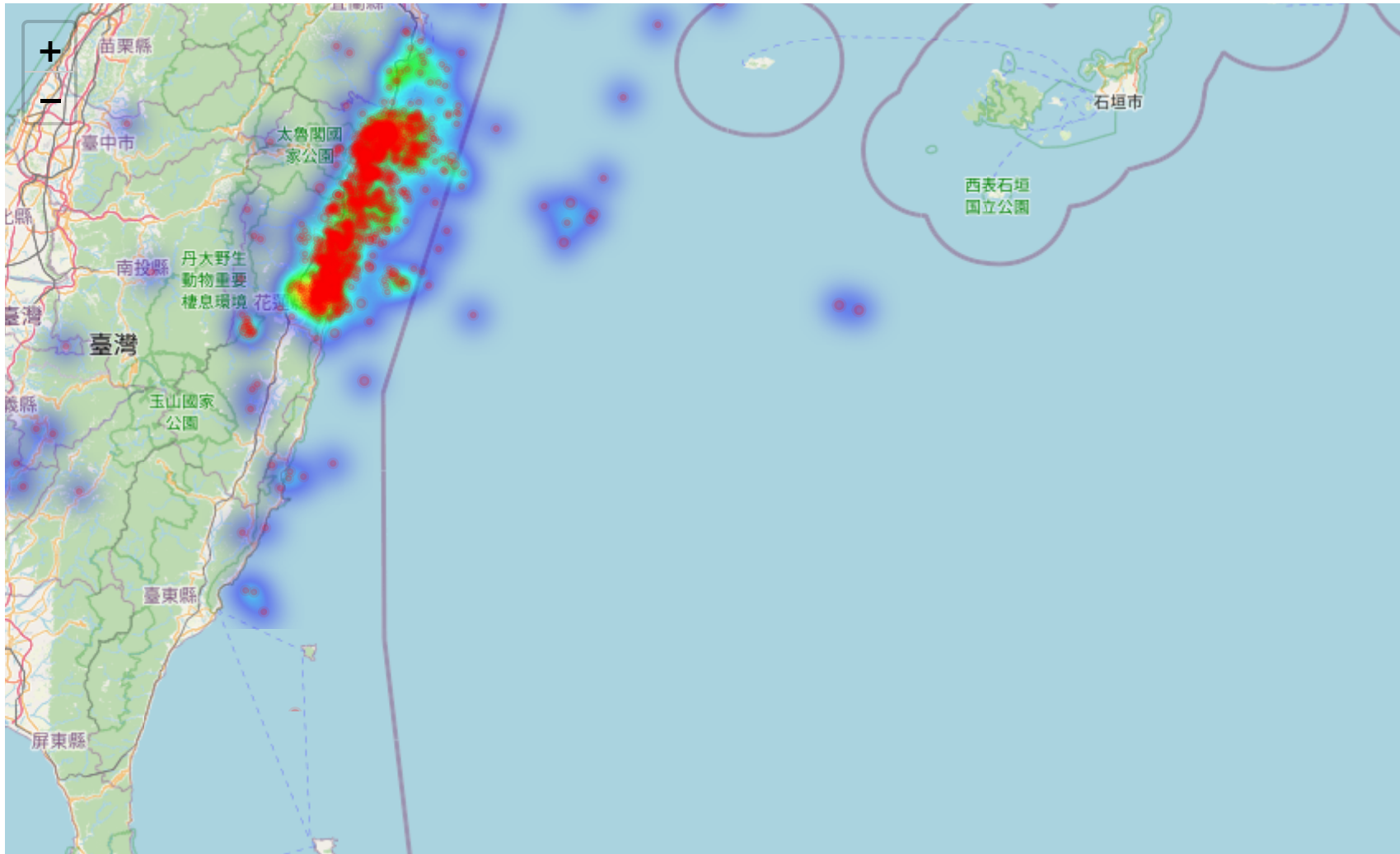
```
1 [lat,lng]=[df['緯度'].mean(),df['經度'].mean()]
```

```
1 tw_coord = [lat,lng]
2 tw_map = folium.Map(location=tw_coord, zoom_start=8)
3 HeatMap(data=df[['緯度', '經度', '規模']], radius=10).add_to(tw_map)
4
5 for index, row in df.iterrows():
6     folium.CircleMarker(
7         location=[row['緯度'], row['經度']],
8         radius=row['規模']/2
```

```

9         radius=1000, 规模=1/2,
10         color='red', opacity=0.1,
11         fill_color='red').add_to(tw_map)
12 tw_map

```



```

1 from folium.plugins import TimestampedGeoJson

```

```

1 df.sort_values(by='Date', inplace=True)
2 df.reset_index(drop=True, inplace=True)
3

```

```
1 df['date']=df['Date'].astype(str)
2 df['date'][0]
```

🔄 '2024-01-02'

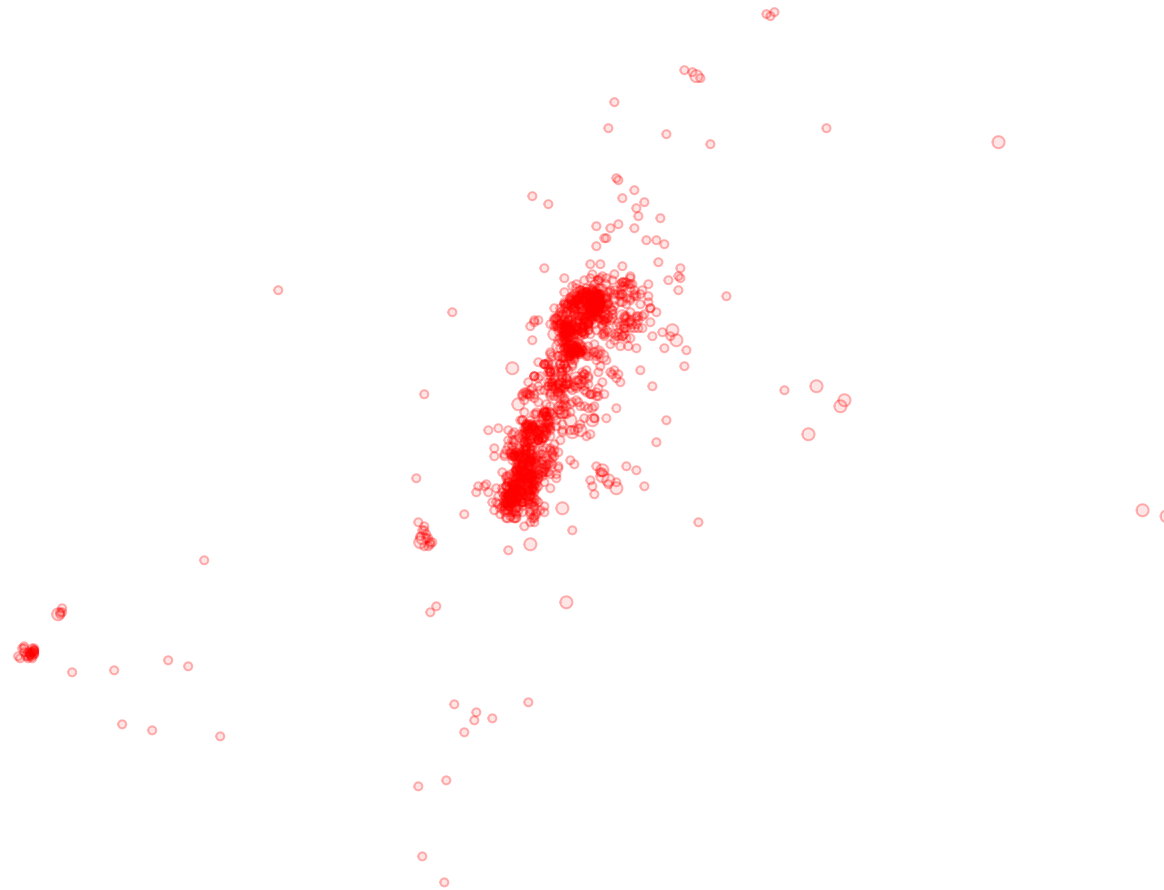
```
1 import datetime
2 def create_geojson_features(df):
3     features = []
4
5     for _, row in df.iterrows():
6         feature = {
7             'type': 'Feature',
8             'geometry': {
9                 'type': 'Point',
10                'coordinates': [row['經度'], row['緯度']]
11            },
12            'properties': {
13                'time': row['date'],
14                'style': {'color': 'red'},
15                'icon': 'circle',
16                'iconstyle': {
17                    #'fillColor': row['fillColor'],
18                    'fillOpacity': 0.1,
19                    'opacity': 0.3,
20                    'weight': 1,
21                    #'stroke': 'true',
22                    'radius': row['規模']/2,
23                    #'radius': row['count'] + 5
24                }
25            }
26        }
27        features.append(feature)
28    return features
```

```
1 df_geojson = create_geojson_features(df)
2 #df_geojson[0]
```



```
1 tw_anim = folium.Map(location = tw_coord,  
2                       titles = "Taiwan EarthQuake 2024",  
3                       zoom_start = 8)  
4  
5 TimestampedGeoJson(df_geojson,  
6                   transition_time = 100,  
7                   #auto_play = True  
8                   ).add_to(tw_anim)  
9 tw_anim
```

↔ Make this Notebook Trusted to load map: File -> Trust Notebook



1 Start coding or [generate](#) with AI.

