

HW2：視覺化資料分析：鐵達尼號乘客名單推敲生還者資訊

41171204H 呂沛修

一、前言與背景

- 研究契機：本身是鐵達尼號迷，以前跟隨家人看《鐵達尼號》這部電影時，便深深為其中的劇情著迷。有一段時間，甚至癡迷到每天都在上網找尋鐵達尼號相關的資料。雖然鐵達尼號的真實故事沒有像電影那般的浪漫，反而多了幾分悲情，但這反而勾起了我更濃烈的興趣。我想要透過在網路上找到的 Kaggle 資料，推敲過去鐵達尼號船難的真相，揭開沉在冰山下神秘的面紗。
- 在 Kaggle 上找到的資料：
<https://www.kaggle.com/datasets/brendan45774/test-file>
- 程式碼網址：https://github.com/PeiHsiuLu/l12-2-Programming-Language/blob/main/Homework/HW2_%E8%A6%96%E8%A6%BA%E5%8C%96%E8%B3%87%E6%96%99%E5%88%86%E6%9E%90%E5%BC%9A%E9%90%B5%E9%81%94%E5%B0%BC%E8%99%9F%E4%B9%98%E5%AE%A2%E5%90%8D%E5%96%AE%E6%8E%A8%E6%95%B2%E7%94%9F%E9%82%84%E8%80%85%E8%B3%87%E8%A8%8A.py
- 數據分析 — 選用 CSV 檔與 Json 檔背後的思考：
在選擇 CSV 檔還是 Json 檔進行資料分析，讓我猶豫了許久，但是後來經過細細思量還是決定選擇 CSV 檔進行分析，原因如下：
 1. 易用性：我從 Kaggle 網站獲取的數據多數以 Excel 檔形式提供，這使得將數據轉換為 CSV 檔變得非常容易。CSV 檔格式直觀，易於處理，並且與許多數據分析工具相容，使得數據分析流程更加順暢。
 2. 輕量節省空間：相對於 JSON 檔，CSV 檔通常佔用更少的存儲空間。這是因為 CSV 檔僅包含數據的原始值，不像 JSON 檔可能包含冗雜的格式化或信息。在大規模數據集的情況下，使用 CSV 檔能夠有效地節省空間和資源。
 3. 專一性：我選擇專注於處理來自同一份檔案的數據，而不額外處理 JSON 檔。太多不同來源的數據可能會導致分析結果的不精確性。因此，選擇僅專注於 CSV 檔能夠確保分析過程的專注性和一致性。
- 可取用數據總數：418 筆資料。實際上，船難發生時，共有 2224 名乘客，

所以，本次的分析，僅僅只能從 18.79% 的資料回推母體實際船難發生時的正確情形，期望能從有效的樣本數量(>30 筆)推得母體概況。

二、數據整理

- 由於從 Kaggle 網站下載的檔案是提供給專門學習數據分析的學習者挑戰與使用，因此，有些數據被上傳者刻意清空，導致有幾筆空資料，而空資料我的處理方式如下：
 1. Age(年齡)：空的資料我用 excel 函式算出中位數，並將空資料全數套入中位數 27。
 2. Fare(船價)：由於每一票價價格不盡相等，但篩選 Pclass 與 Fare 資料後可以得知，每一艙等(Pclass)的船價還是有略微的不同。所以，我便特意篩選空的那筆 Fare 資料，得知其乘客是二等艙的乘客後，我便將二等艙乘客全部用 Excel 篩選起來，並分析乘客中所支付票價的金額的中位數，最後將中位數(Median=14.4542)代入空資料的二等艙乘客。
 3. Cabin 項目：由於 Cabin 欄位過多空資料，所以後來經過細細斟酌決定將該項目刪去。刪去原因有二：其一，找不到分析的目的；其二，太多欄位缺失(僅有 4 筆資料，資料總計 418 筆)，若是任意補缺值，可能造成資料分析上的不準確。
- 另外，在處理 Age(年齡)時，發現到有小數的情形，便利用 rounddown 函式將資料全數轉換為整數。(無條件捨去，讓資料符合真實性)
- 數值全數轉換為文字，在數據分析時，可以更容易分析資料，不受數值影響：
 1. Pclass：1 轉為 First；2 轉為 Second；3 轉為 Third。(class)
 2. Survived：1 轉為 alive；0 轉為 death
- 當數據整理完畢後，便可進入程式撰寫與分析的作業。

三、程式碼撰寫

- 環境建置：建設 Pandas, numpy, matplotlib 架設基礎數據分析環境，並設定 plt.rcParams，使圖表可讀取中文，並將字體設為「微軟正黑體」。

```
# 零、架設Matplotlib與Pandas環境：

# 00. 架設pandas,numpy,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']
```

圖一：環境建置

- 設定問題：設定四個問題，方便為找到資料分析找到合適的研究方向：
 1. 我想知道，不同艙等階級的人數有多少？
 2. 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？
 3. 我想知道票價是否與艙等有關係？
 4. 我想瞭解不同階層不同年齡層是否會影響生存人數？
- 設定這四大問題，理由有四：
 1. 首先，我想要知道三個艙等具體的人數大致有多少。
 2. 再者，由於西方社會當時階級森嚴（頭等艙多為社會階級頂層人物；二等艙多為作家、中產階級人士；三等艙則多為各色人種與社會底層人士）。據說，就連鐵達尼號各艙等的菜單也有區別。由於社會階級的不同而有不同的待遇，更讓我心生疑惑：是否居住艙等也會影響生還機率？另外，所居住艙等是否也會因為支付票價而有所差異？
 3. 最後，因為當時有一個傳言是：當時由於救生艇數量不足以救援全數的乘客，所以讓婦孺與老人優先登艇。所以我好奇：是否小孩與老人這兩大年齡群的生還數目真的大大提升了？
- 另外，我曾經企圖嘗試分析性別是否影響到生還率，但後來經過分析發現：由於所得資料有限，男性全數為死亡，存活者僅為女性。因此，便捨棄了這項分析。但由此也可推知鐵達尼號當時情況，男性應為弱勢者，相較於女性，較不容易被救援。
- 針對所提出問題進行資料分析與篩選：
 我透過資料的交集與 groupby, sum 的計算，找到適切資料進行篩選分析，以下是我分析與計算的九大資料：

1. 讀取整體資料
2. 計算各艙等的總乘客數量
3. 計算三個艙等各自的總生還者
4. 計算三個艙等的總死亡人數
5. 計算總生還人數
6. 計算總死亡人數
7. 將各年齡層分組(兒童：0-12 歲；青壯年：13-60 歲；老人：61-80 歲。此年齡範圍依據當時時代進行推斷分析)
8. 篩選各年齡層生還人數
9. 篩選各年齡層死亡人數

```
# 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()
```

圖二：計算與篩選所需檔案資料

1. 將問題所需資料轉換為字典或是列表形式：

由於要進行 pandas 資料分析，將資料轉為 dataframe，所以將部分資料轉為字典或是列表形式會較為容易處理。意欲轉換為字典或是列表的資料如下：

 1. 將各艙層乘客人數轉為字典形式
 2. 將階級設為列表
 3. 將各艙層的乘客與所支付票價轉換為列表形式
 4. 將各年齡層乘客是否生還的人數轉為字典形式

```

# 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()]

```

圖三：將問題所需資料轉換為字典或是列表形式

2. 利用 Pandas 將各式重要資料(問題)顯示出來。這樣用各式視覺化圖表顯示時，就能夠清楚顯示數據。並且，為了讓資料不顯得混亂難懂，僅僅有註解是不夠的，必須使用函式進行包裝，有結構化的程式才比較整齊。
 1. 各艙層乘客人數(回應問題：我想知道，不同艙等階級的人數有多少?)
 2. 各個不同艙層的生還率 DataFrame 圖表(回應問題：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?)
 3. 各個不同艙層所花費的票價(回應問題：我想知道票價是否與艙等有關係)
 4. 各年齡層乘客生還與否 DataFrame 圖表(回應問題：我想瞭解不同階層不同年齡層是否會影響生存人數?)

```

# 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各艙層的生還率如下:\n\n{df_1}')
    return df_1

```

圖四：各式 Pandas 圖表

- 統計圖表：

由於要回答的問題數多，且會有幾個問題會用到重複的圖表。為了避免程式碼過度冗長，所以我便設計了四大函式，分別對應到我會使用到的四張統計圖表：

1. 折線圖
2. 柱狀圖
3. 散佈圖
4. 圓餅圖

```

125 |     return df_3
126 |
127 |
128 | # 四、統計圖表
129 |
130 | # 00. 柱狀圖
131 | def bar_plot(x, y, width, color='blue'):
132 |     plt.bar(x, y, width, color=color)
133 |
134 | # 01. 折線圖
135 | def line_plot(x,y,color="blue"):
136 |     plt.plot(x,y,color=color)
137 |
138 | # 02. 散佈圖
139 | def scatter_plot(x,y,c=None,s=None):
140 |     plt.scatter(x,y,c=c,s=s)
141 |
142 | # 03. 圓餅圖
143 | def pie_plot(x, labels=None, labeldistance=0.5, title=None):
144 |     plt.title(title)
145 |     plt.pie(x, labels=labels, labeldistance=labeldistance)
146 |     plt.legend()
147 |

```

圖五：各式統計圖表

- 其中，柏拉圖會用折線圖與柱狀圖的函式來共同合成一張圖，各式問題對

應到的統計圖表如下：

1. 我想知道不同艙等階級的人數有多少？（柏拉圖）
 - ◆ 利用柏拉圖，清楚知道哪個艙層人數最多，哪個艙層人數最少。
 2. 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）
 - ◆ 透過百分比，清楚得知各艙層生還比例。另外，順帶多加一個總體生還率與死亡率，透過圓餅圖，清楚分析百分比。
 3. 我想知道票價是否與艙等有關係（散佈圖）
 - ◆ 透過散佈圖中的資料點，了解艙等高低與票價的關係。
 4. 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）
 - ◆ 透過三張圖（每張圖各有三張子圖）來清楚呈現各個年齡層（兒童、青壯年、老人）不同艙層的生還、死亡人數有何不同。一張圖呈現一個年齡層，子圖呈現艙等不同。
- 問題函式：將所想提問的問題與回答放入各式問題函式中(同時放入回應問題的 DataFrame)
1. 我想知道不同艙等階級的人數有多少？

```
plt.pie(x, labels=labels, labeldistance=labeldistance,
plt.legend()

# 五、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==0):
```

圖六：我想知道不同艙等階級的人數有多少？

2. 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？
 - ◆ 我利用迴圈 for 與 if 來設計總體生還率的圓餅圖與各艙層生還率不同的三張子圖


```
# 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')
```

圖七：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？

3. 我想知道票價是否與艙等有關係

```
# 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.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)
```

圖八：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？

4. 我想瞭解不同階層不同年齡層是否會影響生存人數？

- ◆ 首先是外層的 for 迴圈，用來迭代三次，用來設定三大年齡層(兒童、青壯年、老人)內建的迴圈則代表艙層，其中內建的 if/else 代表設定在不同情況下的 ylabel。另外，colors 列表也被放入迴圈迭代，讓每一子圖的柱狀圖顏色均有所不同。

```
# 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('乘客人數', rotation=0, labelpad=10, fontsize=15, color='black', fontweight='bold')
```

六、主程式

圖九：我想瞭解不同階層不同年齡層是否會影響生存人數？

- 最終，利用主程式接收所有函式，並回傳，回應所有問題。

```
# 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'''
```

圖十：主程式

四、分析結果

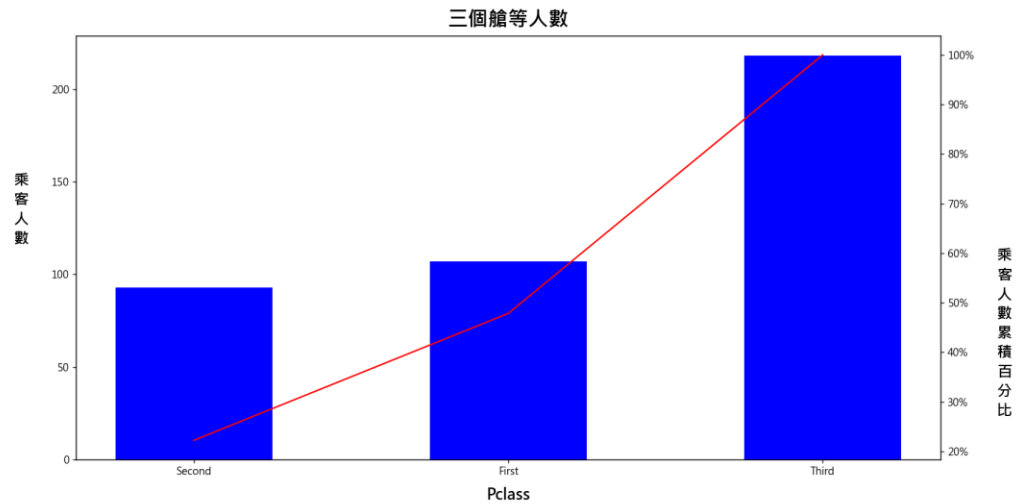
- 問題 01：我想知道不同艙等階級的人數有多少？

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

圖十一：各艙層乘客人數表(DataFrame)

問題零



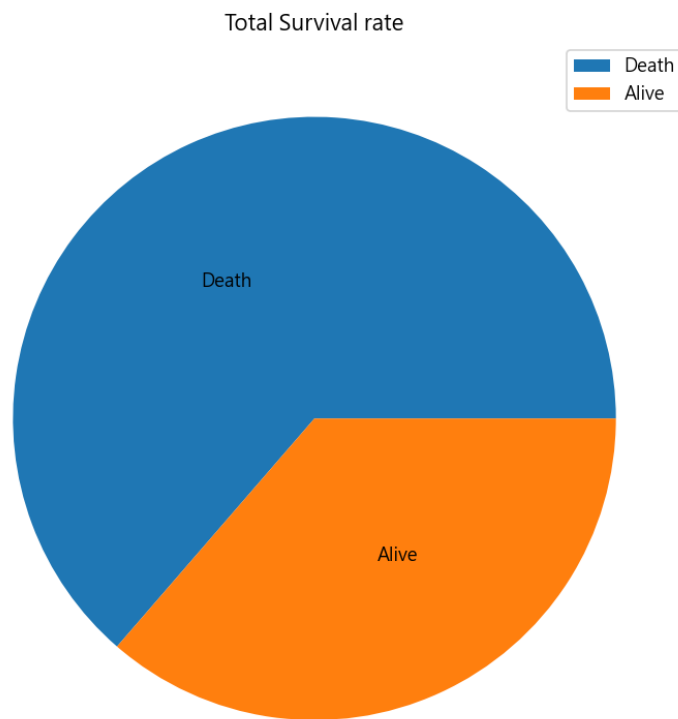
圖十二：各艙層乘客人數表(柏拉圖)

- 回應問題 01：我想知道不同艙等階級的人數有多少？
 1. 分析結果得出，二等艙人數占最小宗：93 人，但僅次頭等艙的 107 人
 2. 三等艙乘客人數最多，有 218 人
 3. 由上述可看出，鐵達尼號絕大部分的乘客屬於三等艙，而三等艙大部分的人皆是社會底層人士或是無錢購票上船的人。中產階級與社會頂層人士屬於二等艙與頭等艙，但兩者相加起來的人數仍小於三等艙的乘客。由此或許可以反映出當時英國的社會概況，從輪船的乘客概況便可推知當時的社會階級觀念是何等嚴重。
- 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？

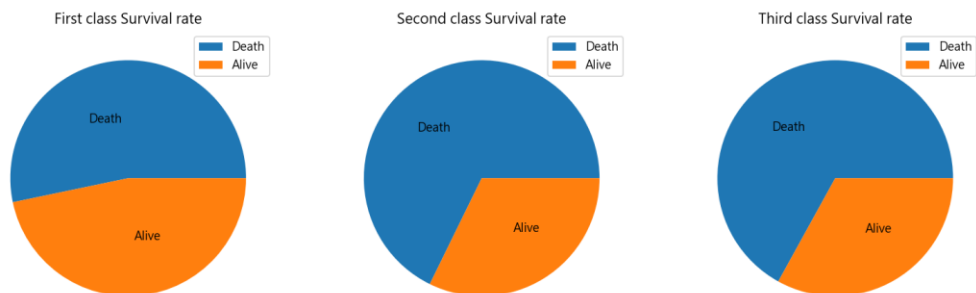
各艙層的生還與死亡率分布：

	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

圖十三：各艙層的生還與死亡率



圖十四：Total Survival rate



圖十五：三個艙層的生還率

- 回應問題 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？
 1. 得到的結論是似乎有影響，但是似乎不適那麼顯著地呈現高低排列：
 - ◆ 頭等艙：死亡率 53.27%
 - ◆ 二等艙：死亡率 67.74%

◆ 三等艙：死亡率 66.97%

並沒有照社會階級高低排列死亡順序，但可見得，頭等艙死亡人數依舊是最低的。

2. 每個艙層的死亡人數均大於 50%，可見，社會階級高低皆盡受到這次船難劇烈的影響

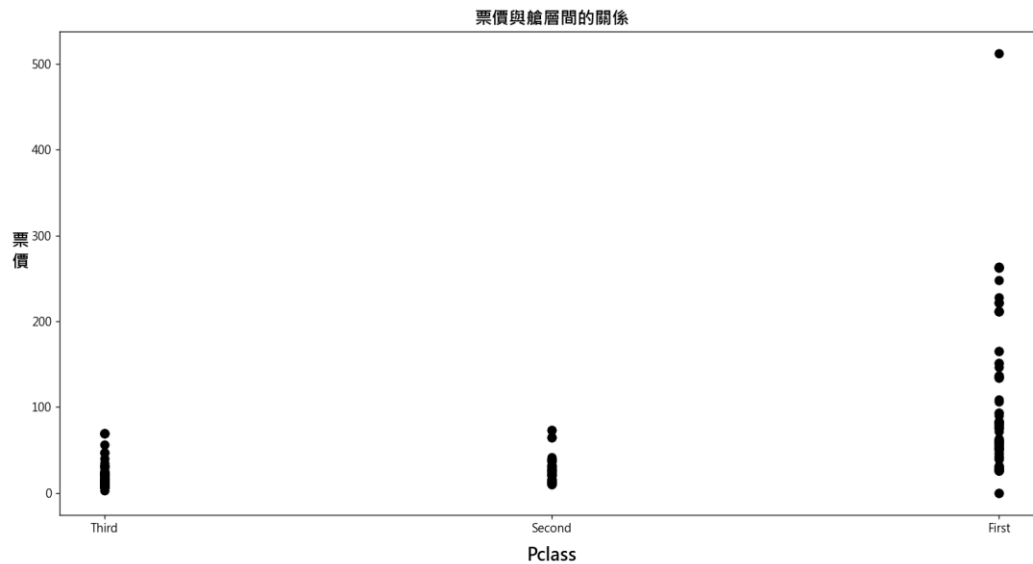
● 問題 03：我想知道票價是否與艙等有關係？

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]

圖十六：不同階層不同乘客所支付票價



圖十七：票價與艙層間的關係(散佈圖)

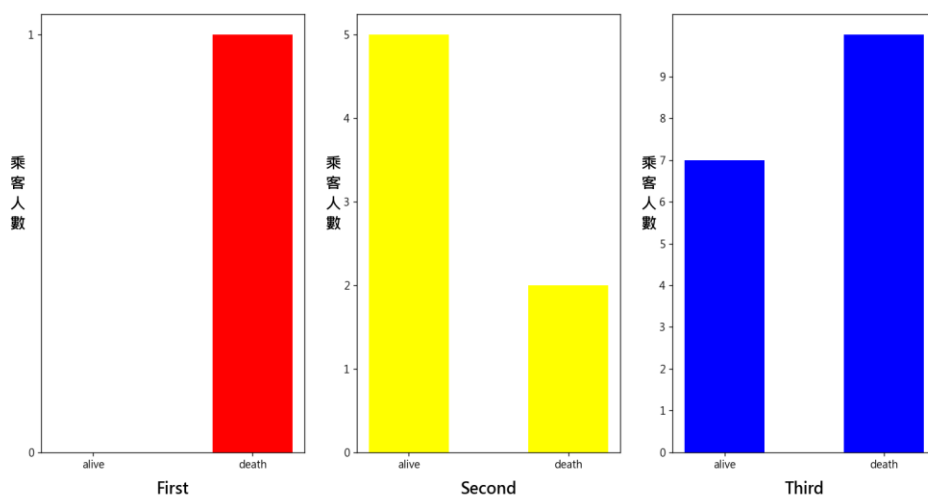
● 回應問題 03：我想知道票價是否與艙等有關係？

1. 由此圖可看出，不同艙層支付票價也不盡相同
2. 由此圖可看出，First class 到 Third class，階層由高到低，票價密集度也是由高到低的分布
3. Second class 的票價大致上與 Third class 一致，但略高於 Third

class；而 First class 離散性則最大，最大值甚至高達 500 以上。

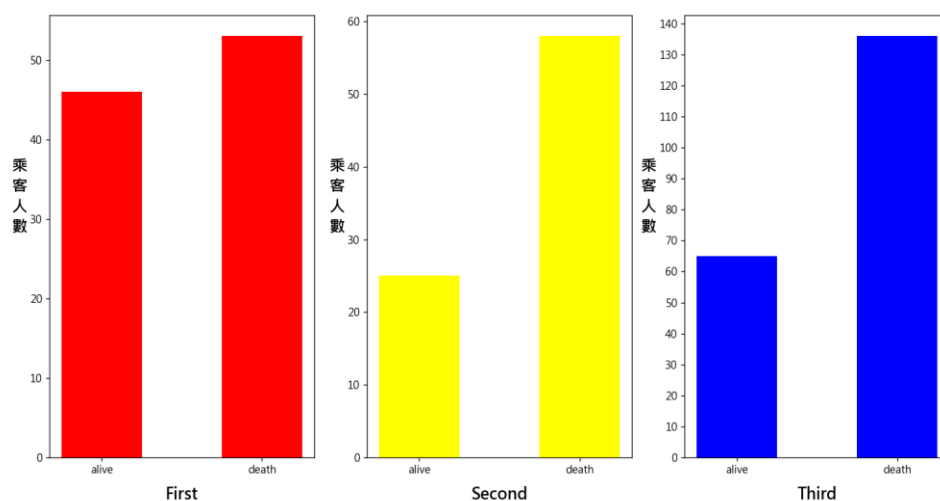
- 問題 04：我想瞭解不同階層不同年齡層是否會影響生存人數？

問題三：不同艙層的兒童存亡概況



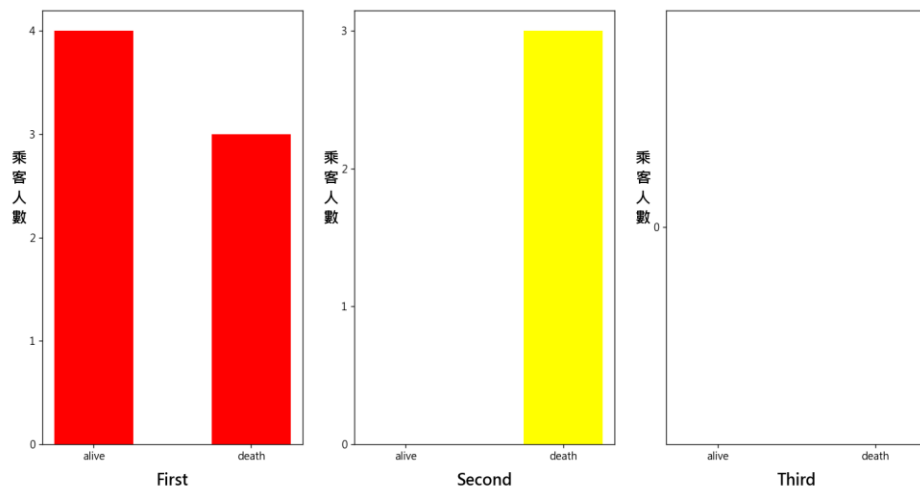
圖十八：不同艙層的兒童存亡概況

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

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

圖二十一：各艙層生還與死亡人數概況

- 回應問題 04：我想瞭解不同階層不同年齡層是否會影響生存人數？
 1. 兒童：頭等艙兒童人數僅有 1 人，所以死亡人數也僅有一人；二等艙與三等艙做比較，發現二等艙兒童生還人數大於死亡人數；三等艙兒童死亡人數高於二等艙。由此可見，在救生艇缺少的情況下，三等艙兒童反而成了被犧牲的對象。
 2. 青壯年：死亡人數均高於生還人數
 3. 老人：三等艙沒有老人，所以用頭等艙與二等艙進行分析，發現頭等艙生還人數大於死亡人數，而二等艙老人無一生還。
 4. 由上述可推知，階層/艙層高低會影響該艙層該年齡層的生存人數

五、總結

- 過去的社會階級高低，由統計數據來推論母體，推知過去人的觀念認為社會階級越是高尚，則生命越有價值；反之，社會底層成了被踐踏的對象
- 即便有「婦孺老人優先上船」一說，但各年齡層各艙層死亡人數大部分仍高於生還人數
- 三等艙佔總乘客人數最多，佔據一半以上的乘客人數，但生還人數(72 人)加起來卻沒有頭等艙與二等艙相加($30+50=80$ 人)來得多。由此可見，三等艙當年獲救機率實際上極低
- 票價反映艙務等級，更是反應個人經濟實力、社會地位

