

Calob Link: [https://colab.research.google.com/drive/1HzV0vJUyib2mI20zrNsAwT0ge\\_aqTmh?usp=sharing](https://colab.research.google.com/drive/1HzV0vJUyib2mI20zrNsAwT0ge_aqTmh?usp=sharing)

109年桃園市交通事故資料表-含第二當事人 資料表

table_109 = Table().read_table('https://data.tycg.gov.tw/opendata/datalist/datasetMeta/download?id=77c34235-6fd9-4c1f-8ebd-1f675cdc7c02&rid=3fa9285a-b5f5-4f20-91e2-b628fcc489ac')																																		
table_109																																		
發生日期	發生時間	發生年度	發生月份	發生星期	到場處理日期	到場處理時間	結束_事故排除_日期	結束_事故排除_時間	GPS經度	GPS緯度	事故類別名稱	地址類型名稱	發生縣市名稱	發生市區鄉鎮名稱	發生地址_村里代碼	發生地址_村里名稱	發生地址_鄉	發生地址_路街代碼	發生地址_路街	發生地址_段	發生地址_巷	發生地址_弄	發生地址_號	發生地址_前幾公尺	發生地址_側名稱	發生地址_村里名稱	發生交叉路口_路街口	發生交叉路口_側名稱	發生交叉路口_段	發生交叉路口_巷	發生交叉路口_弄	發生地址_其他	死亡人數_24小時內	死亡人數_2_30日內
20200101	56	2020	1	三	20200101	815	20200101	2315	121.208	25.0058	A2	其他	桃園市	中壢區	nan	nan	nan	nan	nan	nan	nan	0		nan	nan	nan	nan	nan	nan	nan	nan	0	0	

110年桃園市交通事故資料表-含第二當事人 資料表

[34] table\_110 = Table().read\_table('https://data.tycg.gov.tw/opendata/datalist/datasetMeta/download?id=ae5d2d9b-59ad-4f15-a9d2-d0f5f3de2475&rid=8b93ee39-5f31-4807-bdfb-4da6bb71db4b')  
table\_110

發生日期	發生時間	發生年度	發生月份	發生星期	到場處理日期	到場處理時間	結束_事故排除_日期	結束_事故排除_時間	GPS經度	GPS緯度	事故類別名稱	地址類型名稱	發生縣市名稱	發生市區鄉鎮名稱	發生地址_村里代碼	發生地址_村里名稱	發生地址_鄉	發生地址_路街代碼	發生地址_路街	發生地址_段	發生地址_巷	發生地址_弄	發生地址_號	發生地址_前幾公尺	發生地址_側名稱	發生地址_村里名稱	發生交叉路口_路街口	發生交叉路口_側名稱	發生交叉路口_段	發生交叉路口_巷	發生交叉路口_弄	發生地址_其他	死亡人數_24小時內	死亡人數_2_30日內
20210101	1600	2021	1	五	20210101	2600	20210101	4100	121.248	24.9571	A2	交叉路口	桃園市	中壢區	nan	nan	6.8e+15	環中東路	nan	nan	0	0	新中北路	nan	nan	0	0							

# 桃園市各區 事故位置類別 數量

```
[35] # 109年 各區域+事故位置類別 (之後統計各區各事故位置數量 要用)
location_109 = table_109.select(['發生市區鄉鎮名稱','事故位置子類別名稱'])
location_109
```

發生市區鄉鎮名稱	事故位置子類別名稱
中壢區	交叉路口內
中壢區	交叉路口內
中壢區	交叉路口內
平鎮區	交叉路口內
平鎮區	交叉路口內
八德區	交叉路口內
八德區	交叉路口內
平鎮區	路肩、路緣
平鎮區	路肩、路緣
平鎮區	路肩、路緣

... (109586 rows omitted)

```
# 109年 各區域事故數量
location_109_total = table_109.group('發生市區鄉鎮名稱')
location_109_total
```

發生市區鄉鎮名稱	count
中壢區	25833
八德區	10314
大園區	3994
大溪區	4357
平鎮區	10526
復興區	283
新屋區	2018
桃園區	21271
楊梅區	6378
蘆竹區	7643

... (3 rows omitted)

```
# 110年 各區域+事故位置類別 (之後統計各區各事故位置數量 要用)
location_110 = table_110.select(['發生市區鄉鎮名稱','事故位置子類別名稱'])
location_110
```

發生市區鄉鎮名稱	事故位置子類別名稱
中壢區	交叉路口內
中壢區	交叉路口內
中壢區	交叉路口內
中壢區	交叉路口內
中壢區	交叉路口內
中壢區	交叉路口內
中壢區	交叉路口內
中壢區	交叉路口內
桃園區	交叉路口內
桃園區	交叉路口內

... (92240 rows omitted)

```
# 110年 各區域事故數量
location_110_total = table_110.groupby('發生市區鄉鎮名稱')
location_110_total
```

發生市區鄉鎮名稱	count
中壢區	20546
八德區	9257
大園區	3600
大溪區	3740
平鎮區	8604
復興區	227
新屋區	1621
桃園區	17344
楊梅區	5419
蘆竹區	6850

... (3 rows omitted)

```
# 109年 <各區域><事故位置類別> 數量統計表
loc_109 = [0 for i in range(13)]
total_109 = [0 for i in range(13)]
a_109 = [0 for i in range(13)]
a2_109 = [0 for i in range(13)]
pd_109 = [0 for i in range(13)]

# 各區域的各事故位置 的數量 之table
for i in range(13):
    loc_109[i] = location_109_total.columns(0).item(i) # 各區域名稱
    total_109[i] = location_109.where('發生市區鄉鎮名稱', are.equal_to(loc_109[i])).groupby('事故位置子類別名稱') # 各區域 分別統計 各事故位置 數量
    pd_109[i] = ((total_109[i].to_df()).rename(columns={'count':loc_109[i]})).set_index('事故位置子類別名稱') # 轉換為dataframe 且 修正欄位名稱(count->各區域名稱)

# 將所有區域table 合併成一個table
for i in range(12):
    loc_109_total = pd_109[i].join(pd_109[i+1], how='outer')
    pd_109[i+1] = loc_109_total

loc_109_total = loc_109_total.astype('Int64') # float轉int
loc_109_total.fillna(0, inplace=True) # 將缺失值填入0
loc_109_total
```

	中壢區	八德區	大園區	大溪區	平鎮區	復興區	新屋區	桃園區	楊梅區	蘆竹區	觀音區	龍潭區	龜山區
事故位置子類別名稱													
一般車道(未劃分快慢車道)	8535	3863	1555	1830	3171	230	881	7091	2701	2870	930	1955	3728
交叉口附近	4021	2094	528	722	2298	17	207	2648	1129	751	239	589	1490
交叉路口內	11410	3611	1385	1273	3739	11	655	9733	1899	3137	933	2019	2881
交通島(含橋化線)	75	9	11	15	49	0	2	26	4	10	2	10	28
人行道	80	10	5	14	9	0	2	54	4	9	2	6	14
公車專用道	0	0	0	0	0	0	0	7	0	0	0	0	2
其他	344	72	53	55	129	8	34	237	194	157	25	24	44
加速車道	5	2	0	2	2	0	0	5	0	0	0	0	0
快車道	218	120	88	69	215	2	61	239	43	113	65	85	154
慢車道	449	279	139	95	272	0	66	494	159	198	141	136	658
機車停等區	77	41	19	24	30	0	3	82	26	26	4	17	22
機車優先道	24	39	50	10	133	0	20	62	25	23	28	11	89
機車專用道	65	2	16	8	95	0	5	113	21	36	2	8	22
機車待轉區	37	10	0	4	14	0	0	20	5	10	0	6	17
減速車道	0	0	0	4	2	0	0	0	0	0	0	0	0
環道匝道	2	0	0	2	0	0	0	2	0	0	0	0	2
直線匝道	4	2	2	0	29	0	5	2	3	3	0	0	0
穿越道附近	14	6	2	0	0	0	5	8	0	4	2	6	4
行人穿越道	131	23	14	17	41	0	2	127	13	31	8	22	34
路肩、路緣	340	131	127	210	282	15	68	315	148	258	117	194	202
迴轉道	2	0	0	3	16	0	2	6	4	7	0	0	2

```
# 110年 <各區域><事故位置類別> 數量統計表
loc = [0 for i in range(13)]
total = [0 for i in range(13)]
a = [0 for i in range(13)]
a2 = [0 for i in range(13)]
pd = [0 for i in range(13)]

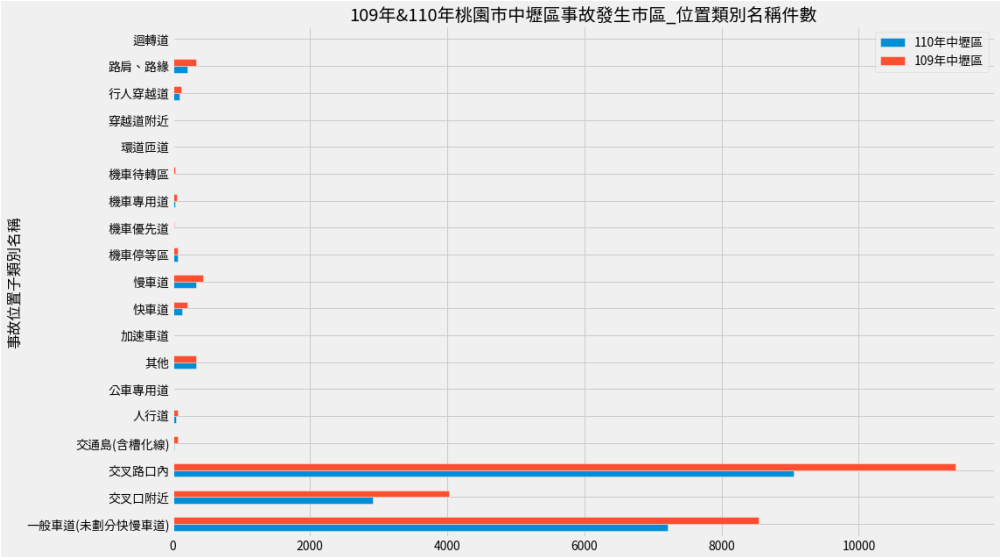
# 各區域的各事故位置 的數量 之table
for i in range(13):
    loc[i] = location_110_total.column(0).item(i) # 各區域名稱
    total[i] = location_110.where('發生市區鄉鎮名稱',are.equal_to(loc[i])).group('事故位置子類別名稱') # 各區域 分別統計 各事故位置 數量
    pd[i] = ((total[i].to_df()).rename(columns={'count':loc[i]})).set_index('事故位置子類別名稱') # 轉換為dataframe 且 修正欄位名稱(count->各區域名稱)

# 將所有區域table 合併成一個table
for i in range(12):
    loc_110_total = pd[i].join(pd[i+1], how='outer')
    pd[i+1] = loc_110_total

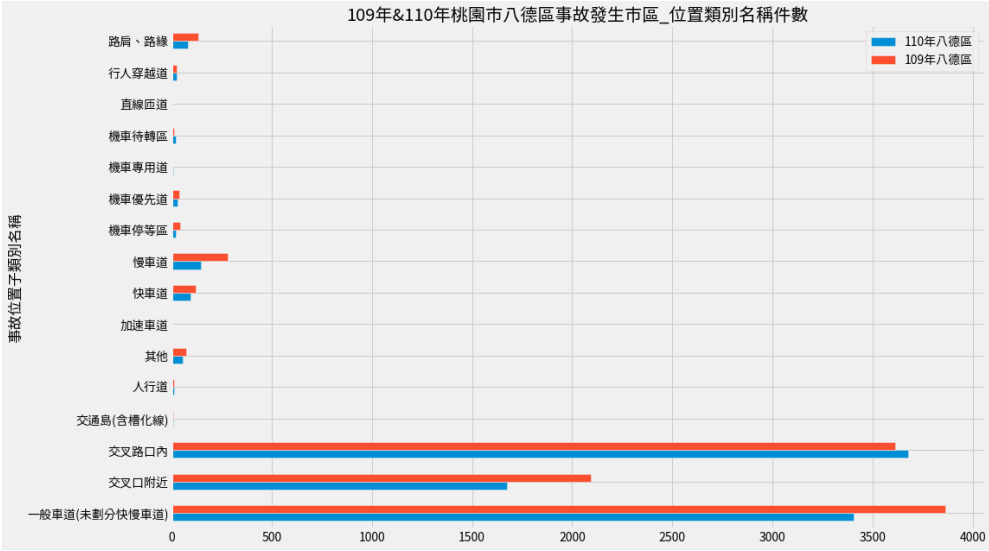
loc_110_total = loc_110_total.astype('Int64') # float轉int
loc_110_total.fillna(0, inplace=True) # 將缺失值填入0
loc_110_total
```

	中壢區	八德區	大園區	大溪區	平鎮區	復興區	新屋區	桃園區	楊梅區	蘆竹區	觀音區	龍潭區	龜山區
事故位置子類別名稱													
一般車道(未劃分快慢車道)	7211	3404	1494	1796	2450	203	710	5739	2453	2797	847	1574	3717
交叉口附近	2921	1673	517	750	1622	0	126	2389	634	605	198	434	1447
交叉路口內	9056	3675	1205	889	3466	8	617	8064	1806	2800	978	1727	2706
交通島(含槽化線)	26	6	4	11	23	2	0	17	8	2	7	7	13
人行道	51	10	2	3	4	0	2	88	10	27	0	3	36
公車專用道	4	0	0	0	0	0	0	4	0	0	0	0	2
其他	345	55	27	17	122	8	22	113	220	58	6	4	38
加速車道	2	4	0	0	0	0	0	0	0	2	2	2	4
快車道	134	96	66	34	190	0	33	134	58	92	81	45	90
慢車道	339	147	74	88	266	0	30	299	71	137	88	80	384
機車停等區	79	20	15	15	38	0	2	67	9	8	4	16	28
機車優先道	16	30	45	7	91	0	20	38	14	42	12	9	39
機車專用道	39	8	14	16	51	0	0	56	6	23	2	6	4
機車待轉區	12	19	0	8	12	0	0	23	0	2	2	2	12
環道匝道	2	0	0	0	4	0	0	0	0	0	2	0	0
直線匝道	0	2	4	6	16	0	0	0	0	0	3	0	2
穿越道附近	2	0	0	0	0	0	0	5	0	0	0	0	0
行人穿越道	94	27	6	16	20	0	2	114	17	33	4	20	18
路肩、路緣	211	81	127	82	221	6	57	192	113	220	124	91	120
迴轉道	2	0	0	2	8	0	0	2	0	2	0	0	2

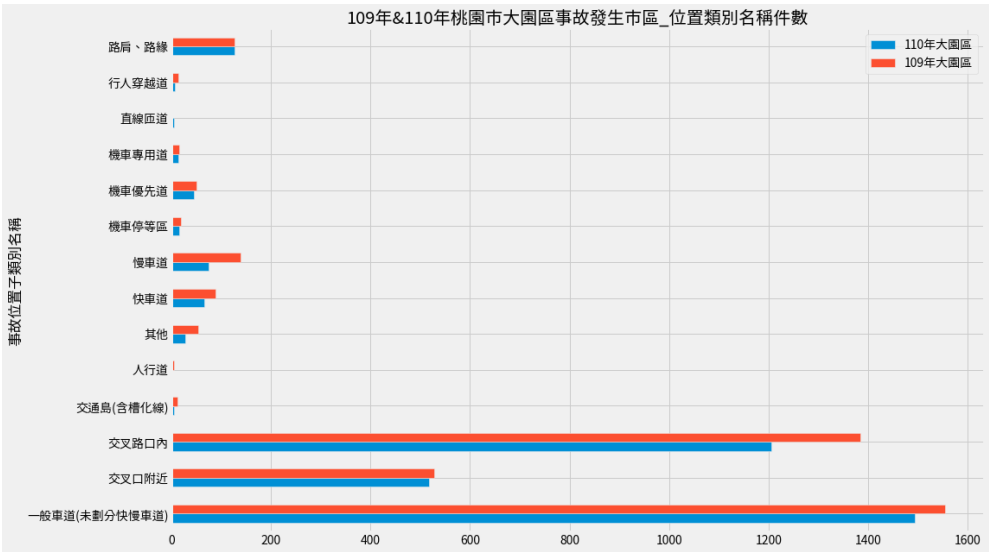
```
# plot 單一區域 109年&110年(mix) 橫直方圖
# total[0]: 中壢區"事故位置子類別"數量統計表
for i in range(13):
    total_bar_109 = ((total_109[i].to_df()).rename(columns={'count':'109年'+loc[i]})).set_index('事故位置子類別名稱') # 109年 轉換為dataframe 且 修正欄位名稱(count->109年各區域名稱)
    total_bar = ((total[i].to_df()).rename(columns={'count':'110年'+loc[i]})).set_index('事故位置子類別名稱') # 110年 轉換為dataframe 且 修正欄位名稱(count->110年各區域名稱)
    total_bar_mix = total_bar.join(total_bar_109) # mix 109年&110年
# plot
total_bar_mix.plot.barh(title='109年&110年桃園市中壢區市事故發生市區_位置類別名稱件數', figsize=(15,10))
plt.show()
```



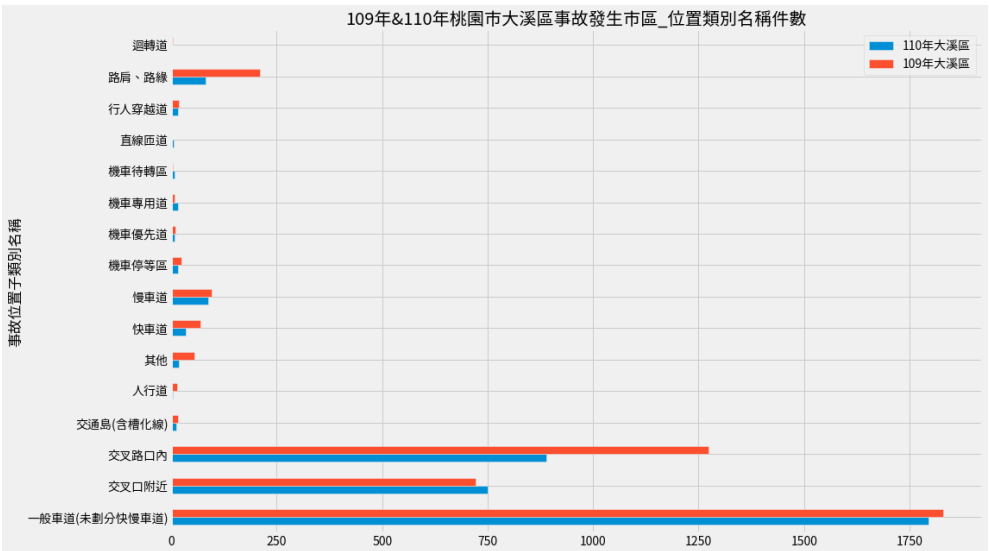
事故位置子類別名稱

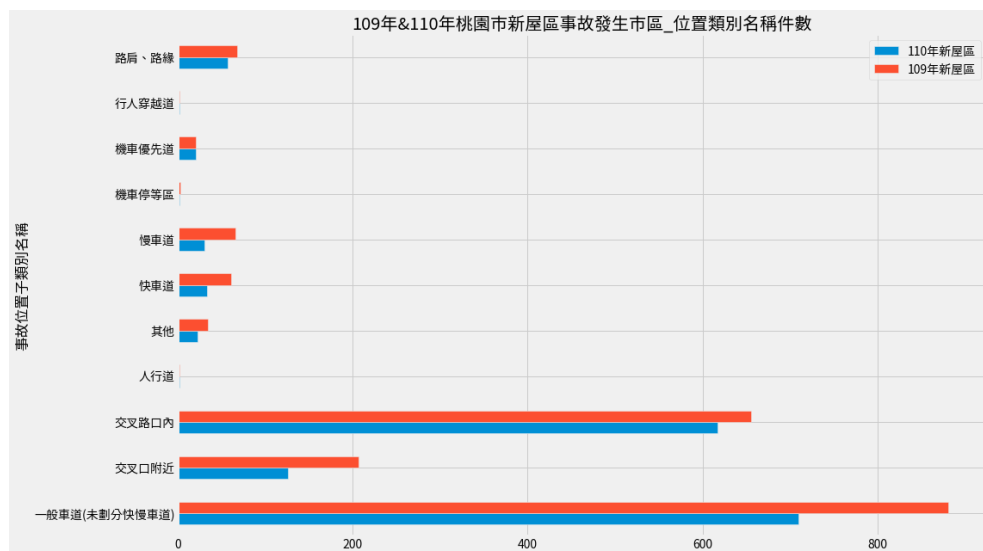
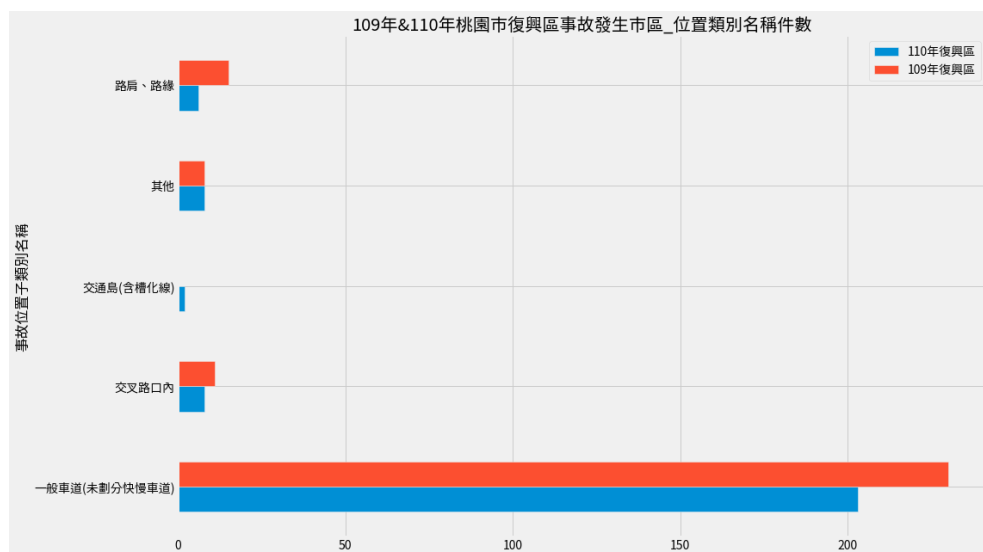
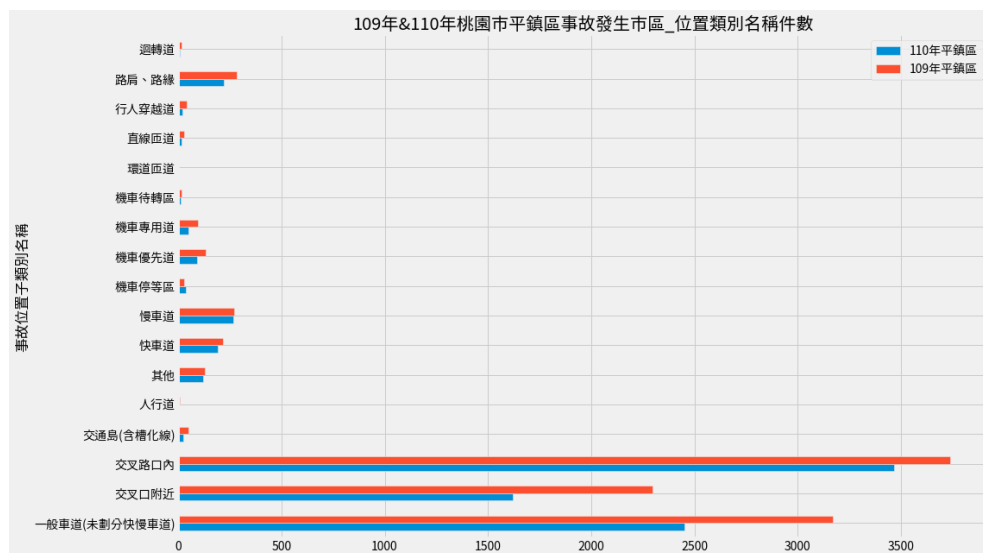


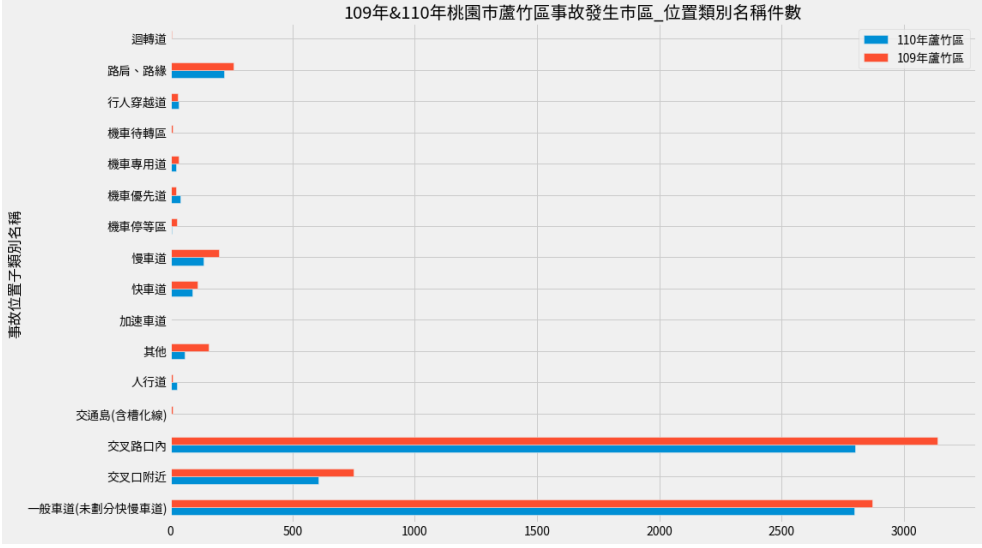
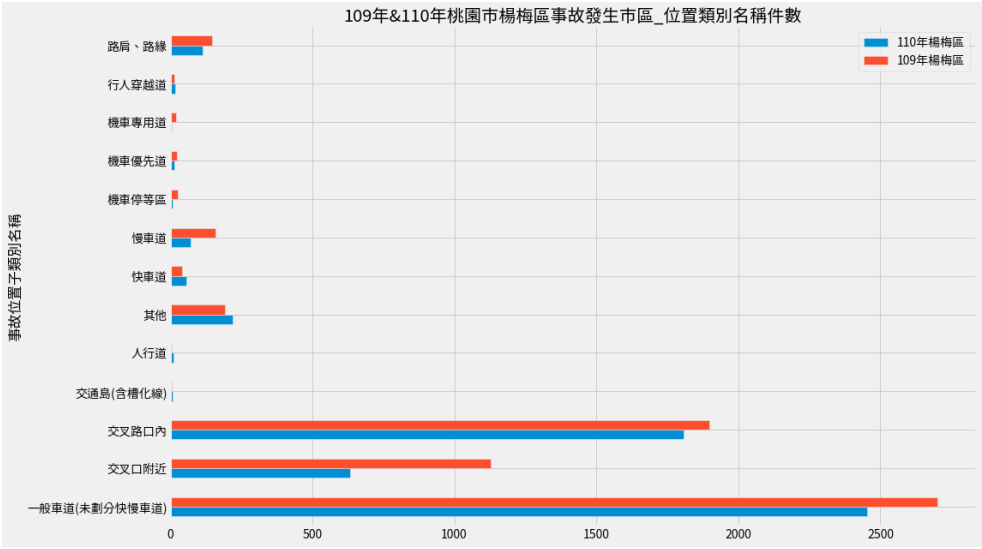
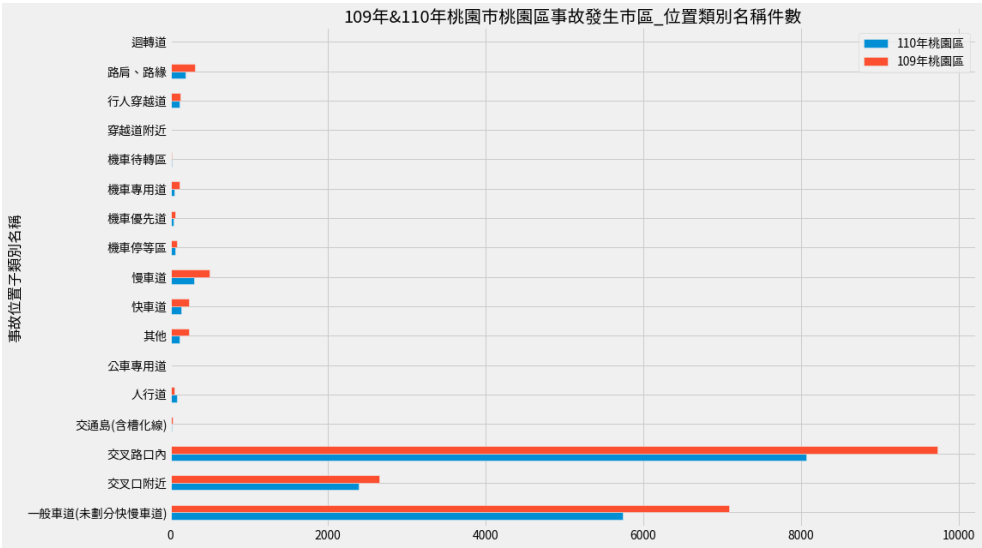
事故位置子類別名稱

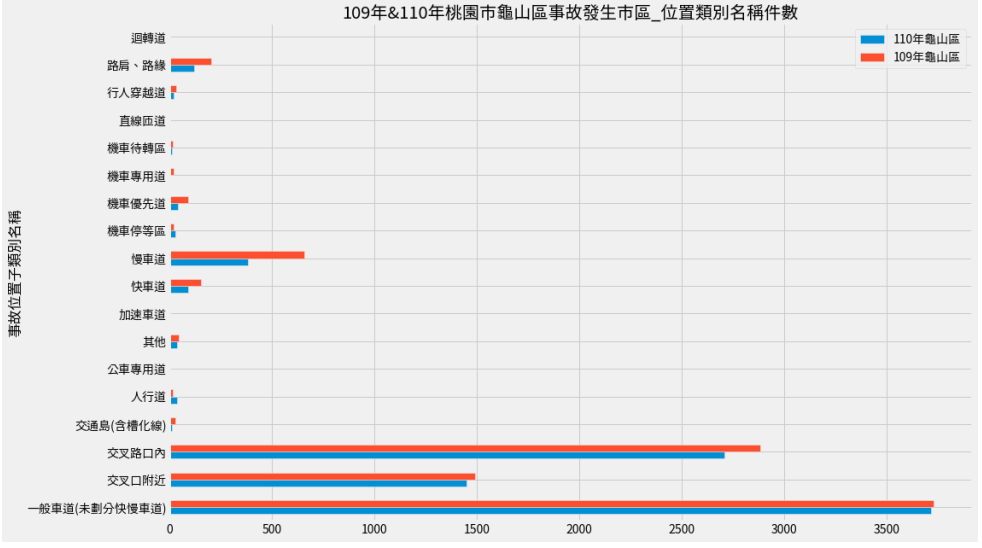
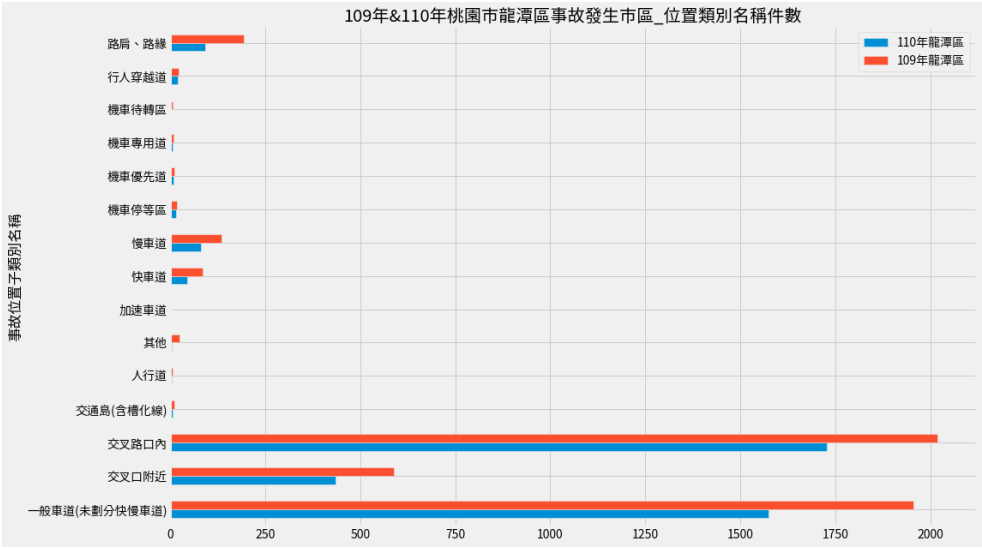
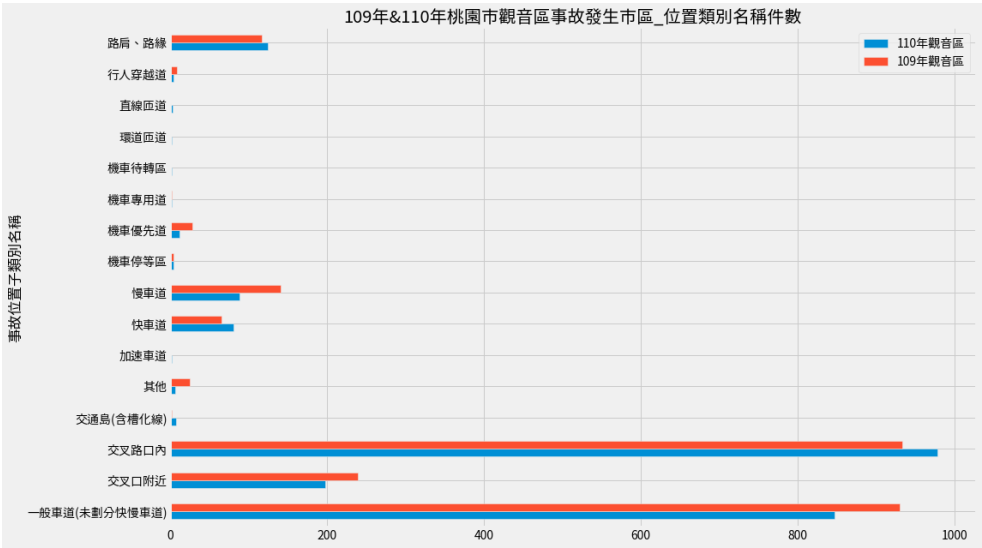


事故位置子類別名稱











## 桃園市 事故位置類別 數量 In-depth Analysis

計算"加權平均" (權重: 整個區域交通事故數量/整個桃園市交通事故數量)

(加權平均=各區域各事故位置類型之數量\*權重 之相加)

=> 桃園市政府應加強"平均比例高的事故位置類別"之相關交通安全政令宣導

\*因各區域地理位置、大小與道路數量等環境因素皆不相同, 故採取加權的方式, 計算整個桃園市交通事故位置平均數量較多之類別

```
# 109年桃園市事故發生位置類別名稱_(加權)平均數量
# 計算加權平均
avg_total_109 = [0 for i in range(21)]
all = table_109.num_rows # 桃園市事故總數量
for i in range(13):
    for j in range(21):
        avg_total_109[j] += loc_109_total.iat[j,i] * (location_109_total.column(1).item(i) / all)

avg_table_109 = Table().with_columns(
    '事故位置子類別名稱', loc_109_total.index,
    '109年桃園市加權平均事故數量', avg_total_109
)
avg_table_109
```

事故位置子類別名稱	109年桃園市加權平均事故數量
一般車道(未劃分快慢車道)	4991.19
交叉口附近	2209.88
交叉路口內	6081.8
交通島(含槽化線)	33.1515
人行道	34.303
公車專用道	1.53001
其他	178.707
加速車道	2.6088
快車道	165.802
慢車道	353.134

... (11 rows omitted)

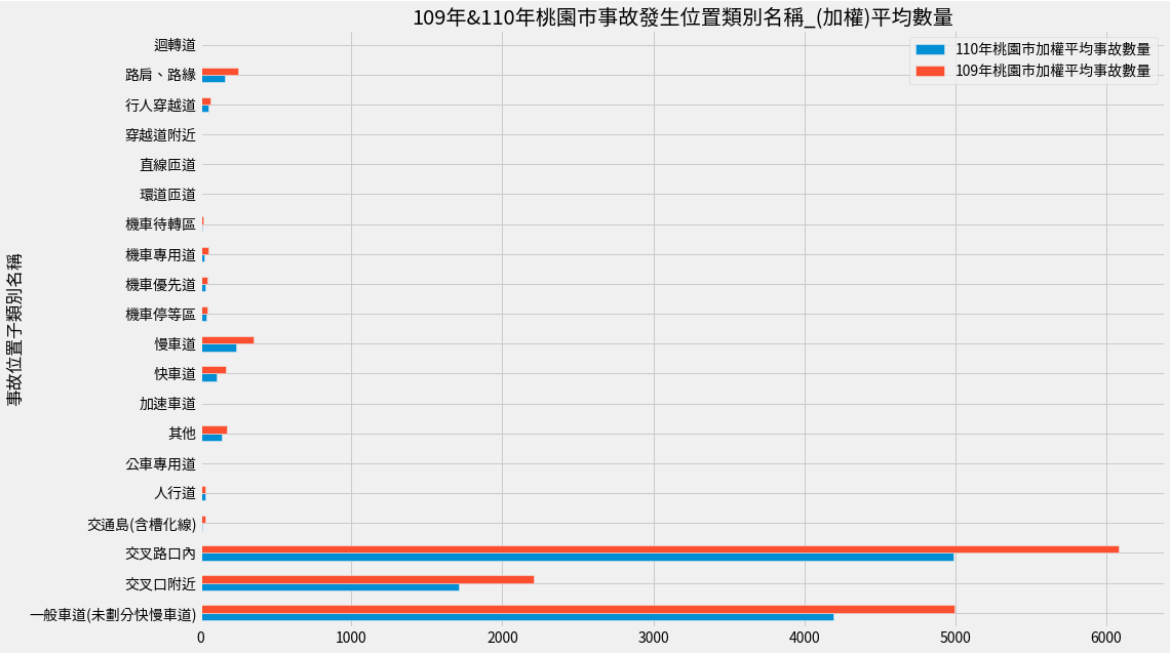
```
# 110年桃園市事故發生位置類別名稱_(加權)平均數量
# 計算加權平均
avg_total_110 = [0 for i in range(20)]
all = table_110.num_rows # 桃園市事故總數量
for i in range(13):
    for j in range(20):
        avg_total_110[j] += loc_110_total.iat[j,i] * (location_110_total.column(1).item(i) / all)

avg_table_110 = Table().with_columns(
    '事故位置子類別名稱', loc_110_total.index,
    '110年桃園市加權平均事故數量', avg_total_110
)
avg_table_110
```

事故位置子類別名稱	110年桃園市加權平均事故數量
一般車道(未劃分快慢車道)	4190.27
交叉口附近	1713.7
交叉路口內	4987.42
交通島(含槽化線)	14.6644
人行道	35.6184
公車專用道	1.83072
其他	138.257
加速車道	1.50925
快車道	109.649
慢車道	234.398

... (10 rows omitted)

```
# plot 109年&110年桃園市事故發生位置類別名稱_(加權)平均數量
avg_table_109 = ((avg_table_109.to_df()).rename(columns={'count':'109年桃園市'})).set_index('事故位置子類別名稱') # 109年 轉換為dataframe 且 修正欄位名稱(count->事故位置子類別名稱)
avg_table_110 = ((avg_table_110.to_df()).rename(columns={'count':'110年桃園市'})).set_index('事故位置子類別名稱') # 110年 轉換為dataframe 且 修正欄位名稱(count->事故位置子類別名稱)
avg_table_mix = avg_table_110.join(avg_table_109) # mix 109年&110年
avg_table_mix.plot.barh(title='109年&110年桃園市事故發生位置類別名稱_(加權)平均數量', figsize=(15,10))
plt.show()
```



# 桃園市交通酒駕事故發生時的年齡分佈

## 110年酒駕發生的死亡人數與受傷人數

```
drunk_table110=table_110n.select('當事者事故發生時年齡','死亡人數_24小時內','死亡人數_2_30日內','飲酒情形名稱').where('飲酒情形名稱',are.equal_to(drunk_name)).to_df()
df1=drunk_table110[(drunk_table110['死亡人數_24小時內'] > 0) | (drunk_table110['死亡人數_2_30日內'] >0) ]
df1
```

當事者事故發生時年齡	死亡人數_24小時內	死亡人數_2_30日內	飲酒情形名稱
62	41	1	0 經呼氣檢測0.26~0.40mg/L或血液檢測0.051%~0.08%
182	42	1	0 經呼氣檢測0.56~0.80mg/L或血液檢測0.111%~0.16%
210	25	1	0 經呼氣檢測0.26~0.40mg/L或血液檢測0.051%~0.08%
215	37	1	0 經呼氣檢測0.56~0.80mg/L或血液檢測0.111%~0.16%
231	44	0	1 經呼氣檢測0.56~0.80mg/L或血液檢測0.111%~0.16%
232	47	1	0 經呼氣檢測0.56~0.80mg/L或血液檢測0.111%~0.16%
323	20	1	0 經呼氣檢測0.26~0.40mg/L或血液檢測0.051%~0.08%
349	20	1	0 經呼氣檢測0.56~0.80mg/L或血液檢測0.111%~0.16%

## 計算110年酒駕事故的各年齡段死亡總人數

```
total1_18=0
total1_26=0
total1_30=0
total1_40=0
total1_50=0
total1_60=0

for i in df1.index:
    if 18<= df1['當事者事故發生時年齡'][i] <=25:
        total1_18+=df1['死亡人數_24小時內'][i]+df1['死亡人數_2_30日內'][i]
    elif 26<= df1['當事者事故發生時年齡'][i] <=29 :
        total1_26+=df1['死亡人數_24小時內'][i]+df1['死亡人數_2_30日內'][i]
    elif 30<= df1['當事者事故發生時年齡'][i] <=39:
        total1_30+=df1['死亡人數_24小時內'][i]+df1['死亡人數_2_30日內'][i]
    elif 40<= df1['當事者事故發生時年齡'][i] <=49:
        total1_40+=df1['死亡人數_24小時內'][i]+df1['死亡人數_2_30日內'][i]
    elif 50<= df1['當事者事故發生時年齡'][i] <=59:
        total1_50+=df1['死亡人數_24小時內'][i]+df1['死亡人數_2_30日內'][i]
    elif df1['當事者事故發生時年齡'][i] >=60:
        total1_60+=df1['死亡人數_24小時內'][i]+df1['死亡人數_2_30日內'][i]
```

```
df_i110=table_110n.select('當事者事故發生時年齡','受傷人數','飲酒情形名稱').where('飲酒情形名稱',are.equal_to(drunk_name)).to_df()
df_i110
```

當事者事故發生時年齡	受傷人數	飲酒情形名稱
0	18	2 經呼氣檢測0.26~0.40mg/L或血液檢測0.051%~0.08%
1	34	1 經呼氣檢測0.26~0.40mg/L或血液檢測0.051%~0.08%
2	22	2 經呼氣檢測0.56~0.80mg/L或血液檢測0.111%~0.16%
3	33	2 經呼氣檢測0.41~0.55mg/L或血液檢測0.081%~0.11%
4	62	1 經呼氣檢測0.26~0.40mg/L或血液檢測0.051%~0.08%
...	...	...
461	53	2 經呼氣檢測0.26~0.40mg/L或血液檢測0.051%~0.08%
462	49	1 經呼氣檢測0.26~0.40mg/L或血液檢測0.051%~0.08%
463	39	2 經呼氣檢測0.56~0.80mg/L或血液檢測0.111%~0.16%
464	35	1 經呼氣檢測0.41~0.55mg/L或血液檢測0.081%~0.11%
465	22	2 經呼氣檢測0.56~0.80mg/L或血液檢測0.111%~0.16%

466 rows × 3 columns

## 計算110年酒駕事故的各年齡段受傷總人數

```

totali1_18=0
totali1_26=0
totali1_30=0
totali1_40=0
totali1_50=0
totali1_60=0

for i in df_i110.index:
    if 18<= df_i110['當事者事故發生時年齡'][i] <=25:
        totali1_18+=df_i110['受傷人數'][i]
    elif 26<= df_i110['當事者事故發生時年齡'][i] <=29 :
        totali1_26+=df_i110['受傷人數'][i]
    elif 30<= df_i110['當事者事故發生時年齡'][i] <=39:
        totali1_30+=df_i110['受傷人數'][i]
    elif 40<= df_i110['當事者事故發生時年齡'][i] <=49:
        totali1_40+=df_i110['受傷人數'][i]
    elif 50<= df_i110['當事者事故發生時年齡'][i] <=59:
        totali1_50+=df_i110['受傷人數'][i]
    elif df_i109['當事者事故發生時年齡'][i] >=60:
        totali1_60+=df_i110['受傷人數'][i]

```

```

die_inj_result=Table().with_columns(
    '年齡區間',make_array('18-25','26-29','30-39','40-49','50-59','60以上'),
    '109年受傷人數',make_array(totali_18,totali_26,totali_30,totali_40,totali_50,totali_60),
    '110年受傷人數',make_array(totali1_18,totali1_26,totali1_30,totali1_40,totali1_50,totali1_60),
    '109年死亡人數',make_array(total_18,total_26,total_30,total_40,total_50,total_60),
    '110年死亡人數',make_array(total1_18,total1_26,total1_30,total1_40,total1_50,total1_60)
)
die_inj_result

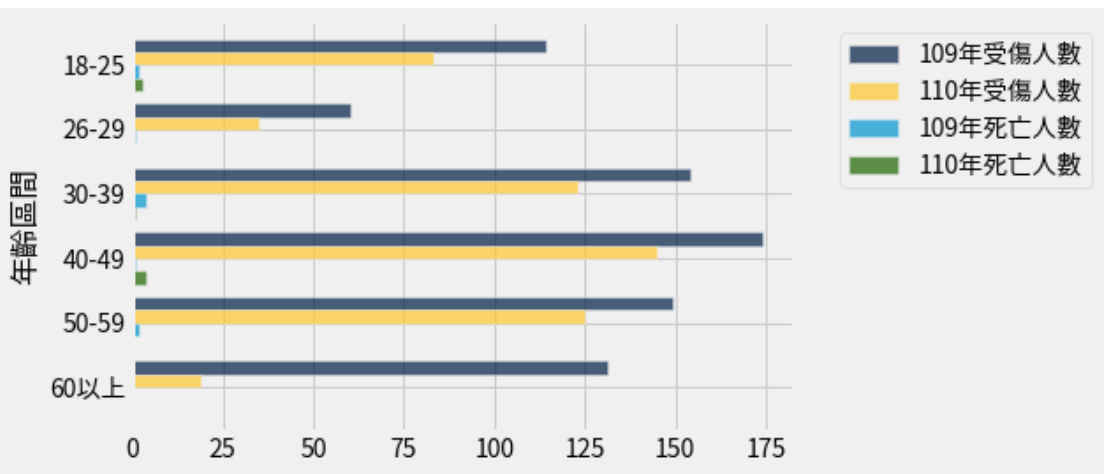
```

年齡區間	109年受傷人數	110年受傷人數	109年死亡人數	110年死亡人數
18-25	114	83	2	3
26-29	60	35	1	0
30-39	154	123	4	1
40-49	174	145	1	4
50-59	149	125	2	0
60以上	131	19	0	0

```

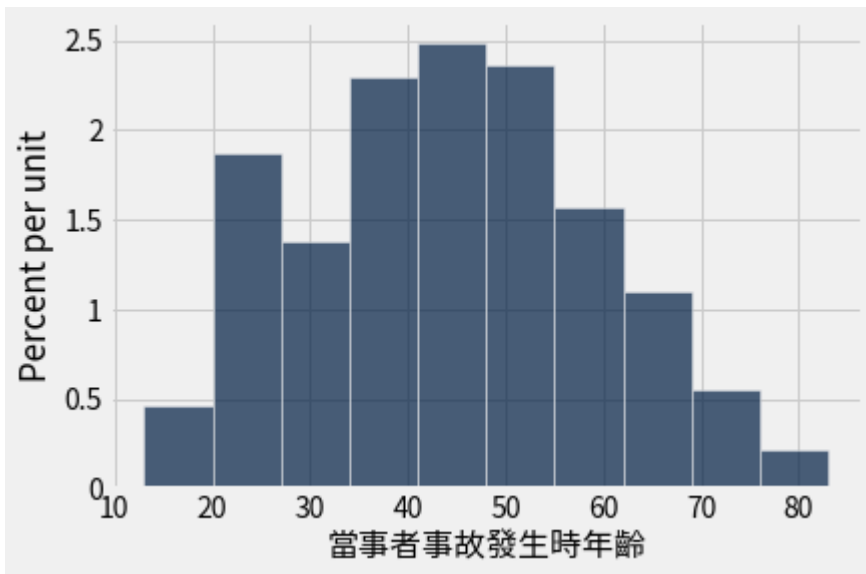
die_inj_result.barh('年齡區間')
# plot

```



## 桃園市交通酒駕事故發生時的年齡分佈In-depth Analysis

```
drunk_name=make_array('經呼氣檢測0.26~0.40mg/L或血液檢測0.051%~0.08%', '經呼氣檢測0.41~0.55mg/L或血液檢測0.081%~0.11%', '經呼氣檢測0.56~0.80mg/L或血液檢測0.111%~0.16%')
df=table_110.select('當事者事故發生時年齡', '受傷人數', '飲酒情形名稱').where('飲酒情形名稱', are.equal_to(drunk_name))
df=df.select('當事者事故發生時年齡')
df.hist()
```

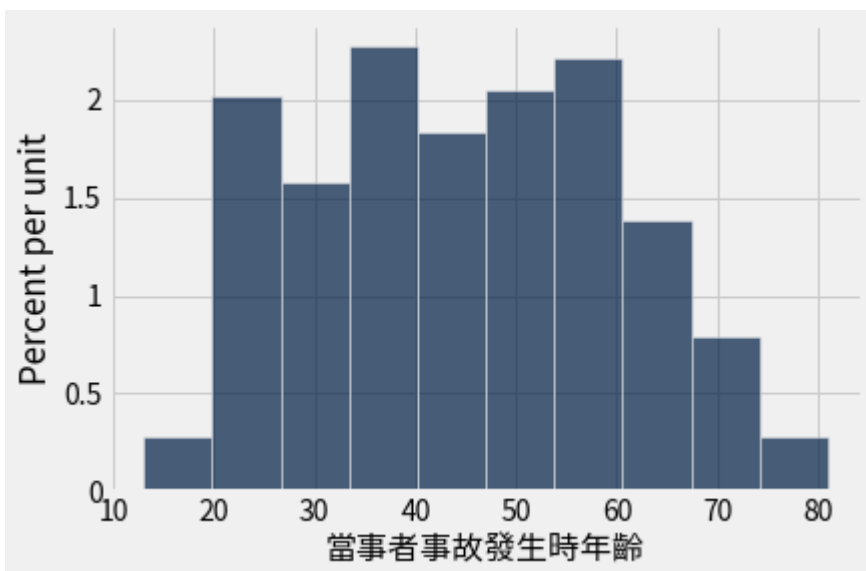


```
pop_median = percentile(50, df.column('當事者事故發生時年齡'))
pop_median
#酒駕事故發生的中位數年齡
```

43

```
resample_1 = df.sample()
#從table中隨機抽第一樣本
```

```
resample_1.select('當事者事故發生時年齡').hist()
#隨機抽第一樣本的直方圖
```



```
resampled_median_1 = percentile(50, resample_1.column('當事者事故發生時年齡'))
resampled_median_1
#隨機抽第一樣本的中位數年齡
```

\*44

```
resample_2 = df.sample()
resampled_median_2 = percentile(50, resample_2.column('當事者事故發生時年齡'))
resampled_median_2
#隨機抽第二樣本的中位數年齡
```

\*42

```
def one_bootstrap_median():
    resampled_table = df.sample()
    bootstrapped_median = percentile(50, resampled_table.column('當事者事故發生時年齡'))
    return bootstrapped_median

one_bootstrap_median()

#bootstrap函數的中位數年齡
```

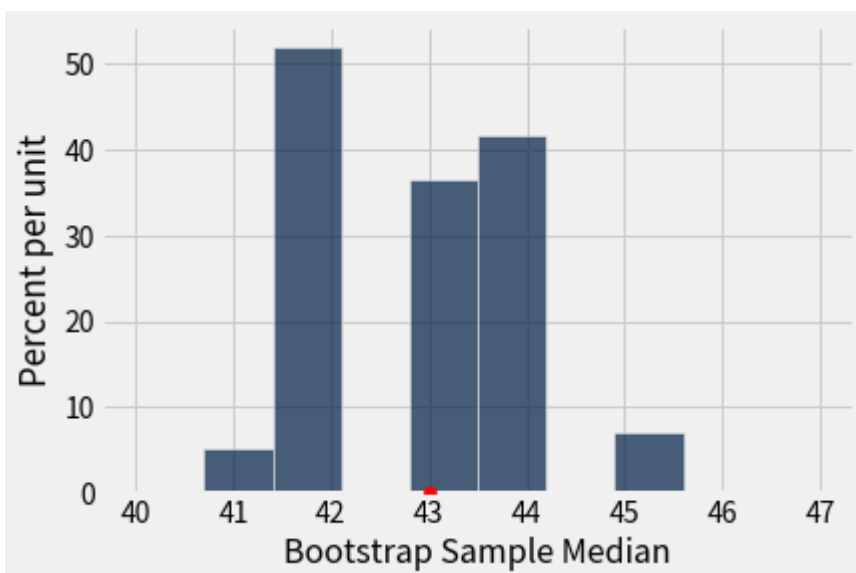
\*44

```
num_repetitions = 5000
bstrap_medians = make_array()
for i in np.arange(num_repetitions):
    bstrap_medians = np.append(bstrap_medians, one_bootstrap_median())
```

#在每次迴圈中，我們將調用函數one\_bootstrap\_median來生成一個基於原始樣本df的bootstrapped median的值，然後我們將把bootstrapped中位數追加到集合數組bstrap\_medians中。

```
resampled_medians = Table().with_column('Bootstrap Sample Median', bstrap_medians)
resampled_medians.hist()

plt.scatter(pop_median, 0, color='red', s=40, zorder=5);
#與原本總體的中位數年齡結果不一致
```



# 桃園市交通事故發生時的年齡分佈

## 計算110年發生事故的各年齡段總人數

110年事故發生的年齡段

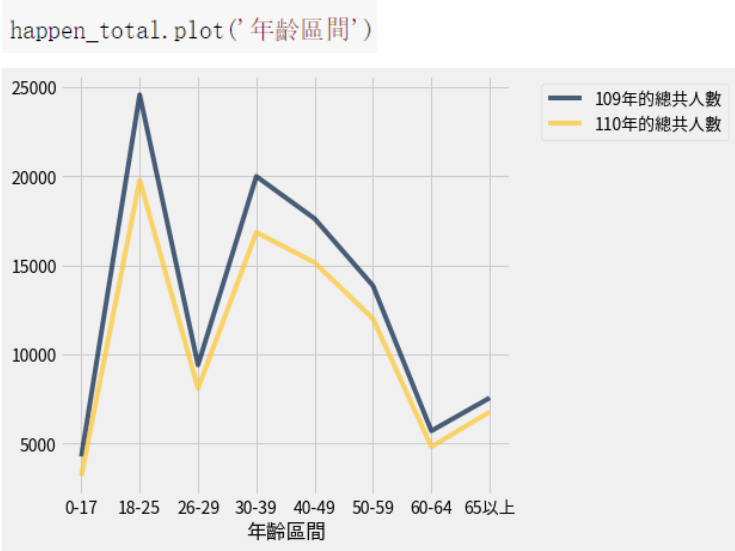
+ 程式碼+ 文字

```
[45] total1_0_17 = sum(table_110n.where('當事者事故發生時年齡', are.between(0,18)).group('當事者事故發生時年齡').column('count'))
total1_18_25 = sum(table_110n.where('當事者事故發生時年齡', are.between(18,26)).group('當事者事故發生時年齡').column('count'))
total1_26_29 = sum(table_110n.where('當事者事故發生時年齡', are.between(26,30)).group('當事者事故發生時年齡').column('count'))
total1_30_39 = sum(table_110n.where('當事者事故發生時年齡', are.between(30,40)).group('當事者事故發生時年齡').column('count'))
total1_40_49 = sum(table_110n.where('當事者事故發生時年齡', are.between(40,50)).group('當事者事故發生時年齡').column('count'))
total1_50_59 = sum(table_110n.where('當事者事故發生時年齡', are.between(50,60)).group('當事者事故發生時年齡').column('count'))
total1_60_64 = sum(table_110n.where('當事者事故發生時年齡', are.between(60,65)).group('當事者事故發生時年齡').column('count'))
total1_65 = sum(table_110n.where('當事者事故發生時年齡', are.above(65)).group('當事者事故發生時年齡').column('count'))
```

對比這兩年每個年齡層發生的事故有沒有減少

```
[46] happen_total = Table().with_columns(
    '年齡區間',make_array('0-17','18-25','26-29','30-39','40-49','50-59','60-64','65以上'),
    '109年的總共人數',make_array(total_0_17,total_18_25,total_26_29,total_30_39,total_40_49,total_50_59,total_60_64,total_65),
    '110年的總共人數',make_array(total1_0_17,total1_18_25,total1_26_29,total1_30_39,total1_40_49,total1_50_59,total1_60_64,total1_65)
)
happen_total
```

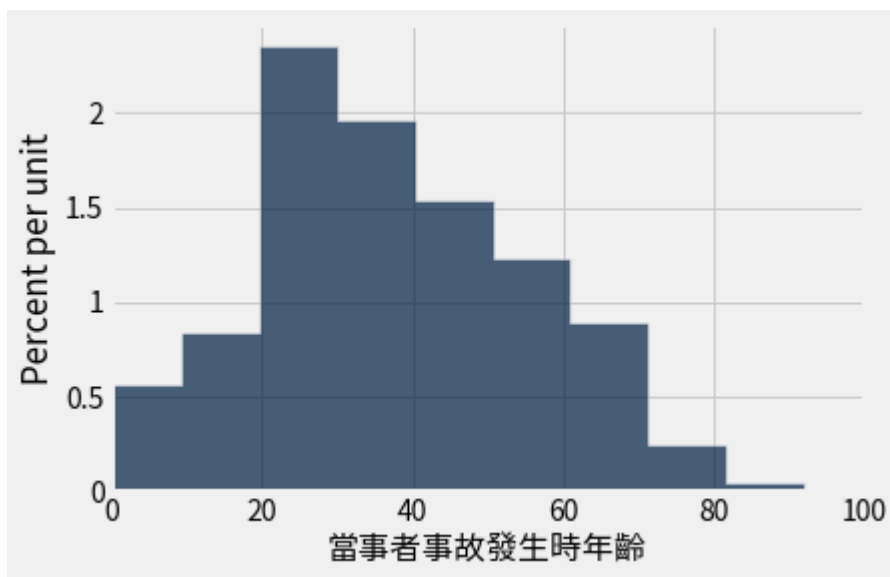
年齡區間	109年的總共人數	110年的總共人數
0-17	4276	3183
18-25	24584	19799
26-29	9413	8093
30-39	20000	16860
40-49	17625	15158
50-59	13858	12008
60-64	5715	4820
65以上	7566	6783



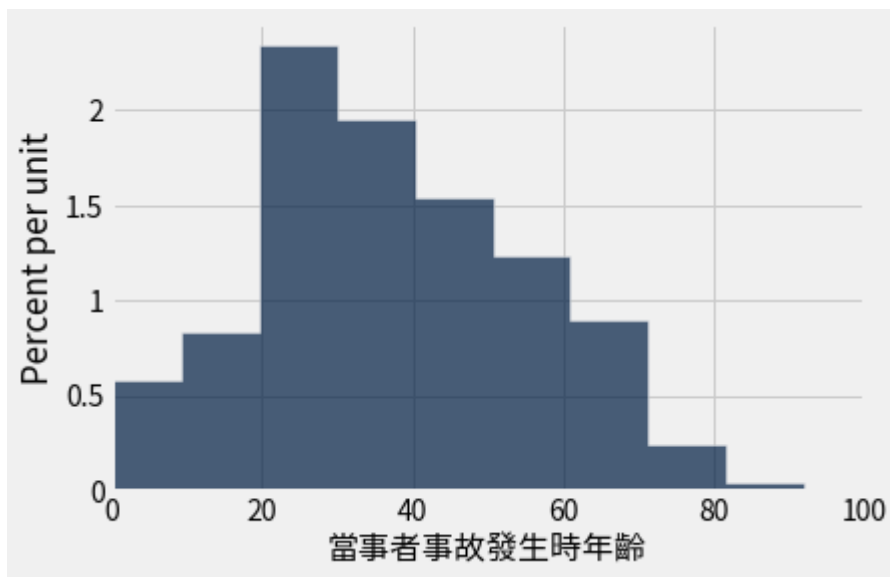
從以上的綫狀圖顯示藍色是109年、黃色是110年，從這邊明顯看得出18-25歲是剛考到駕照的時候，而且這個年齡層發生事故比其它年齡層都要得多。

## 桃園市交通事故發生時的年齡分佈In-depth Analysis

```
total = table_110.select('當事者事故發生時年齡')
pop_median1 = percentile(50, total.column('當事者事故發生時年齡'))
#事故發生的中位數年齡
total.hist(bins=100)
plt.xlim(0, 100)
```



```
resample1 = total.sample()
resample1.select('當事者事故發生時年齡').hist(bins=100)
plt.xlim(0, 100)
#從總體抽隨機第一樣本
```



```
resampled_median1 = percentile(50, resample1.column('當事者事故發生時年齡'))
resampled_median1
#第一樣本的中位數年齡
```

\*36

```
resample2 = total.sample(100)
resampled_median2 = percentile(50, resample2.column('當事者事故發生時年齡'))
resampled_median2
#從總體抽100的隨機第二樣本的中位數年齡
```



\*35

```
def one_bootstrap_median1():
    resampled_table = total.sample(500)
    bootstrapped_median = percentile(50, resampled_table.column('當事者事故發生時年齡'))
    return bootstrapped_median

one_bootstrap_median1()

#bootstrap函數的中位數年齡
```

\*36

```
num_repetitions1 = 5000
bstrap_medians1 = make_array()
for i in np.arange(num_repetitions):
    bstrap_medians1 = np.append(bstrap_medians1, one_bootstrap_median1())
```

#在每次迴圈中，我們將調用函數one\_bootstrap\_median來生成一個基於原始樣本total的bootstrapped median的值，然後我們將把bootstrapped中位數追加到集合數組bstrap\_medians中。

```
resampled_medians1 = Table().with_column('Bootstrap Sample Median1', bstrap_medians1)

resampled_medians1.hist()

plt.scatter(pop_median1, 0, color='orange', s=40, zorder=10);

#與原本總體的中位數年齡結果相似
```

