Stat_HW01 Answer

Exercise 1.3 (9%) 每小題3%

a. 母體為當地25000個選民\ b. 樣本為其中200個被抽樣的選民\ c. statistic，因為是從樣本得到的統計值

Exercise 1.7 (9%) 每小題3%

a. 略\ b. 略\ c. 略

Exercise 2.5 (20%) 每小題4%

a. Interval\ b. Ordinal or Interval\ c. Nominal\ d. Interval\ e. Nominal

Exercise 2.11 (12%) 每小題4%

a. Nominal\ b. Interval\ c. Ordinal

Exercise 2.13 (10%) 圖5%，解釋5%

沒寫標題、座標軸全錯，解釋酌量扣分。

In [2]:
```
#匯入套件，設定圖的大小
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
plt.rcParams["figure.dpi"] = 120

def autolabel(rects):
    """Attach a text label above each bar in *rects*, displaying its height."""
    for rect in rects:
        height = rect.get_height()
        plt.annotate('{}'.format(height),
                     xy=(rect.get_x() + rect.get_width() / 2, height),
                     xytext=(0, 1),        # 1 points vertical offset
                     textcoords="offset points",
                     size=6,               # fontsize=7
                     ha='center',va='bottom')
```

In [3]:
```
#讀excel檔
df = pd.read_excel('Xr02-13.xlsx')

#看看資料
print(df)
```

```
                 Country  Oil Reserves (Barrels)
0                 Brazil             15310000000
1                 Canada            172500000000
2                  China             24650000000
3                   Iran            157800000000
4                   Iraq            144200000000
5             Kazakhstan             30000000000
6                 Kuwait            104000000000
7                  Libya             48360000000
8                Nigeria             37070000000
9                  Qatar             25240000000
10                Russia            103200000000
11          Saudi Arabia            268300000000
12  United Arab Emirates             97800000000
13         United States             36520000000
14             Venezuela            298400000000
```

In [4]:
```
#將檔案內資料由大到小排序
df = df.sort_values(by=['Oil Reserves (Barrels)'],ascending=False)

#看看資料(已排序)
print(df)
```

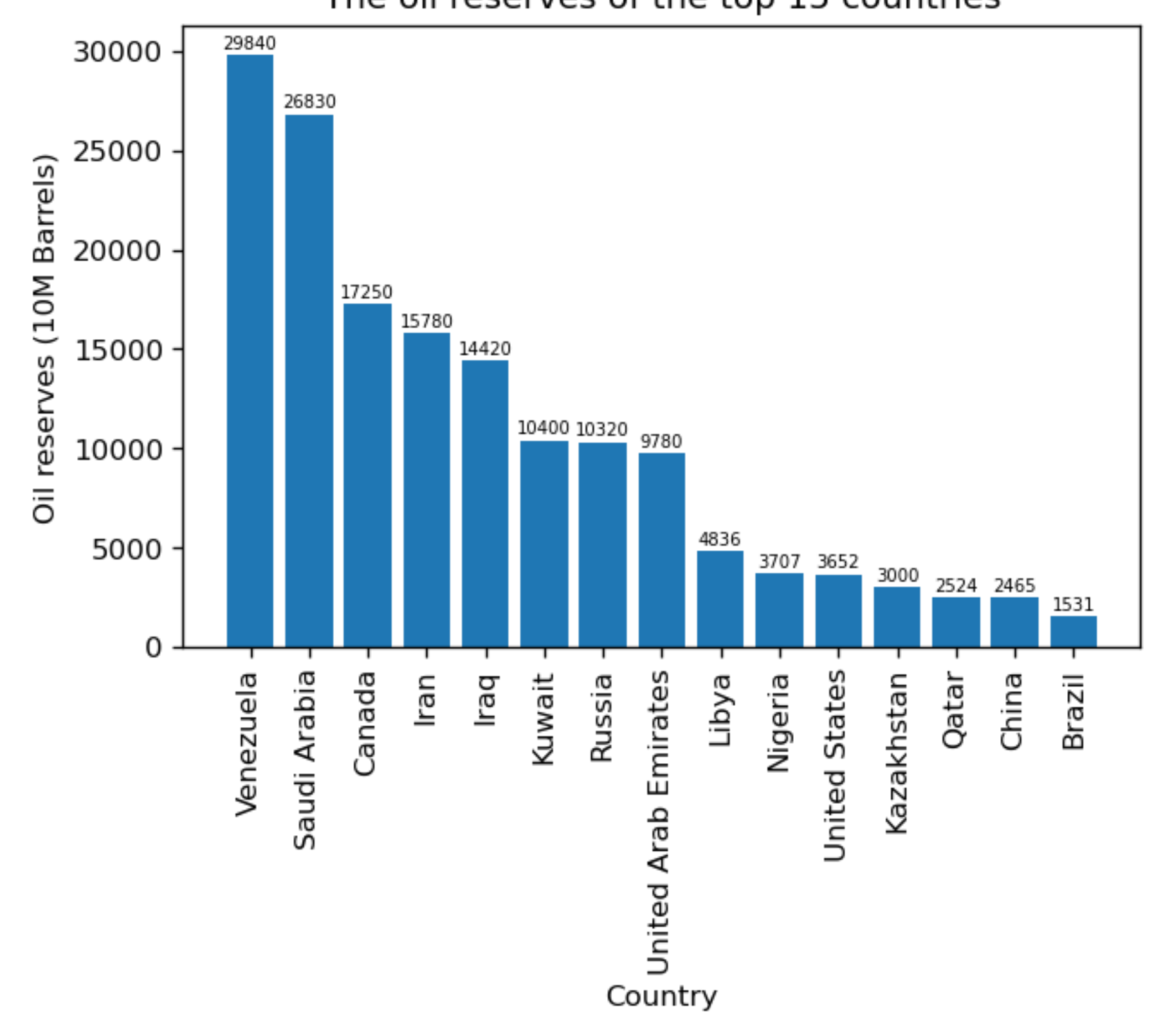
```
                 Country  Oil Reserves (Barrels)
14             Venezuela            298400000000
11          Saudi Arabia            268300000000
1                 Canada            172500000000
3                   Iran            157800000000
4                   Iraq            144200000000
6                 Kuwait            104000000000
10                Russia            103200000000
12  United Arab Emirates             97800000000
7                  Libya             48360000000
8                Nigeria             37070000000
13         United States             36520000000
5             Kazakhstan             30000000000
9                  Qatar             25240000000
2                  China             24650000000
0                 Brazil             15310000000
```

In [5]:
```
labels = df['Country']

#sizes數字過大不易在圖中表示，自行調整，記得注意單位
sizes = []
for num in df['Oil Reserves (Barrels)']:
    sizes.append(int(num/10000000))
```

In [6]:
```
fig, ax = plt.subplots()
rects = ax.bar(labels, sizes, width=0.8, bottom=None, align='center')
plt.title('The oil reserves of the top 15 countries') #記得改標題
plt.ylabel('Oil reserves (10M Barrels)') # Y軸 label 配合單位
plt.xlabel('Country')
plt.xticks(labels,rotation='vertical')   # X軸 laebl 顯示角度調整，避免字被擋到

autolabel(rects)
plt.show()
```



Exercise 2.29 (20%)

a. frequently distribution 5%, relative frequently distribution 5%

b. 圖5%，解釋5%

沒有程式碼直接有結果全錯、沒寫標題全錯，解釋酌量扣分。

In [7]:
```
df = pd.read_excel('Xr02-29.xlsx')

#看看資料
print(df)
```

```
     Newspaper
0            2
1            1
2            2
3            3
4            1
..         ...
355          1
356          1
357          2
358          1
359          1

[360 rows x 1 columns]
```

In [8]:
```
#frequency
Ncount_tab = df['Newspaper'].value_counts()
print(Ncount_tab)

#Relative frequency
Npropotion_tab = Ncount_tab/len(df['Newspaper'])
print(Npropotion_tab)
```

```
1    141
2    128
4     59
3     32
Name: Newspaper, dtype: int64
1    0.391667
2    0.355556
4    0.163889
3    0.088889
Name: Newspaper, dtype: float64
```

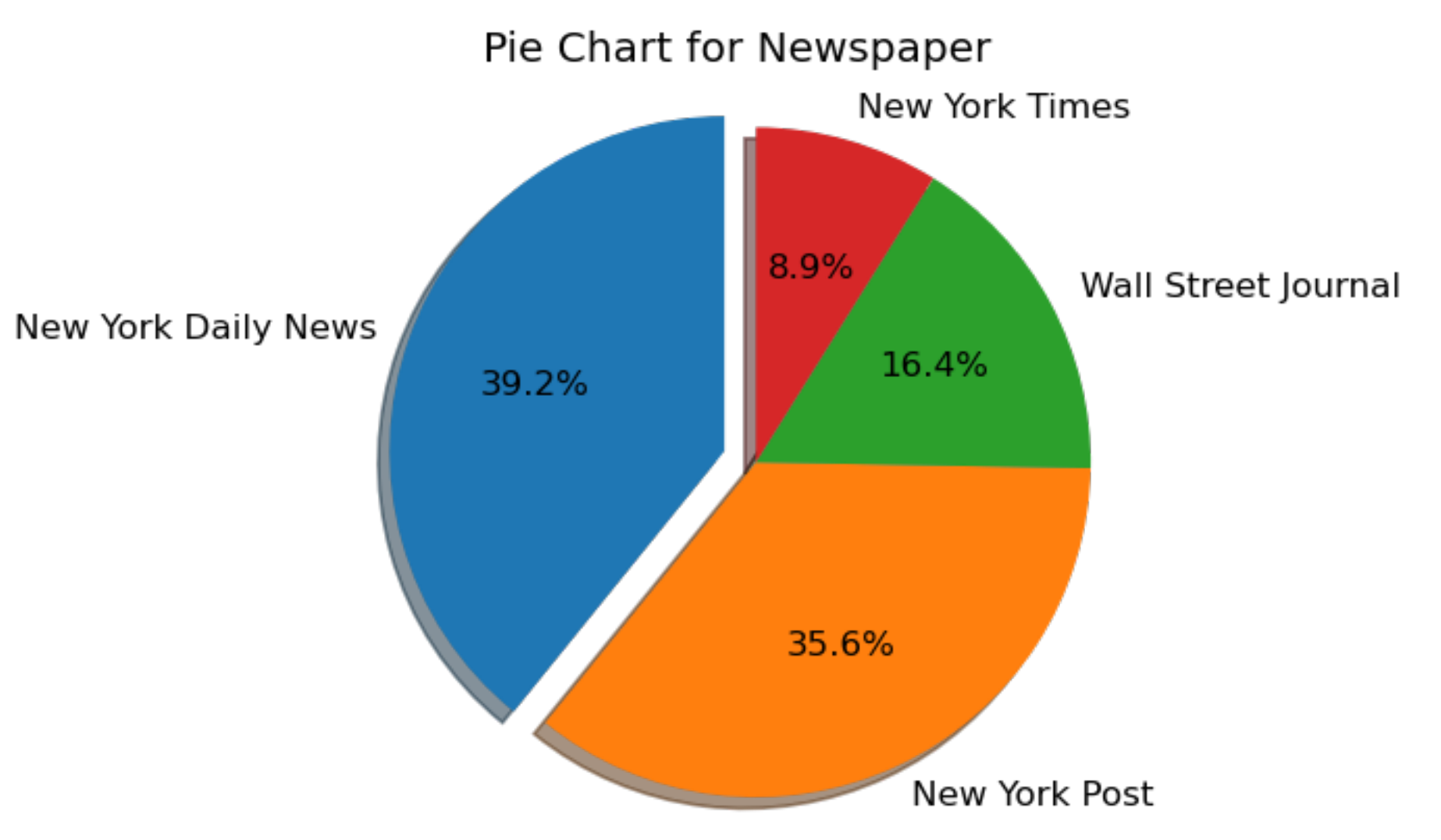
In [9]:
```
#A dataframe for Newspaper Name, Frequency, Relative Frequency
#注意 3. New Tork Times 和 4. Wall Street Journal的排序，因次數的結果顯示1>2>4>3，為了對應名稱方便，NName的順序也是1,2,4,3
NName = ("New York Daily News", "New York Post", "Wall Street Journal","New York Times")

fre_tab={"Newspaper": NName, "Frequency":Ncount_tab, "Relative Frequency": Npropotion_tab}
print(pd.DataFrame(fre_tab))
```

```
             Newspaper  Frequency  Relative Frequency
1  New York Daily News        141            0.391667
2        New York Post        128            0.355556
4  Wall Street Journal         59            0.163889
3       New York Times         32            0.088889
```

In [10]:
```
explode=(0.1, 0, 0, 0)

fig, ax = plt.subplots()
ax.pie(Ncount_tab,explode=explode, labels=NName, autopct='%1.1f%%', shadow=True, startangle=90)
ax.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.title('Pie Chart for Newspaper')
plt.show()
```



Exercise 2.41 (10%) 圖5%，解釋5%

沒寫標題、座標軸全錯，解釋酌量扣分。

In [11]:
```
df = pd.read_excel('Xr02-41.xlsx')

#看看資料
print(df)
```

```
            Education   2000   2005   2010   2015
0  Less than high school  27854  28017  25711  24582
1            High school  58086  60893  62456  62575
2           Some college  44445  48076  53920  56031
3       College graduate  44845  52381  59840  68945
```

In [12]:
```
Y_label = list(df.columns)[1:]
```

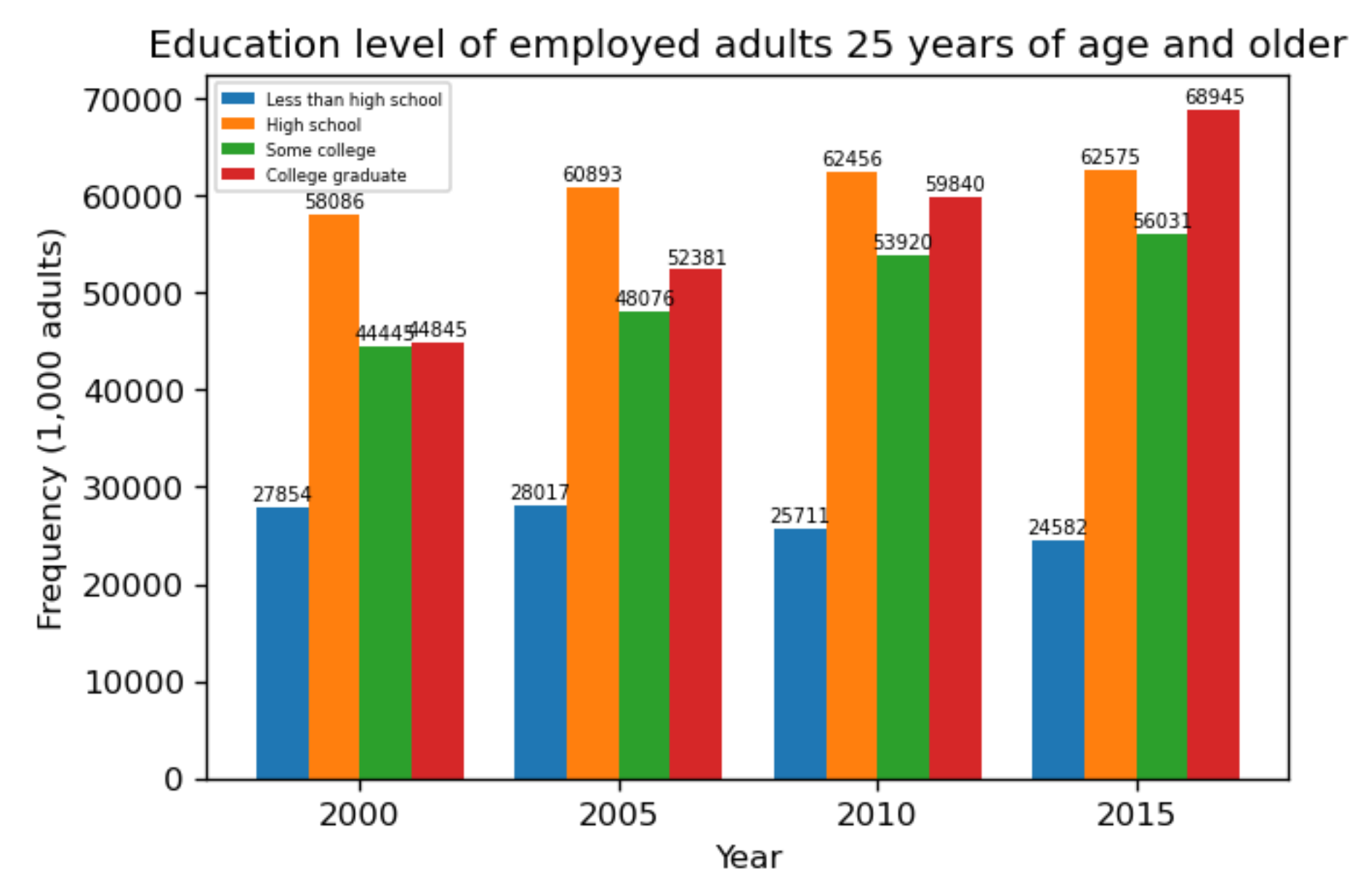
In [14]:
```
x = np.arange(len(Y_label))  # the label locations
width = 0.2                  # the width of the bars
fig, ax = plt.subplots()

#根據資料調整rects數，X軸上位置，以及label
rects1 = ax.bar(x - 3*width/2, df.iloc[0][Y_label], width, label = "Less than high school")
rects2 = ax.bar(x - width/2,   df.iloc[1][Y_label], width, label = "High school")
rects3 = ax.bar(x + width/2,   df.iloc[2][Y_label], width, label = "Some college")
rects4 = ax.bar(x + 3*width/2, df.iloc[3][Y_label], width, label = "College graduate")

# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('Frequency (1,000 adults)')
ax.set_xlabel('Year')
ax.set_title('Education level of employed adults 25 years of age and older')
ax.set_xticks(x)
ax.set_xticklabels(Y_label)
ax.legend(fontsize=5)         #可調整大小

autolabel(rects1)
autolabel(rects2)
autolabel(rects3)
autolabel(rects4)

plt.show()
```



Exercise 2.51 (10%) 圖5%，解釋5%

沒寫標題、座標軸全錯，解釋酌量扣分。

In [15]:
```
df = pd.read_excel('Xr02-35.xlsx')

#看看資料
print(df)
```

```
      Share  Political View
0         3               2
1         3               3
2         1               1
3         3               3
4         3               3
...     ...             ...
1096      2               3
1097      3               1
1098      3               3
1099      3               2
1100      1               1

[1101 rows x 2 columns]
```

In [16]:
```
#crosstab:分群在前，選項在後
contb = pd.crosstab(  df["Political View"],df["Share"])
print(contb)
```

```
Share             1   2    3   4
Political View
1               122  81  187  18
2                70  39  236   7
3                34  41  259   7
```

In [17]:
```
P_label = ("Conservative", "Moderate", "Liberal")
```

In [18]:
```
x = np.arange(len(P_label))  # the label locations
width = 0.2                  # the width of the bars
fig, ax = plt.subplots()

#根據資料調整rects數，X軸上位置，以及label
rects1 = ax.bar(x - 3*width/2, contb[1], width, label = "Fair share")
rects2 = ax.bar(x - 1*width/2, contb[2], width, label = "Too much")
rects3 = ax.bar(x + 1*width/2, contb[3], width, label = "Too Little")
rects4 = ax.bar(x + 3*width/2, contb[4], width, label = "No opinion")

# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('Frequency')
ax.set_xlabel('Political view')
ax.set_title('Frequency by Share and Political View')
ax.set_xticks(x)
ax.set_xticklabels(P_label)
ax.legend()

autolabel(rects1)
autolabel(rects2)
autolabel(rects3)
autolabel(rects4)

plt.show()
```