

Matplotlib 圖表(柱狀圖、圓餅圖、柏拉圖、DataFrame)

備註：圖表的部分，實際上，我原先有放在另外一個檔案：

[HW2^7視覺化資料分析^7鐵達尼號乘客名單推敲生還者資訊.pdf](#) (四、分析報告的地方我有放圖表。也建議可以看看我做的4000字分析報告 (前言、程式碼、以及分析的結果...))

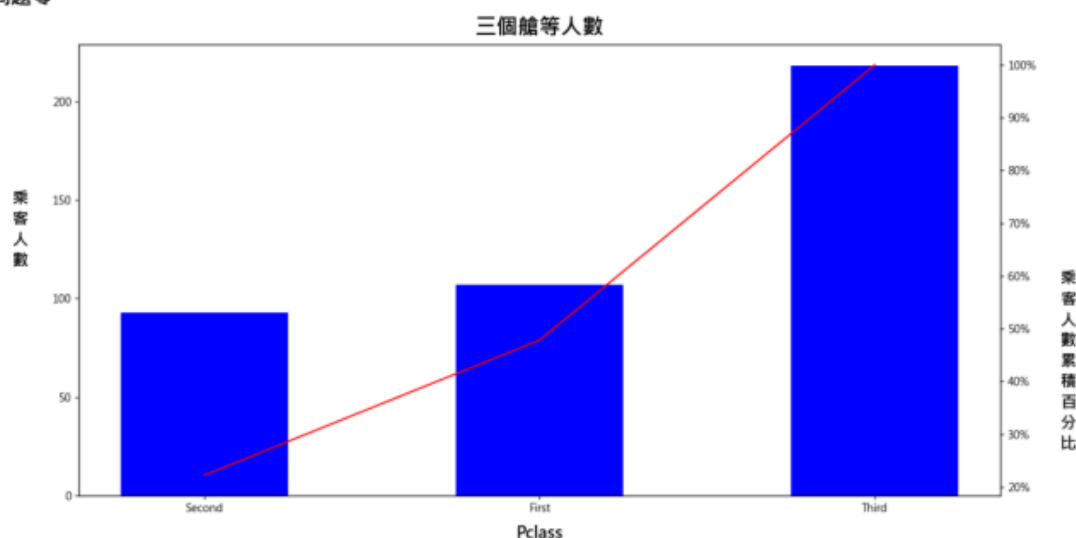
一、DataFrame & Matplotlib圖表

1. 問題 01:我想知道不同艙等階級的人數有多少？

各艙層乘客人數如下：

	count	perc(%)	cumperc(%)
Second	93	22.25	22.25
First	107	25.60	47.85
Third	218	52.15	100.00

問題零

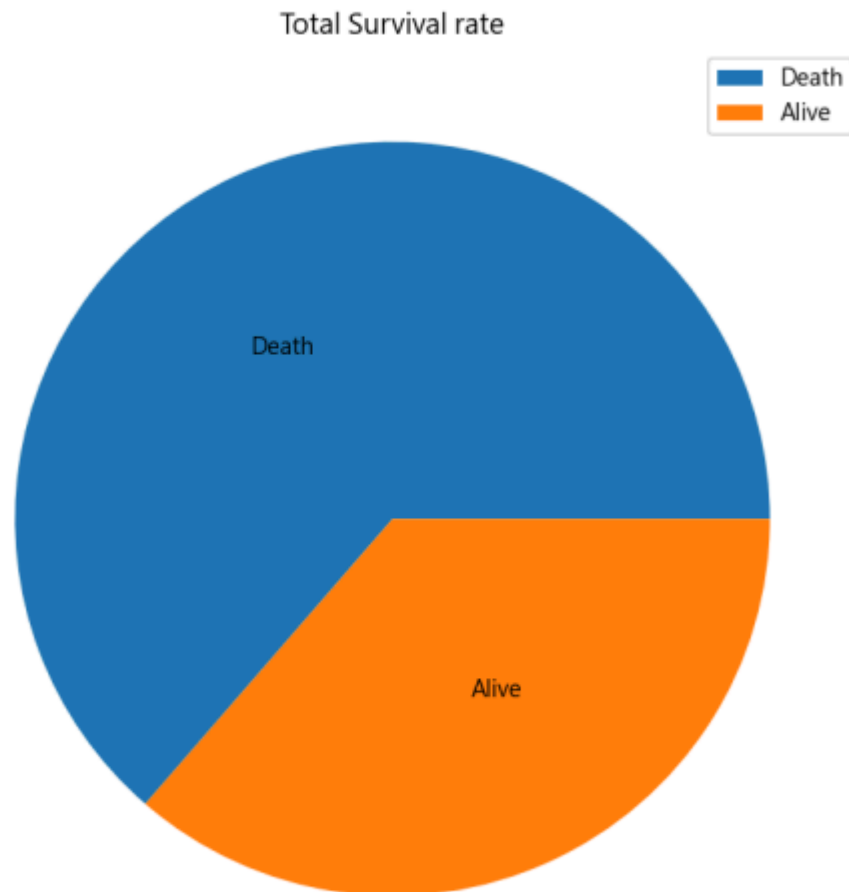


圖十二：各艙層乘客人數表(柏拉圖)

2. 問題 02:我想知道三個艙等的生還者人數的高低落差，是否會因為社會階級不同而影響到生還率高低？

各艙層的生還與死亡率分布：

	Alive	Death	Alive perc(%)	Death perc(%)
First	50	57	46.73	53.27
Second	30	63	32.26	67.74
Third	72	146	33.03	66.97
Total	152	266	36.36	63.64



圖十四：Total Survival rate



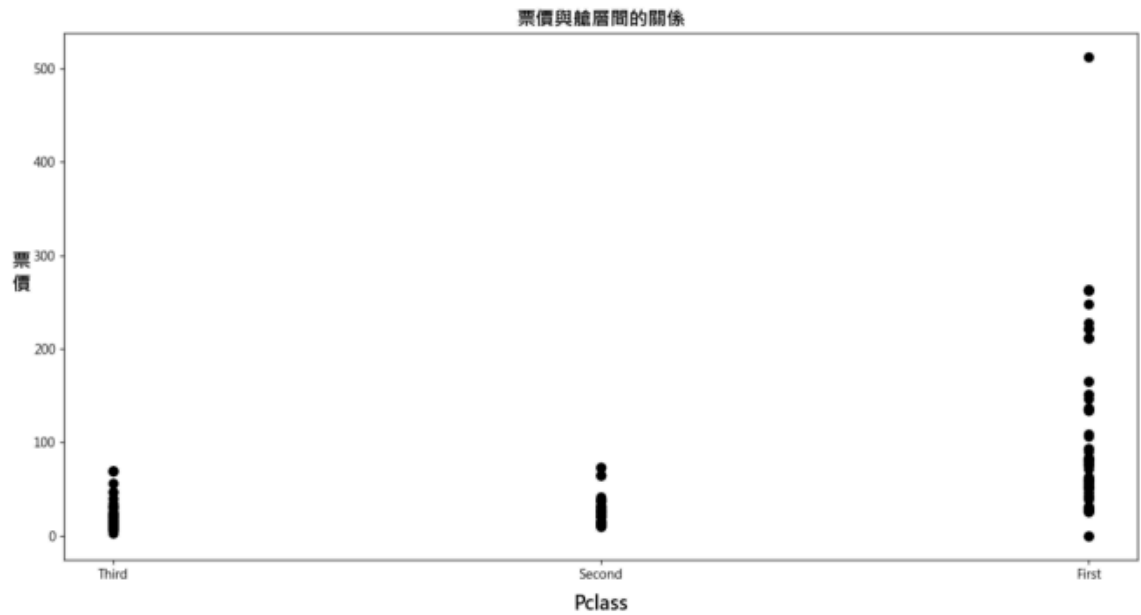
圖十五：三個艙層的生還率

3. 問題 03:我想知道票價是否與艙等有關係？

不同階層不同乘客所支付票價一覽表：

	Pclass	Fare
892	Third	7.8292
893	Third	7.0000
894	Second	9.6875
895	Third	8.6625
896	Third	12.2875
...
1305	Third	8.0500
1306	First	108.9000
1307	Third	7.2500
1308	Third	8.0500
1309	Third	22.3583

[418 rows x 2 columns]



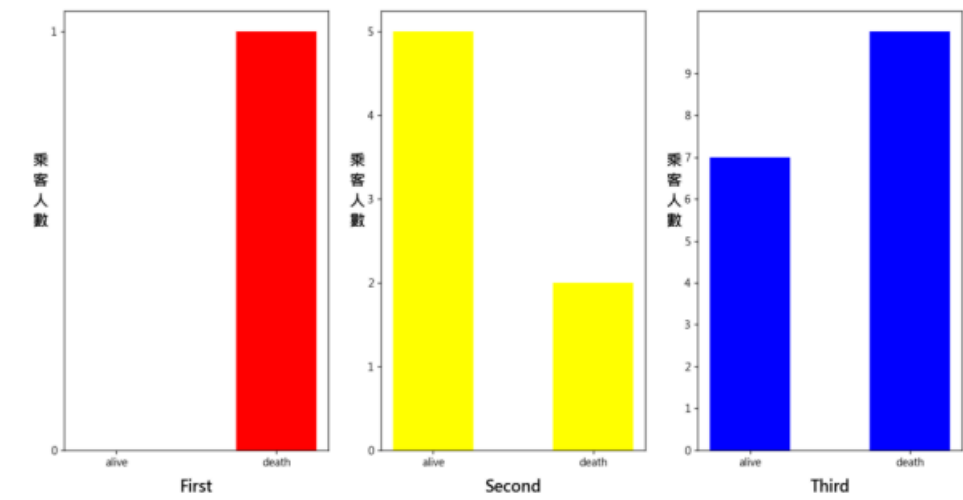
圖十七：票價與艙層間的關係(散佈圖)

4. 問題 04:我想瞭解不同階層不同年齡層是否會影響生存人數？

各艙層生還與死亡人數概況：

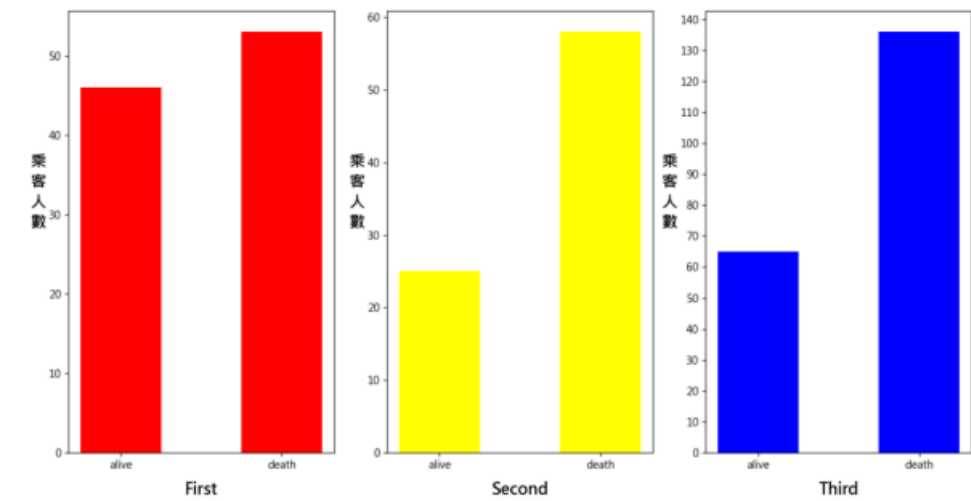
	Alive	Death
(兒童, First)	0	1
(兒童, Second)	5	2
(兒童, Third)	7	10
(青壯年, First)	46	53
(青壯年, Second)	25	58
(青壯年, Third)	65	136
(老人, First)	4	3
(老人, Second)	0	3
(老人, Third)	0	0

問題三：不同艙層的兒童存亡概況



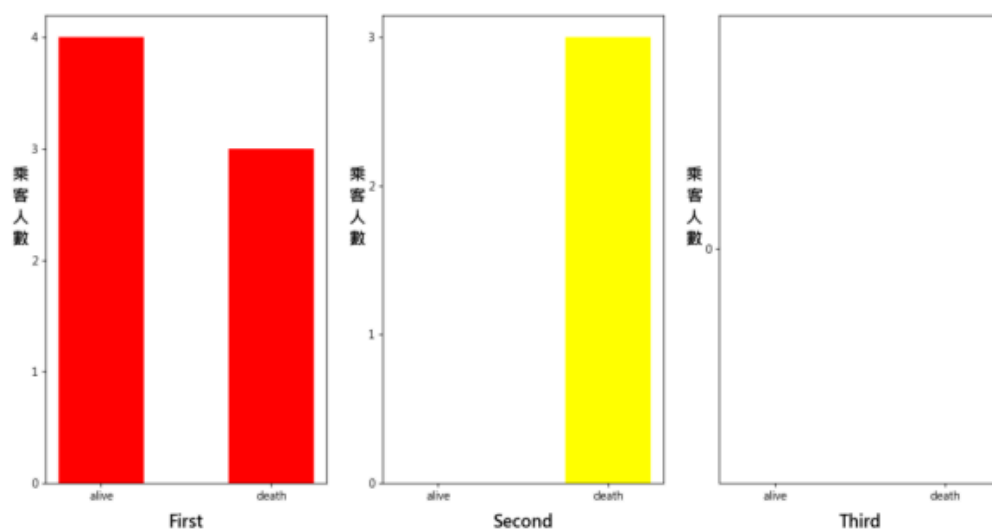
圖十八：不同艙層的兒童存亡概況

問題三：不同艙層的青壯年存亡概況



圖十九：不同艙層的青壯年存亡概況

問題三：不同艙層的老人存亡概況



圖二十：不同艙層的老人存亡概況

二、所使用的程式碼

1. 架設pandas,nupy,matplotlib環境

```
# 00. 架設pandas,nupy,matplotlib環境
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.ticker import PercentFormatter

# 01. 將中文字體設為微軟正黑體 (使圖表可讀中文字體)
plt.rcParams['font.sans-serif'] = ['Microsoft JhengHei']
```

2. 先做篩選與計算各個項目的資訊(生還者、死亡人數)

```

# 一、先做篩選與計算各個項目的資訊(生還者、死亡人數)

# 00. 讀取資料
df = pd.read_csv["titanic_data_repair.csv"]

# 01. 計算各艙等的總乘客數量
passenger_df = df[df['Pclass'].isin(["First", "Second", "Third"])].groupby('Pclass').size()

# 02. 計算各艙等的總生還者
survival_data = df[(df['Survived'] == 'alive') & (df['Pclass'].isin(["First", "Second", "Third"]))].groupby('Pclass').size()

# 03. 計算各艙等的總死亡人數
death_data = df[(df['Survived'] == 'death') & (df['Pclass'].isin(["First", "Second", "Third"]))].groupby('Pclass').size()

# 04. 計算總生還人數
total_survival_data = (df['Survived'] == 'alive').sum()

# 05. 計算總死亡人數
total_death_data = (df['Survived'] == 'death').sum()

# 06. 將各年齡層依序分組
age_bins = [0, 13, 61, 81] # 0-12歲為一組(兒童); 13-60歲為一組(青壯年); 61-80歲為一組(老人)
age_labels = ['兒童', '青壯年', '老人']
df['Age_Group'] = pd.cut(df['Age'], bins=age_bins, labels=age_labels, right=False)

# 07. 篩選出各年齡層生還人數
Pclass_survival = df[df['Survived'] == 'alive'].groupby(['Age_Group', 'Pclass'], observed=False).size()

# 08. 篩選出各年齡層死亡人數
Pclass_death = df[df['Survived'] == 'death'].groupby(['Age_Group', 'Pclass'], observed=False).size()

```

3. 將問題所需資料轉換成字典或是列表 (DataFrame 事前作業)

```

# 二、將問題所需資料轉換成字典或是列表 ( DataFrame 事前作業)

# 00. 問題零：將各艙層乘客人數轉為字典形式
passenger_df_dict = passenger_df.to_dict()
keys = [key for key in passenger_df_dict.keys()]
values = [value for value in passenger_df_dict.values()]

# 01. 問題零：各個不同艙層的生還與死亡轉換為字典形式
survival_data_dict = survival_data.to_dict()
survival_data_dict.update({'Total Alive':total_survival_data})
survival_data_values = [value for value in survival_data_dict.values()]
death_data_dict = death_data.to_dict()
death_data_dict.update({'Total Death':total_death_data})
death_data_values = [value for value in death_data_dict.values()]

# 02. 問題一、問題三：將階級設為列表，方便在回答問題一、問題三時回應
list_Pclass=['First','Second','Third']
list_Pclass.append('Total')

# 03. 問題二：將各艙層的乘客與所支付票價轉換為列表形式
Pclass_passenger_ID = df['PassengerId'].to_list()
Pclass_passenger_Fare = df['Fare'].to_list()

# 04. 問題三：將各年齡層乘客是否生還的人數轉為字典形式
Pclass_Survival_dict = Pclass_survival.to_dict()
Survival_values= [value for value in Pclass_Survival_dict.values()]
Survival_keys = [key for key in Pclass_Survival_dict.keys()]
Pclass_Death_dict = Pclass_death.to_dict()
Death_values= [value for value in Pclass_Death_dict.values()]

```

4. Pandas 圖表 (DataFrame)

```

# 三、Pandas 圖表
def All_Passenger_data_Infor():
    print(f"\n乘客資訊總表:\n\n{df}")

# 01. 各艙層乘客人數:
def Pclass_passengers():

    df_0 = pd.DataFrame({'count': values})
    df_0.index = keys

    df_0 = df_0.sort_values(by='count', ascending=True)

    df_0['perc(%)'] = round(df_0['count']/df_0['count'].sum()*100,2)

    df_0['cumperc(%)'] = round(df_0['count'].cumsum()/df_0['count'].sum()*100,2)

    print(f'\n各艙層乘客人數如下:\n\n{df_0}')
    return df_0

# 02. 各個不同艙層的生還率 DataFrame 圖表
def Pclass_alive_or_death_perc():
    df_1 = pd.DataFrame({'Alive':survival_data_values})
    df_1.index = list_Pclass
    df_1['Death'] = death_data_values
    df_1['Alive perc(%)'] = round(df_1['Alive']/(df_1['Alive']+df_1['Death'])*100,2)
    df_1['Death perc(%)'] = round(df_1['Death']/(df_1['Alive']+df_1['Death'])*100,2)
    print(f'\n各艙層的生還與死亡率分布:\n\n{df_1}')
    return df_1

# 03. 各個不同艙層所花費的票價
def Pclass_Fare():
    df_2 = pd.DataFrame({'Pclass':df['Pclass']})
    df_2.index = Pclass_passenger_ID
    df_2['Fare'] = Pclass_passenger_Fare
    print(f'\n不同階層不同乘客所支付票價一覽表:\n\n{df_2}')
    return df_2

# 04. 各年齡層乘客生還與否 DataFrame 圖表
def Pclass_alive_or_death():
    df_3 = pd.DataFrame({"Alive":Survival_values})
    df_3.index = Survival_keys
    df_3['Death'] = Death_values
    print(f'\n各艙層生還與死亡人數概況:\n\n{df_3}')
    return df_3

```

5. 統計圖表

四、統計圖表

00. 柱狀圖

```
def bar_plot(x, y, width, color='blue'):  
    plt.bar(x, y, width, color=color)
```

01. 折線圖

```
def line_plot(x,y,color="blue"):  
    plt.plot(x,y,color=color)
```

02. 散佈圖

```
def scatter_plot(x,y,c=None,s=None):  
    plt.scatter(x,y,c=c,s=s)
```

03. 圓餅圖

```
def pie_plot(x, labels=None, labeldistance=0.5, title=None):  
    plt.title(title)  
    plt.pie(x, labels=labels, labeldistance=labeldistance)  
    plt.legend()
```

6. 問題函式

```
# 00. 問題零：我想知道，不同艙等階級的人數有多少？(柏拉圖呈現)  
def question_0():  
    plt.subplots(1, 1)  
    plt.subplot(1, 1, 1)  
    df_0 = Pclass_passengers()  
    bar_plot(df_0.index, df_0['count'], width=0.5, color="blue")  
    plt.xlabel('Pclass', labelpad=10, fontsize=15, color='black', fontweight='bold')  
    plt.ylabel('乘\客\人\數', rotation=0, labelpad=30, fontsize=15, color='black', fontweight='bold')  
    plt.title('三個艙等人數', pad=10, fontsize=20, color='black', fontweight='bold')  
    plt.twinx()  
    line_plot(df_0.index, df_0['cumperc(%)'], color='red')  
    plt.ylabel('乘\客\人\數\累\積\百\分\比', rotation=0, labelpad=30, fontsize=15, color='black', fontweight='bold')  
    plt.gca().yaxis.set_major_formatter(PercentFormatter())  
    plt.figtext(0.05, 0.95, '問題零', fontsize=20, color='black', fontweight='bold')  
  
# 01. 問題一：我想知道，三個艙等的生還者人數的高低落差，是否會因為社會階級不同而影響到生還率高低？(圓餅圖)  
def question_1():  
    Pclass_alive_or_death_perc() # 呼叫函式並接收返回的 DataFrame  
    plt.subplots(1,3)  
    for n in range(4):  
        if(n+1==4):  
            plt.subplots(1,1)  
            plt.subplot(1,1,1)  
            plt.figtext(0.05, 0.95, '問題一', fontsize=20, color='black', fontweight='bold')  
            x = [death_data_values[3], survival_data_values[3]]  
            pie_plot(x,labels=['Death','Alive'],labeldistance=0.5, title=f'{list_Pclass[n]} Survival rate')  
        else:  
            plt.subplot(1,3,n+1)  
            plt.figtext(0.05, 0.95, '問題一', fontsize=20, color='black', fontweight='bold')  
            x = [death_data_values[n], survival_data_values[n]]  
            pie_plot(x,labels=['Death','Alive'],labeldistance=0.5, title=f'{list_Pclass[n]} class Survival rate')
```

```
# 02. 問題二：我想知道票價是否與艙等有關係（散佈圖）
def question_2():
    df_2=Pclass_Fare()
    pclass_list = df_2['Pclass'].to_list()
    plt.subplots(1,1)
    plt.subplot(1,1,1)
    plt.xlabel('Pclass',labelpad=10,fontsize=15, color='black',fontweight='bold')
    plt.ylabel('票價', rotation=0, labelpad=10, fontsize=15, color='black', fontweight='bold')
    plt.title('票價與艙層間的關係',fontsize=15, color='black',fontweight='bold')
    plt.figtext(0.05, 0.95, '問題二', fontsize=20, color='black', fontweight='bold')
    scatter_plot(pclass_list,Pclass_passenger_Fare,c="black",s=50)

# 03. 問題三：我想瞭解不同階層不同年齡層是否會影響生存人數？（柱狀圖）
def question_3():
    colors = ['red','yellow','blue']
    Pclass_alive_or_death()
    for i in range(3):
        plt.subplots(1,3)
        plt.figtext(0.05, 0.95, f'問題三：不同艙層的{age_labels[i]}存活概況', fontsize=20, color='black', fontweight='bold')
        for n in range(3):
            plt.subplot(1, 3, n+1)
            if i ==0:
                plt.yticks(np.arange(0, 10, 1))
            elif i ==1:
                plt.yticks(np.arange(0, 150, 10))
            else:
                plt.yticks(np.arange(0, 10, 1))
            bar_plot(['alive','death'],[Survival_values[n+i*3],Death_values[n+i*3]],width=0.5,color={colors[n]})
            plt.xlabel(f"{list_Pclass[n]}", labelpad=10, fontsize=15, color='black', fontweight='bold')
            plt.ylabel('乘客\死亡人數', rotation=0, labelpad=10, fontsize=15, color='black', fontweight='bold')
```

7. 主程式&回答問題

```
# 六、主程式
def main():
    # 問題零-問題三
    All_Passenger_data_Infor()
    question_0()
    question_1()
    question_2()
    question_3()
    # 顯示從問題零-問題三的所有圖表
    plt.show()

# 七、四大問題 & 回答問題

# 00. 問題零：我想知道，不同艙等階級的人數有多少？（柏拉圖呈現）
# 01. 問題一：我想知道，三個艙等的生還者人數的高低落差，是否會因為社會階級不同而影響到生還率高低？（利用圓餅圖呈現）
# 02. 問題二：我想知道票價是否與艙等有關係（散佈圖）
# 03. 問題三：我想瞭解不同階層不同年齡層是否會影響生存人數？（柱狀圖）

'''以下是四大問題的作答，回答問題（程式如下）'''
main()
'''以上四大問題回答完畢，感謝助教辛苦批閱'''
```