

Python應用實作專題：

「用Python視覺化112年1-10月交通事故通報資料」

112403543 資管一 王曄元

題目發想

- 想說做資料視覺化，然後想起好像有看過一個資料集可以拿來用
- 使用的Python套件
 - Pandas
 - Matplotlib

資料來源

■ <https://freeway2024.tw/links#links>



112年1-10月交
通事故簡訊通報
資料



112年1-10月道
路施工路段資料

進行步驟

- 轉檔
- 觀察可以拿來繪圖的數據關係
- 資料處理
- 繪圖

轉檔

■請GPT幫我寫一個可以將Excel轉成CSV的程式

讀取 Excel 檔案

```
excel_file = '112年1-10月道路施工路段資料.xlsx'
```

```
df = pd.read_excel(excel_file)
```

寫入 CSV 檔案

```
csv_file = 'output_csv_file.csv'
```

```
df.to_csv(csv_file, index=False)
```

```
print(f'{excel_file} 已成功轉換成 {csv_file}')
```

觀察數據間的關係

經過觀察覺得可以用以下幾種關係來繪圖：

- 一天內的事故發生次數分佈
- 不同國道次數分佈
- 不同事故類型發生次數分佈
- 每月事故次數分佈
- 排除所花時間次數分佈

程式主要架構

■導入套件

```
import pandas as pd  
import matplotlib.pyplot as plt
```

■解決無法顯示中文問題

```
plt.rcParams['font', family='Microsoft JhengHei']
```

■讀取資料(已將檔名改成accident.csv)

```
accidents_df = pd.read_csv("accident.csv")
```

程式主要架構

■計算

用 `.value_counts()`

例 `accidents_count = accidents_df["時"].value_counts()`

■排序(對於index)(月份之類的)

用 `.sort_index()`

例 `accidents_count = accidents_count.sort_index()`

程式主要架構

■繪圖

■設定圖的大小

```
plt.figure(figsize=(10, 6))
```

■圖的樣式

```
accidents_count.plot(kind="bar", color="skyblue")
```

■圖的標籤

```
plt.title("112年1-10月一天內事故發生分佈長條圖")
```

```
plt.xlabel("時間")
```

```
plt.ylabel("事故次數")
```

程式主要架構

■設定x軸

```
plt.xticks(range(len(accidents_count)), accidents_count.index, rotation=45)
```

■添加數據標籤

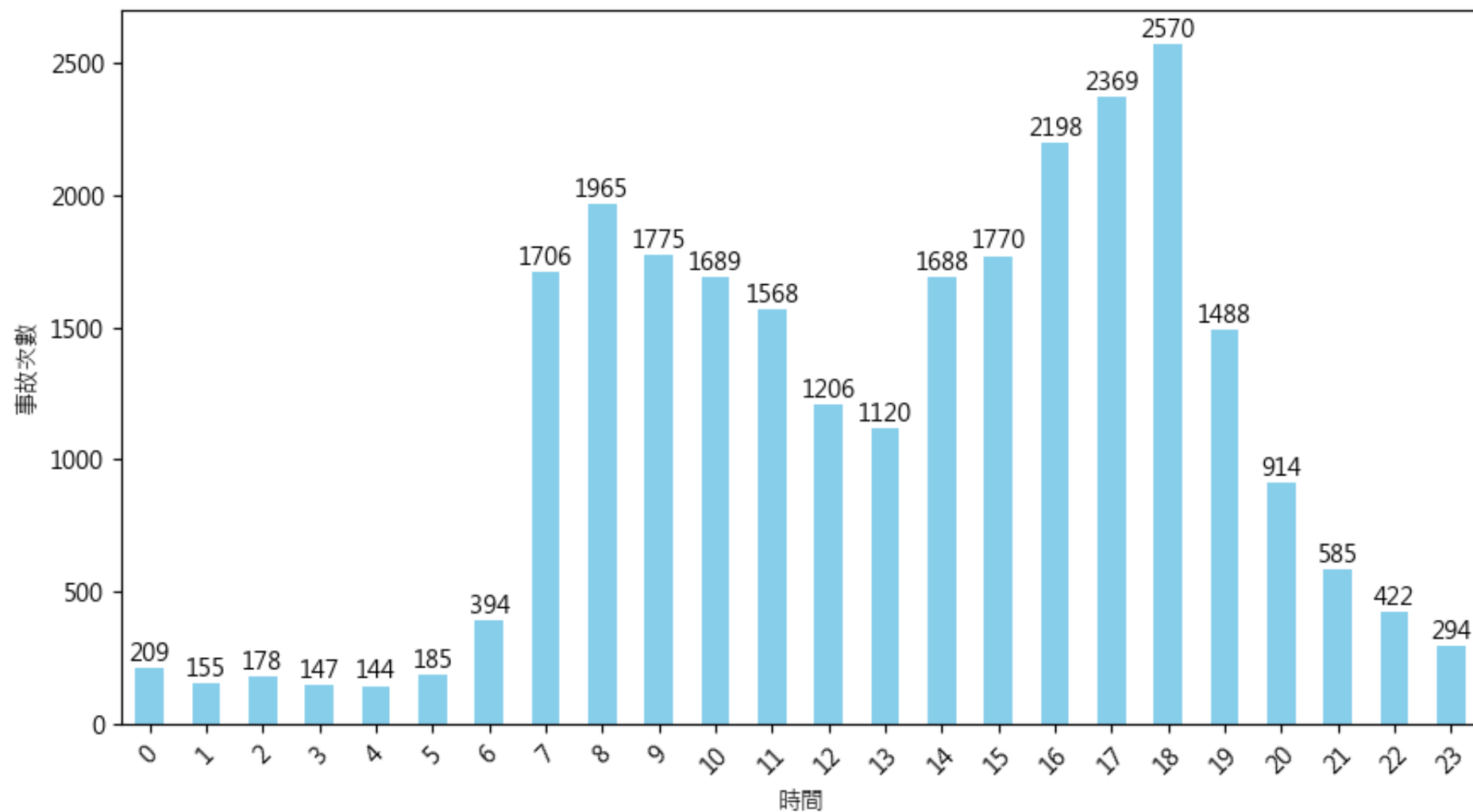
```
for i, v in enumerate(accidents_count):  
    plt.text(i, v + 1, str(v), ha='center', va='bottom')
```

■存檔

```
plt.savefig("112年1-10月一天內事故發生分佈長條圖.png")  
plt.show()
```

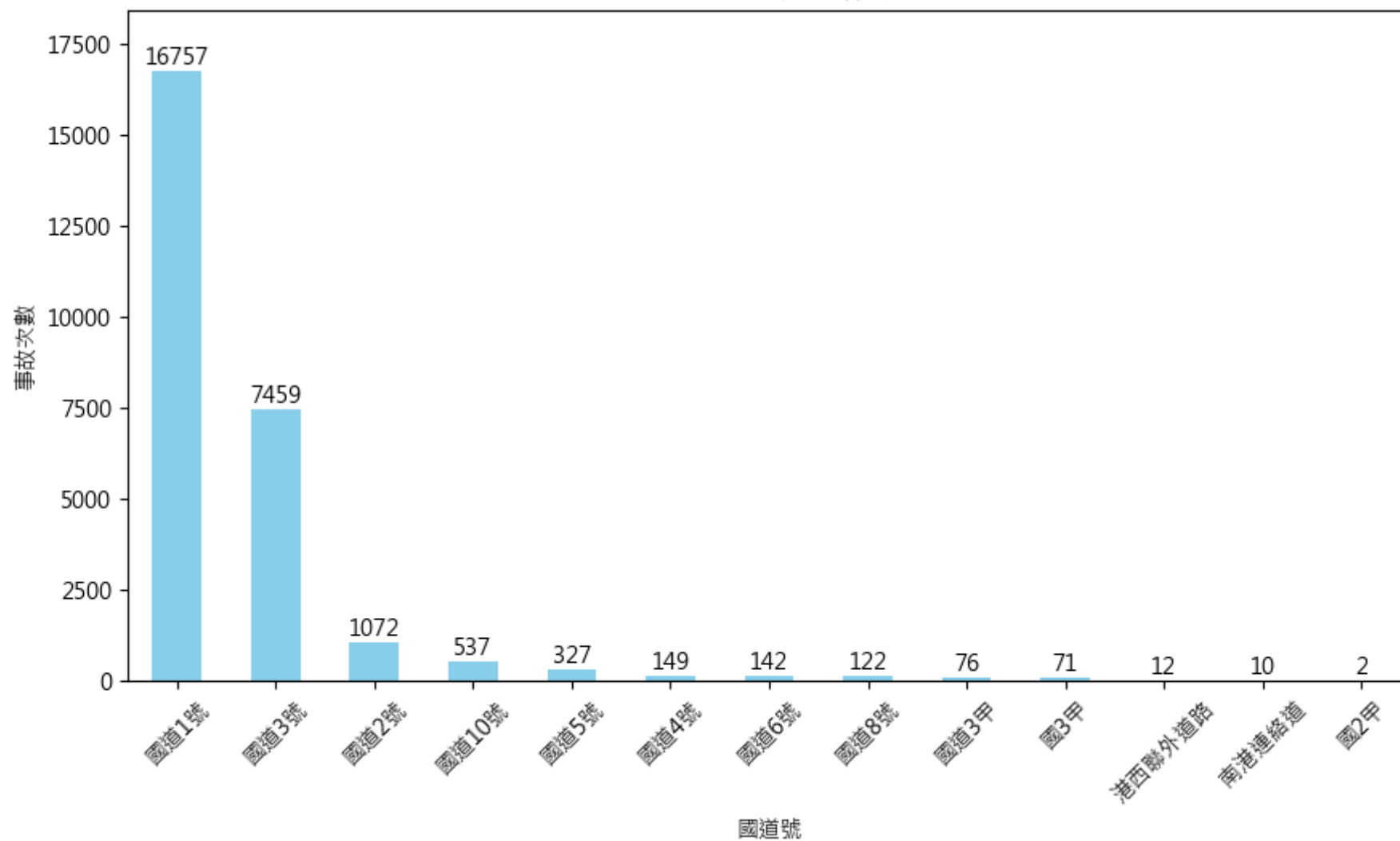
繪製結果

112年1-10月一天內事故發生分佈長條圖



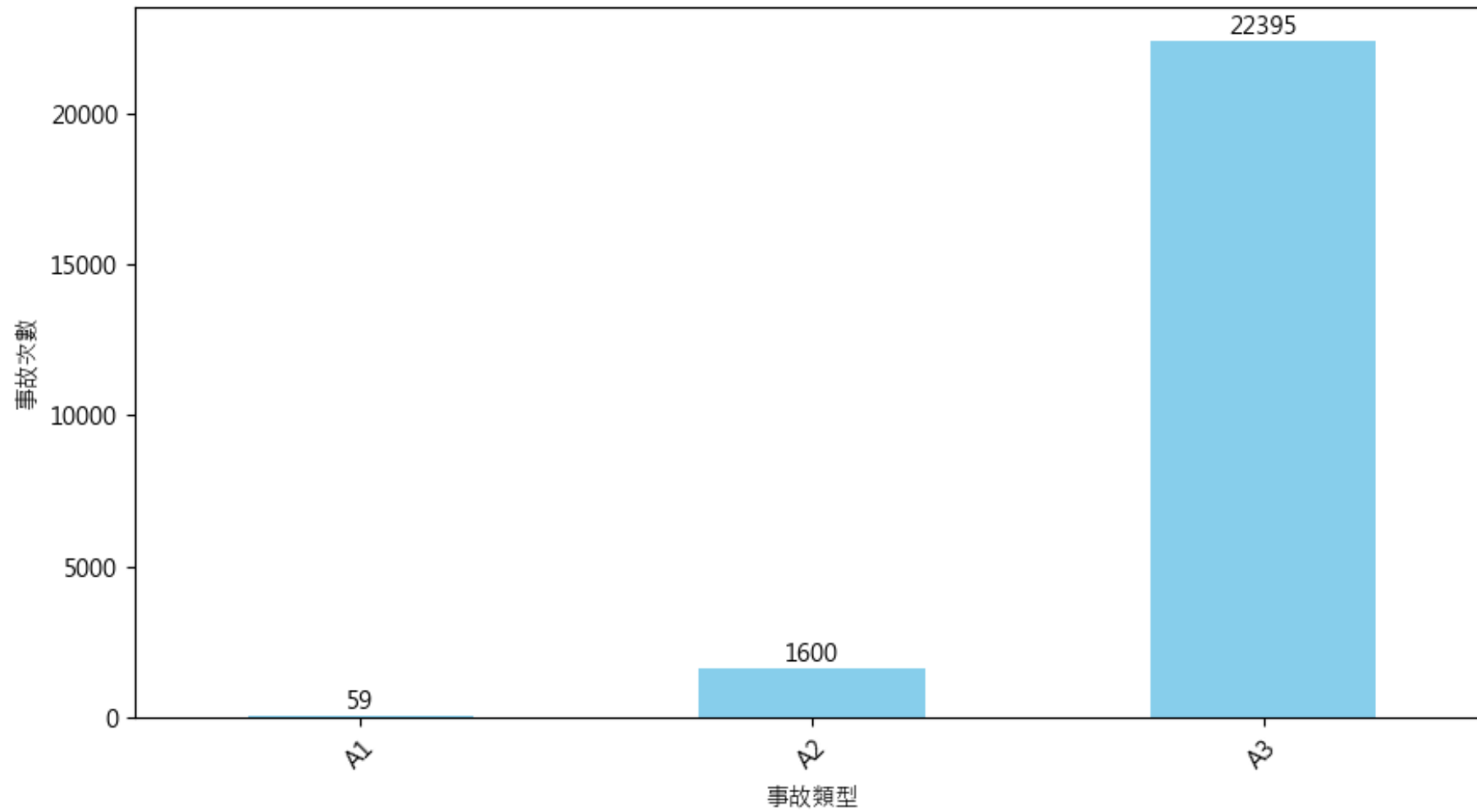
繪製結果

112年1-10月每條國道事故次數長條圖



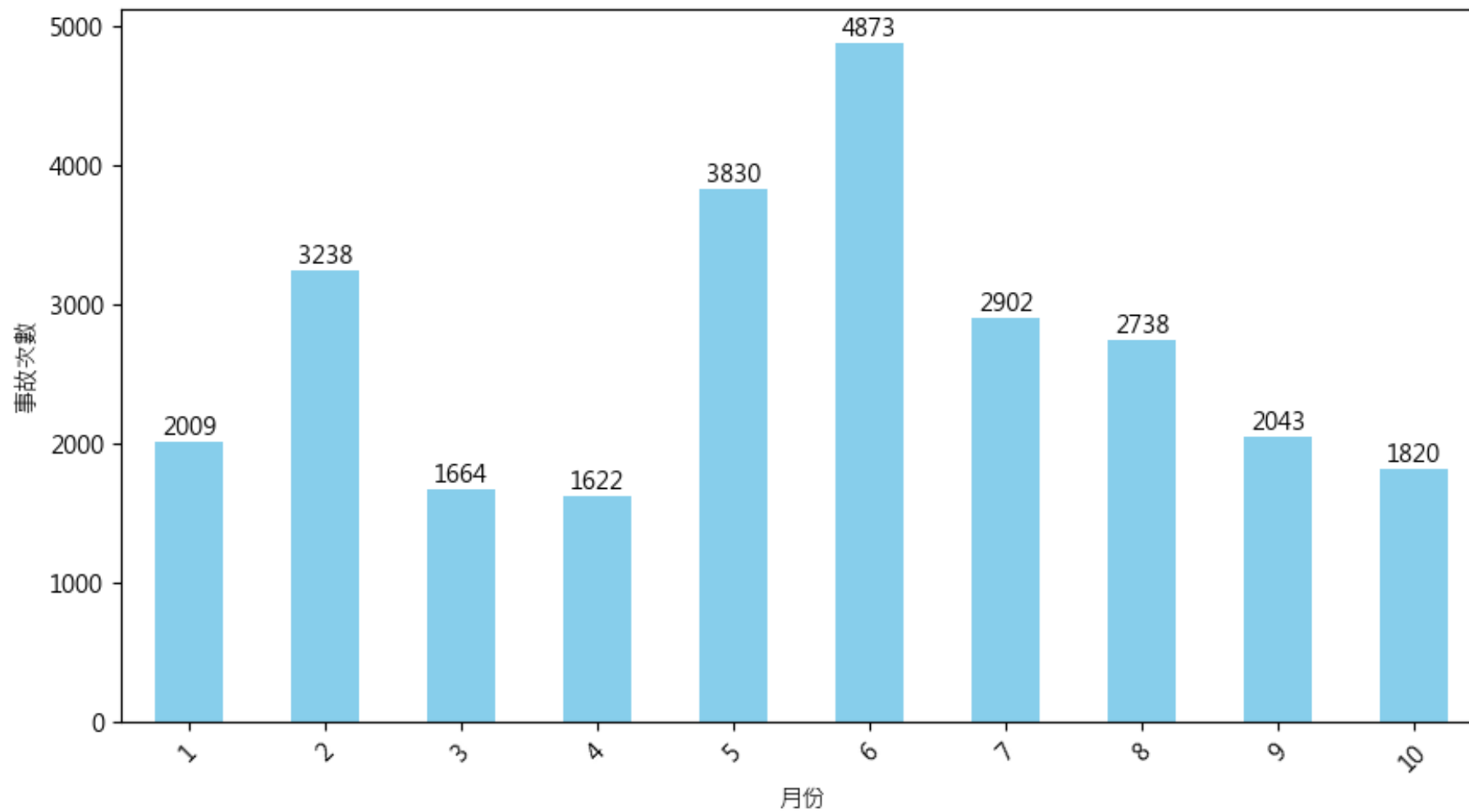
繪製結果

112年1-10月事故類型長條圖



繪製結果

112年1-10月每月事故量長條圖



計算事故排除時間

- 原始資料並沒有排除時間這個column
- 用"事件排除" - "事件發生" 取得事故排除時間
- 處理有些資料格式問題

```
.apply(lambda x: '00:00' if len(x) < 5 else x)
```

- 部分資料的冒號打成分號，改掉

```
.str.replace(';', ':')
```

- 將時間換算成分鐘數

```
.apply(lambda x: int(x.split(':')[0]) * 60 + int(x.split(':')[1]))
```

計算事故排除時間

- 假設：所有事故都是在24小時內排除
- 如果排除時間比發生時間還小，表示有跨夜，額外加上一天

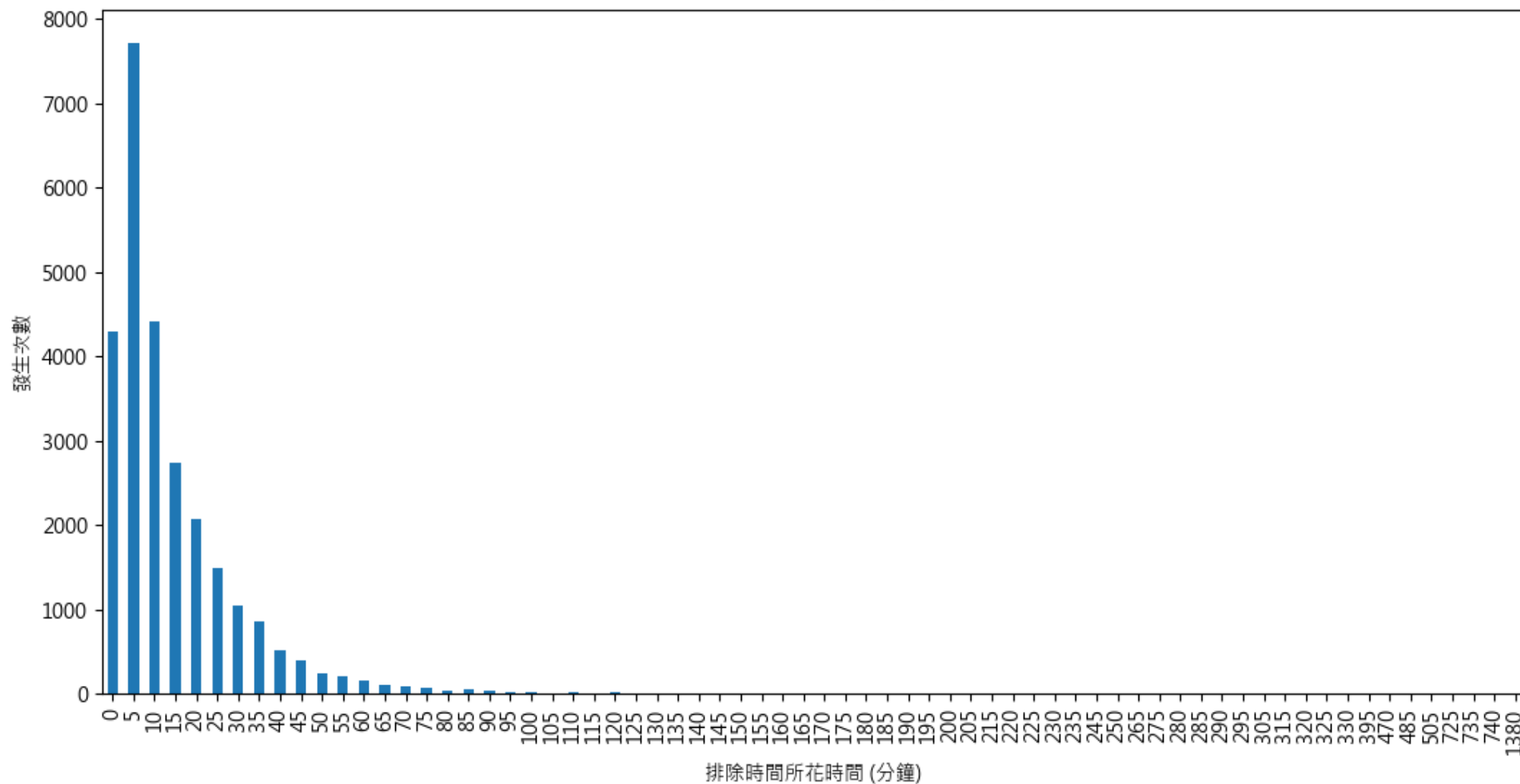
```
data['事件排除所花時間'] = data['事件排除'] - data['事件發生']  
data.loc[data['事件排除所花時間'] < 0, '事件排除所花時間'] += 24 * 60
```

- 將每五分鐘當作一個間隔繪圖
- 100分鐘以後的次數較少，將超過一百的再拉出來繪圖

```
.apply(lambda x: 100 if x > 100 else x // 5 * 5)
```


繪圖結果

排除時間所花時間與發生次數



繪圖結果

排除時間所花時間與發生次數(5分鐘為間隔)(100分鐘以上合併)

