

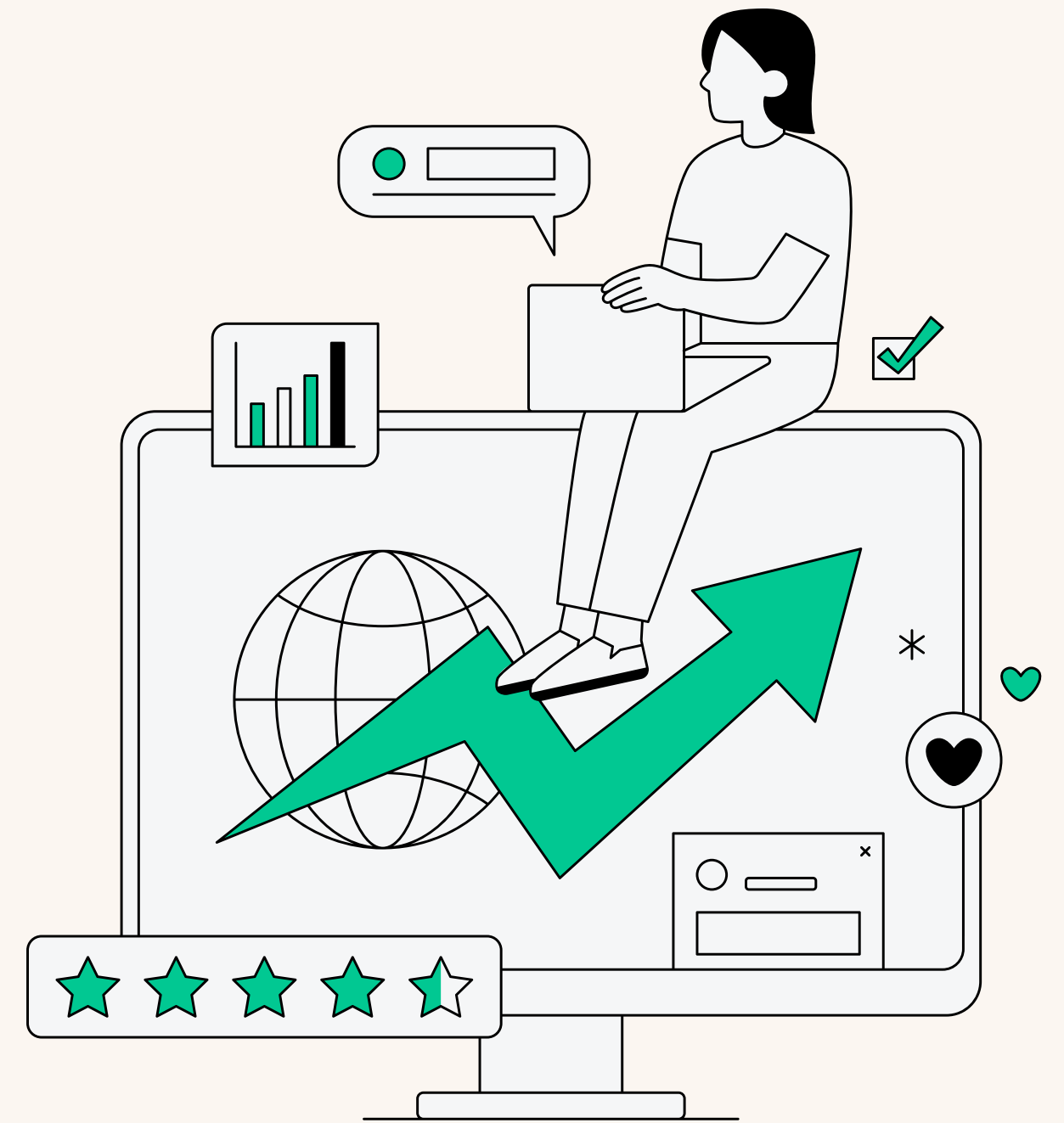
# 道路交通事故調查 資料分析

第13組

111701017 楊守荃

111701030 程品諭

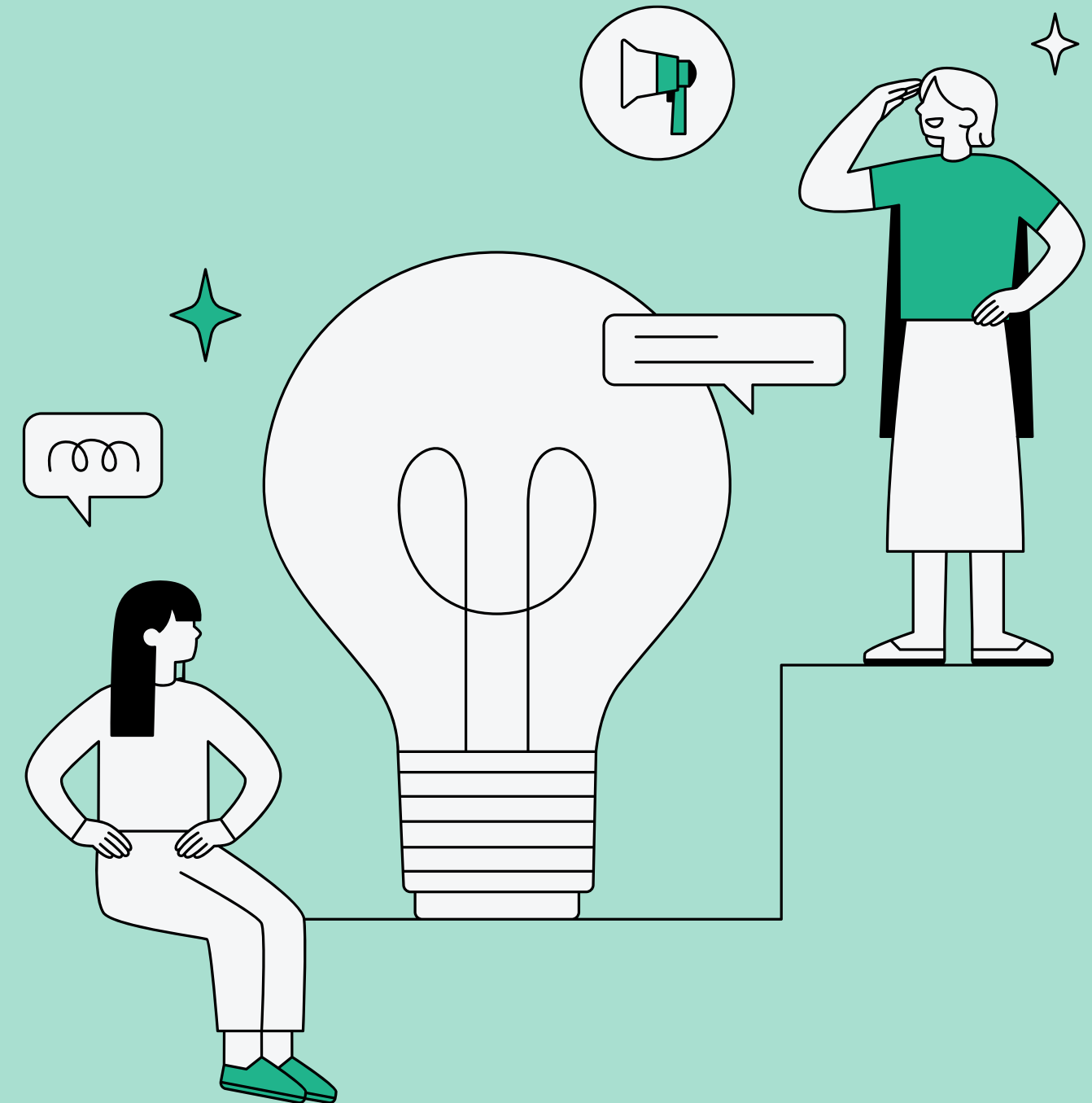
111701031 黃浚瑀



# 所選資料

## 道路交通事故調查報告表一表二及30日 死亡清單

<https://www.motc.gov.tw/uploaddowndoc?file=datagov/1118823766994980864.csv&filedisplay=106-111%E5%B9%B4%E9%81%93%E8%B7%AF%E4%BA%A4%E9%80%9A%E4%BA%8B%E6%95%85%E8%AA%BF%E6%9F%A5%E5%A0%B1%E5%91%8A%E8%A1%A8%E4%B8%80%E8%A1%A8%E4%BA%8C%E5%8F%8A30%E6%97%A5%E6%AD%BB%E4%BA%A1%E6%B8%85%E5%96%AE2.csv&flag=doc>





# 資料處理過程

## 01.讀取資料程式碼及結果



#讀取資料資訊

```
import pandas as pd
from requests import get
import io
url = 'https://www.motc.gov.tw/uploaddowndoc?file=datagov/'
response = get(url)
data = pd.read_csv(io.StringIO(response.text))
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17101 entries, 0 to 17100
Data columns (total 35 columns):
#   Column                Non-Null Count  Dtype
---  -
0   權責單位              17101 non-null  object
1   年度                  17101 non-null  int64
2   發生日期              17101 non-null  int64
3   d_date                17101 non-null  object
4   死因                  17101 non-null  int64
5   天數                  17101 non-null  int64
6   國籍                  17101 non-null  int64
7   SEX                   17101 non-null  int64
8   發生縣市              17101 non-null  object
9   市區鄉鎮              17101 non-null  object
10  飲酒情形              17101 non-null  int64
11  當事者區分            17101 non-null  object
12  肇因（主要）          17101 non-null  int64
13  肇因（個別）          17101 non-null  int64
14  道路類別              17101 non-null  int64
15  事故位置              17101 non-null  int64
16  事故類型及型態        17101 non-null  int64
17  死亡                  17101 non-null  int64
18  受傷                  17101 non-null  int64
19  2-30                  17101 non-null  int64
20  旅次目的              17101 non-null  int64
21  職業                  17101 non-null  int64
22  肇事逃逸              17101 non-null  int64
23  受傷程度              17101 non-null  int64
24  當事者序號            17101 non-null  int64
25  age                   17100 non-null  float64
26  filename               17101 non-null  object
```

# 資料處理過程

## 02.資料清洗程式碼及結果

```
[ ] #資料清洗:將肇事逃逸那欄的「1」變為「沒有肇逃」，「2」變為「有肇逃」
data['肇事逃逸'] = data['肇事逃逸'].replace({1: '沒有肇逃', 2: '有肇逃'})
print (data['肇事逃逸'].value_counts())
#將sex那欄的「1」變為「男」，「2」變為「女」
data['SEX'] = data['SEX'].replace({1: '男', 2: '女'})
#將道路類別、事故類型及型態、飲酒情形、事故位置這四欄，也依照對照表進行轉換
data['道路類別'] = data['道路類別'].replace({1: '國道', 2: '省道', 3:'縣道', 4:'鄉道'})
data['事故類型及型態'] = data['事故類型及型態'].replace({1: '對向通行中', 2: '同向通行中'})
data['飲酒情形'] = data['飲酒情形'].replace({1: '經觀察未飲酒', 2: '經檢測無酒精反應'})
data['事故位置'] = data['事故位置'].replace({1: '交岔路口內', 2: '交岔口附近', 3: '機'})
print (data)
```

肇事逃逸  
沒有肇逃 17069  
有肇逃 32  
Name: count, dtype: int64

	權責單位	年度	發生日期	d_date	死因	天數	國籍	SEX	發生縣市	市區鄉鎮	...	age	\
0	新北市	106	1060101	2017/1/11	1 10	1	男	新北市	平溪區	...	19.0		
1	高雄市	106	1060101	2017/1/1	1 0	1	男	高雄市	前金區	...	87.0		
2	桃園市	106	1060101	2017/1/1	1 0	1	男	桃園市	大溪區	...	15.0		
3	桃園市	106	1060101	2017/1/1	1 0	1	男	桃園市	大溪區	...	16.0		
4	新北市	106	1060101	2017/1/12	1 11	1	男	新北市	三重區	...	57.0		
...	...	...	...	...	...	...	...	...	...	...	...	...	...
17096	彰化縣	111	1111130	2022/11/30	7 0	1	男	彰化縣	溪湖鎮	...	24.0		
17097	臺南市	111	1111130	2022/11/30	8 0	1	女	臺南市	東區	...	67.0		
17098	雲林縣	111	1111130	2022/11/30	8 0	1	女	雲林縣	西螺鎮	...	74.0		
17099	宜蘭縣	111	1111130	2022/12/2	8 2	1	男	宜蘭縣	羅東鎮	...	47.0		
17100	臺南市	111	1111130	2022/11/30	8 0	1	女	臺南市	歸仁區	...	59.0		

	filename	主要肇事因素	處理單位代碼	單位簡稱	公路	公里	公尺	車道方向	車道位置
0	A2	22	ADD91	瑞芳分局瑞芳交通分隊	NaN	0.0	0.0	NaN	NaN
1	A1	43	AV691	新興分局交通分隊	NaN	0.0	0.0	NaN	NaN
2	A1	6	AE391	大溪分局大溪交通分隊	NaN	0.0	0.0	NaN	NaN
3	A1	6	AE391	大溪分局大溪交通分隊	NaN	0.0	0.0	NaN	NaN
4	A2	23	AD491	三重分局交通隊	NaN	0.0	0.0	NaN	NaN
...	...	...	...	...	...	...	...	...	...
17096	A1	23	BJ691	溪湖分局交通分隊	NaN	0.0	0.0	NaN	其他
17097	A1	15	AC191	第一分局交通分隊	NaN	NaN	NaN	NaN	NaN
17098	A1	9	BL591	西螺分局交通分隊	NaN	NaN	NaN	NaN	NaN
17099	A2	23	BT4U1	開羅派出所	NaN	NaN	NaN	NaN	NaN
17100	A1	25	ACA91	歸仁分局交通分隊	NaN	NaN	NaN	NaN	NaN

# 資料處理過程

## 03.將主要肇事因素轉換為中文

由於檔案中主要肇事因素是以數字表示，因此需先將他換成以中文標題的形式。

```
data['主要肇事因素'] = data['主要肇事因素'].replace({
1: '違規超車',
2: '爭(搶)道行駛',
3: '蛇行、方向不定',
4: '逆向行駛',
5: '未靠右行駛',
6: '未依規定讓車',
7: '變換車道或方向不當',
8: '左轉彎未依規定',
9: '右轉彎未依規定',
10: '迴轉未依規定',
11: '橫越道路不慎',
12: '倒車未依規定',
13: '超速失控',
14: '未依規定減速',
15: '搶越行人穿越道',
16: '未保持行車安全距離',
17: '未保持行車安全間隔',
18: '停車操作時，未注意其他車 (人) 安全',
19: '起步未注意其他車(人) 安全',
20: '吸食違禁物後駕駛失控',
21: '酒醉(後)駕駛失控',
22: '疲勞(患病)駕駛失控',
23: '未注意車前狀態'
```



# 資料視覺化



# 01. 製作圓餅圖

我們將飲酒情形  
以圓餅圖的方式  
呈現出來。

```
#畫飲酒情形圓餅圖
!pip install brewer2mpl
import brewer2mpl
import matplotlib.pyplot as plt
import matplotlib as mpl
from matplotlib.font_manager import fontManager
!wget -O TaipeiSansTCBeta-Regular.ttf https://drive.google.com/uc?id=1eGAsTN1HBpJAKEVM57_C7ccp7hbgSz3_&export=download
fontManager.addfont('TaipeiSansTCBeta-Regular.ttf')
mpl.rc('font', family='Taipei Sans TC Beta')

bmap = brewer2mpl.get_map('Set3', 'qualitative', 11)
colors = bmap.mpl_colors

def my_autopct(pct):
    return '{:.1f}%'.format(pct)

fig, ax = plt.subplots(figsize=(20, 10), subplot_kw=dict(aspect="equal"))

drink = data['飲酒情形'].value_counts()
wedges, colors, autopct = ax.pie(drink, colors=colors, autopct='%1f%%', wedgeprops=dict(width=0.5))

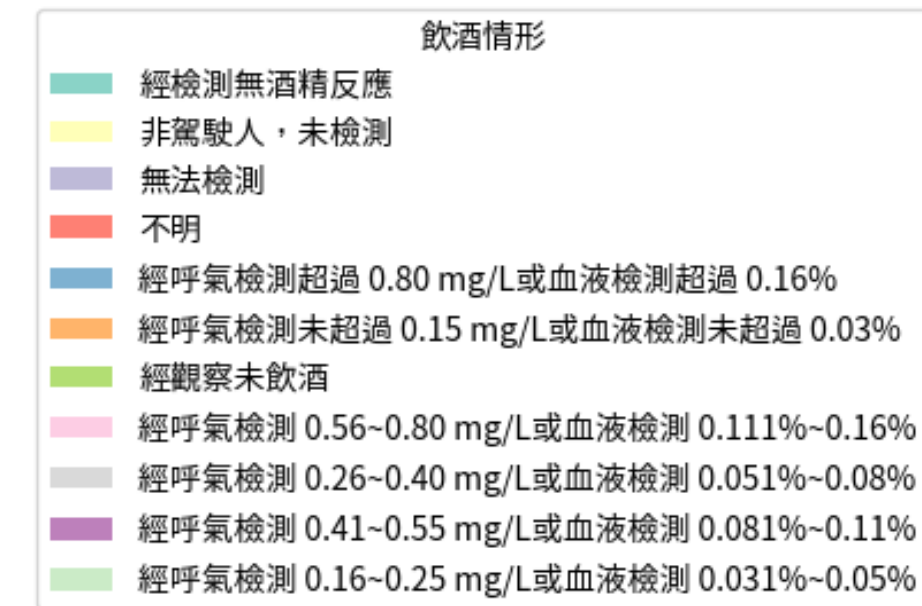
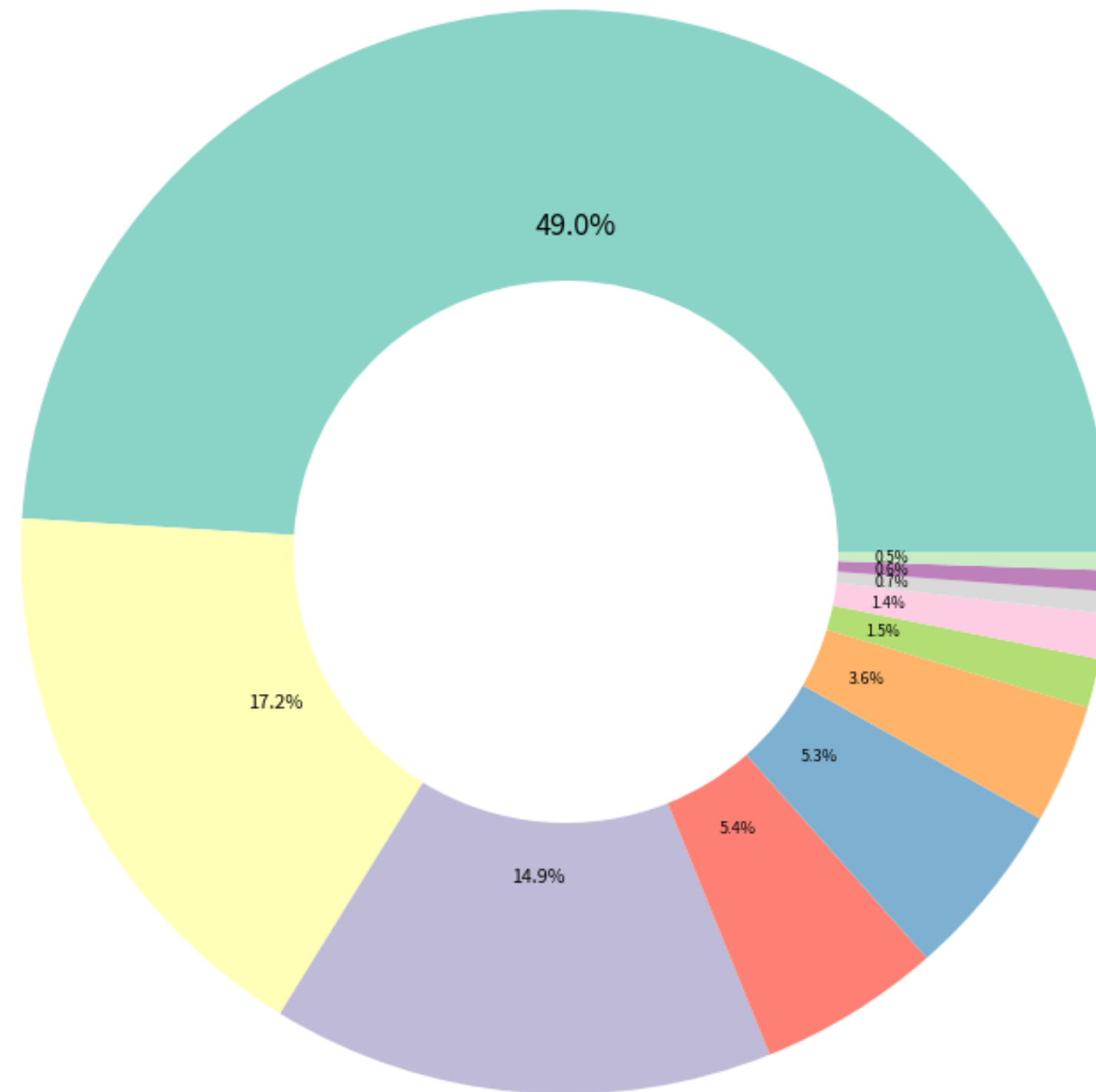
for autotext, pct in zip(autopct, drink):
    autotext.set_fontsize(6 + 0.1 * pct / 150)

ax.legend(drink.index, title='飲酒情形', loc='center left', bbox_to_anchor=(1.0, 0.5))
plt.title("飲酒情形圓餅圖", fontsize=30, loc="center")

plt.show()
```



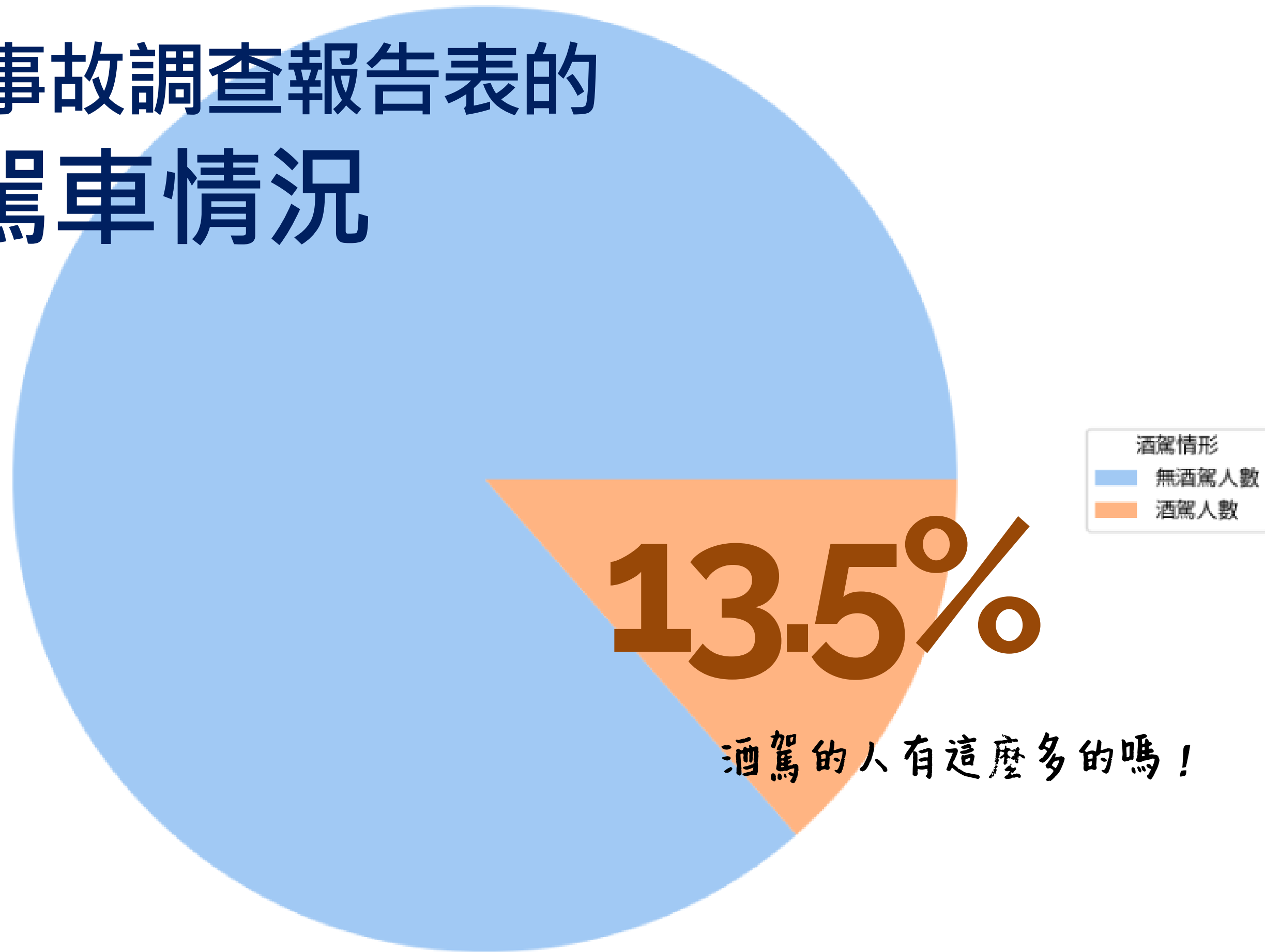
飲酒情形圓餅圖



道路事故交通  
調查中，  
飲酒情形細項  
的分佈情況



# 道路交通事故調查報告表的 酒後駕車情況



## 近年交通事故 酒駕佔比

年別	交通事故 總件數	案件主要肇因為 酒駕之事故件數	比例
103年	307,842	7,513	2.44%
104年	305,413	6,658	2.18%
105年	305,556	5,695	1.86%
106年	296,826	5,039	1.70%
107年	320,315	4,652	1.45%
108年	341,972	4,212	1.23%
109年	362,393	4,224	1.17%
110年 01-10月	288,274	3,308	1.15%

資料產製時間：1月12日  
資料來源：交通部  
整理製表：NOWnews



“ 道路交通事故調查報告表的  
資料  
應是

挑選高傷亡案例 ”



## 02. 製作長條圖

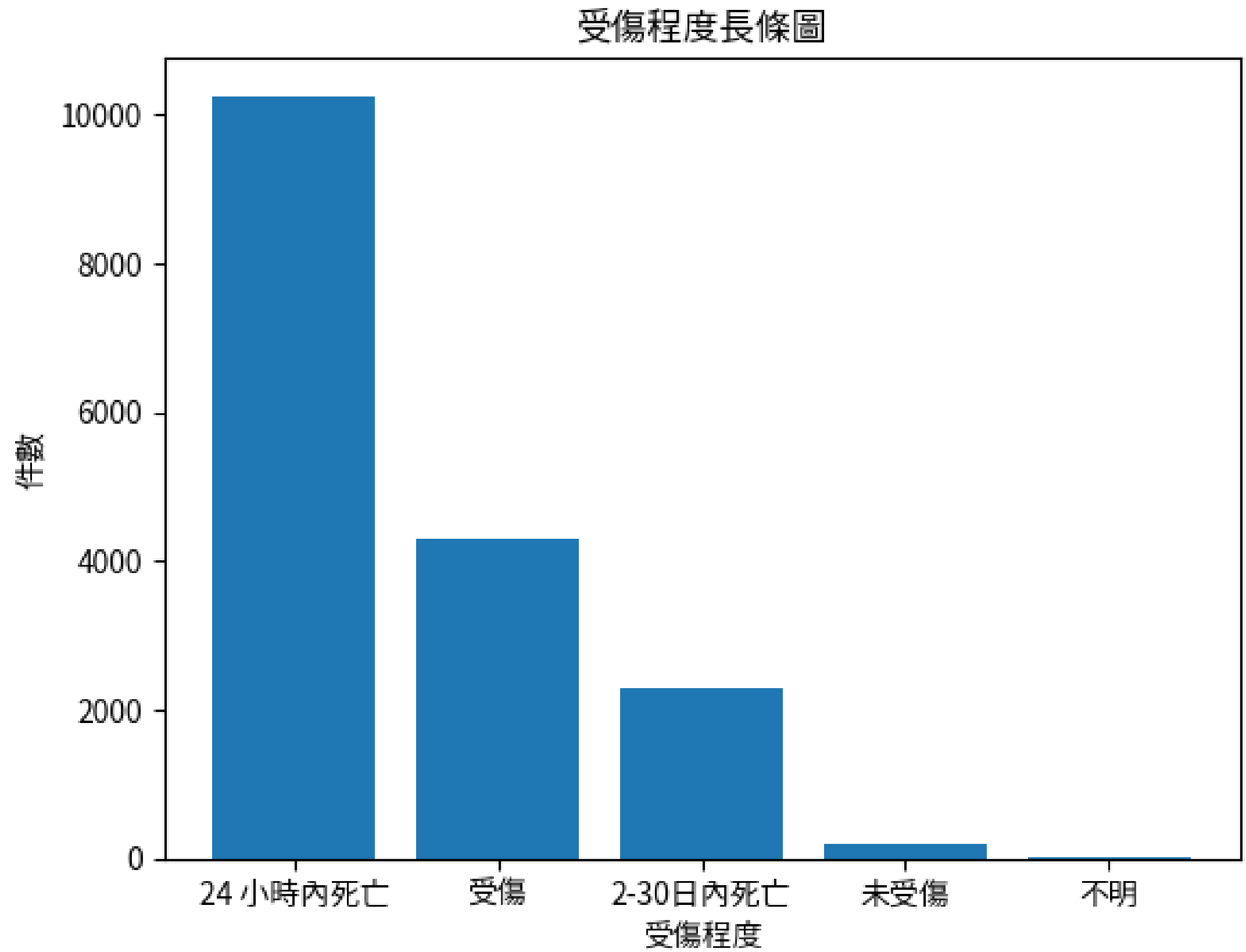
我們將受傷程度  
以長條圖的方式  
呈現出來。

```
[ ] #畫受傷程度長條圖
data['受傷程度'] = data['受傷程度'].replace({1: '24 小時內死亡', 2: '受傷', 3: '未受傷', 4: '不明', 5: '2-30日內死亡'})
hurt = data['受傷程度'].value_counts()
plt.bar (hurt.index, hurt)
plt.title ('受傷程度長條圖')
plt.xlabel ('受傷程度')
plt.ylabel ('件數')
plt.show()
```



# 受傷程度

- 在資料內所有案件中，傷者在24小時內死亡的案件高達10000多件，所佔的比例最高



# 03. 製作散布圖

我們將年齡分布  
以散布圖的方式  
呈現出來。



#畫年齡分布散佈圖

```
age = data['age'].value_counts().sort_index(ascending=False)
```

#按照年齡降序排序

```
plt.scatter (age.index, age)
```

```
plt.xlabel('age')
```

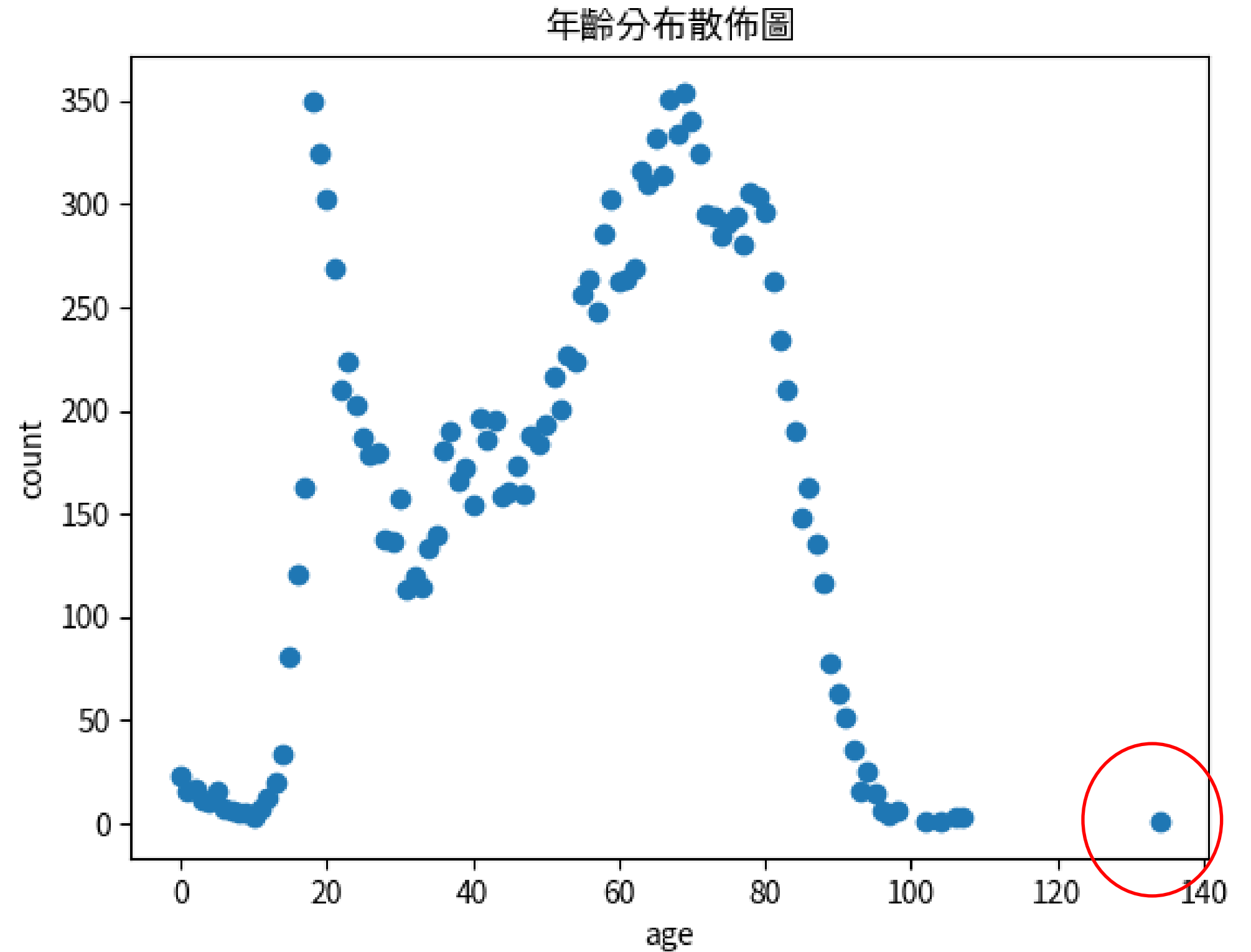
```
plt.ylabel('count')
```

```
plt.title ('年齡分布散佈圖')
```

```
plt.show()
```

# 年齡分布

- 這張散佈圖描述了各年齡層的案件數量。
- 我們也發現了有一個超高齡駕駛出車禍(紅色圓圈圈起來處)，可能是資料輸入有誤導致這種特別奇怪的結果。



# 04. 製作折線圖

我們將每年度的案件數量以折線圖的方式呈現出來。

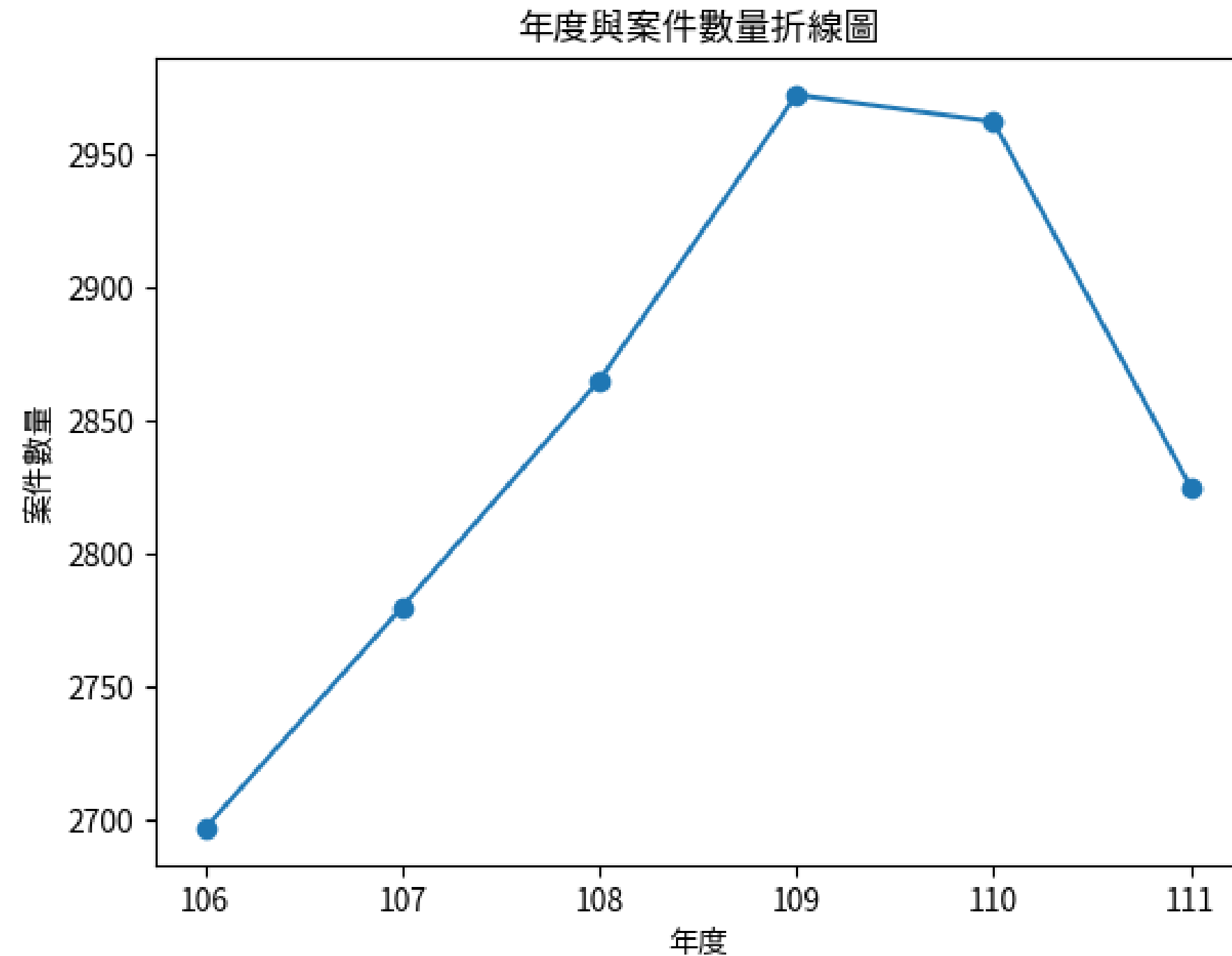


```
data_year = data['年度'].value_counts().sort_index(ascending=True)
plt.plot (data_year, marker = 'o')
plt.title ('年度與案件數量折線圖')
plt.xlabel ('年度')
plt.ylabel ('案件數量')
plt.show()
```



## 各年度案件數量

- 這個折線圖表示了各年度所發生的車禍案件數量，由此可知在106~109年所發生的車禍案件數量成上漲趨勢，而在109年之後則呈現下降的趨勢。



# 05. 製作折線圖

我們將案件數量與受傷程度以折線圖的方式呈現出來，同時以不同顏色的線表示為不同年份。

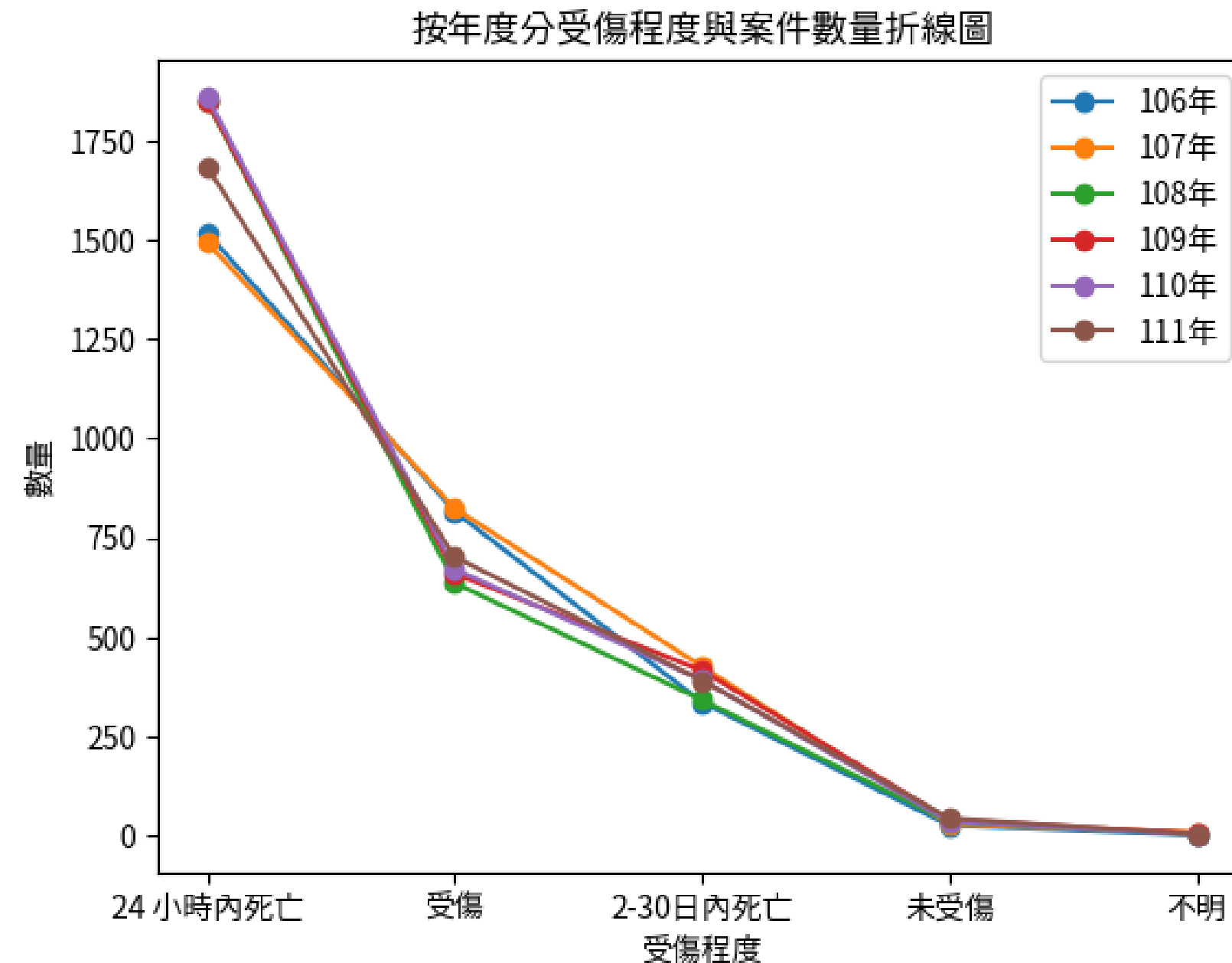
```
[11] data['受傷程度'] = data['受傷程度'].replace({1: '24 小時內死亡', 2: '受傷', 3: '未受傷', 4: '不明', 5: '2-30日內死亡'})

data_106 = data[(data['年度'] == 106)]
data_107 = data[(data['年度'] == 107)]
data_108 = data[(data['年度'] == 108)]
data_109 = data[(data['年度'] == 109)]
data_110 = data[(data['年度'] == 110)]
data_111 = data[(data['年度'] == 111)]

plt.plot (data_106['受傷程度'].value_counts().index, data_106['受傷程度'].value_counts(), marker = 'o')
plt.plot (data_107['受傷程度'].value_counts().index, data_107['受傷程度'].value_counts(), marker = 'o')
plt.plot (data_108['受傷程度'].value_counts().index, data_108['受傷程度'].value_counts(), marker = 'o')
plt.plot (data_109['受傷程度'].value_counts().index, data_109['受傷程度'].value_counts(), marker = 'o')
plt.plot (data_110['受傷程度'].value_counts().index, data_110['受傷程度'].value_counts(), marker = 'o')
plt.plot (data_111['受傷程度'].value_counts().index, data_111['受傷程度'].value_counts(), marker = 'o')
plt.title ('按年度分受傷程度與案件數量折線圖')
plt.xlabel ('受傷程度')
plt.ylabel ('數量')
plt.legend (['106年', '107年', '108年', '109年', '110年', '111年'])
plt.show()
```


## 各年度案件數量

- 由此折線圖可知，不論案件在哪一年發生，有人在24小時內死亡的比例都是最高的，而未有人受傷及不明的案件都佔總案件的少數。





過程中遇到的困難





# 01.

我們在使用Colab時有遇到一些bug，例如一開始輸入`plt.xlabel`時會顯示函式庫沒有此指令，但我們按下「重起執行階段」後，同樣的程式碼又能運作了

# 02.

在爬取附件的資料時，我們發現有些不具有ssl憑證的網站無法透過`requests`來讀取資料，因此我們將`get`函數中的`verify`設為`false`。古怪的是，“縣市即時路況”可以使用vs code抓取，卻不能在colab運行，推測是防火牆擋住了

# 03.

使用pandas中一些功能時(如`to_frame`函式)，即使已輸入`!pip install pip/pandas`以更新這兩項套件了，仍然無法使用某些語法，執行後都會跑出錯誤訊息:沒有這種語法



心得



# 心得與反思

在一開始做這份作業時，我們在資料的處理上有遇到些許困難，但在大家的努力下，這些困難都一一解決了，因此，組員們都一致認為這份作業提升了我們資料處理以及分析的能力，在分析完結果並將圖表畫出來後，我們也發現了國內許多交通問題的現狀及問題，例如：近幾年的車禍案件數量並沒有顯著的下降，我覺得政府可以加強大眾之間的宣導，推廣交通安全教育，相信國內交通事故案例一定能有顯著降低，減少人員傷亡。





# 分工表





# 黃浚瑀

讀取資料  
資料清洗  
製作圖表

# 程品諭

讀取資料  
製作圖表  
分析圖表提出結果

# 楊守荃

製作簡報  
製作影片  
分析圖表提出結果



Colab 連結



## 第十三組\_colab連結

- <https://colab.research.google.com/drive/1Byplt32hHCES40cwrRG0VqtUrzplQ6gx>

Thank  
you very  
much!

