Statistics - Final team project - Group 10

110550085房天越、110550089周冠辰、110550096梁恩齊、110550122柯凱軒

1. Motivations & our questions

(1) 資工研究所報考人數與工程師薪資的相關性

近年來，隨著科技的快速發展，科技產業的前景一片看好，薪水也逐年攀升，使工程師成為許多學子心中夢想的職業。而能夠進入頂尖大學的研究所就讀並取得學位即被視為取得踏入科技業的門票，也因此在國內的研究所相關科系報考人數逐年增加。

我們想探討的是，歷年報考交大資工研究所的人數是否與歷年的資訊工程師薪水有正相關，並以此推理出人們目前對於科技業的追求是否是盲目的跟隨熱潮，亦或是對高薪工作的合理追求。

(2) 大學與研究所學歷對社會新鮮人薪資的影響

即將邁入大學最後一年的我們，也必須在邁入就業市場以及繼續升學之間做出抉擇。因此我們也會探討大學學歷以及研究所學歷在初任就業市場上對於薪資的影響，從中探討多花兩年以上的時間所得到的研究所文憑是否是值得的投資。

2. The process of data collection

(1) 資工研究所報考人數與工程師薪資的相關性

我們的資料從網路上的各個網站進行蒐集。歷年的工程師薪資透過勞動部網站的職類別薪資調查動態查詢進行蒐集(https://pswst.mol.gov.tw/psdn/)，我們從中所有選項中挑選了所有最有可能是資工相關的項目，包括: 電機工程師、電子工程師、電信工程師、資訊系統分析及設計師、軟體開發及程式設計師以及資料庫及網路專業人員作為資訊工程師的行業別，並依據人數進行加權平均以計算出資訊工程師的平均薪水。

而歷年報考交大資工研究所的人數則是從官方網站以及各補習班的統計數據蒐集。有了歷年的資訊工程師薪水以及歷年的交大資訊聯招報考人數，我們便能計算其相關係數並進一步的分析。

我們取得了大約十年的數據(除了缺少107年交大資訊聯招報考人數以外)。而我們的統計誤差主要來自政府統計薪資資料的誤差，以及我們樣本數的不足，可能樣本分布不會近似於常態分佈。

資料名稱	資料型態	樣本數量
歷年資訊工程師薪資	數字	11
歷年交大資聯報考人數	數字	11

為了考慮通貨膨脹帶來的影響，我們同時取得了100年至112年的消費者物價指數（CPI），接著以100年的薪資為準，依照各年份CPI與100年CPI的比值，調整了各年份的平均薪資，以獲得更加準確的結果。

(2) 大學與研究所學歷對社會新鮮人薪資的影響

在勞動部的資料庫網站上，我們同樣可以爬取歷年初任人員在不同產業以及不同學歷時的薪資分布。而因為每一年的每一種產業都會對應到一組的大學學歷薪資以及研究所學歷薪資，每一年有20組不同產業的資料，並且我們取了10年份的內容，也因此我們會有200組樣本可以比較研究所學歷以及大學學歷是否有差距。而統計誤差也同樣來自於政府提供的數據誤差，以及樣本數可能偏少，使分布可能不會近似於常態分布。

資料名稱	資料型態	樣本數量
各年不同產業大學學歷初任人員薪資	數字	200
各年不同產業研究所學歷初任人員薪資	數字	200

從政府網站取得的資料格式為xlsx文件，我們是會先將文件一份一份加入額外的first row header使其更改為能轉變成csv文件的格式，再轉成csv文件後，透過python的panda library讀取並取出我們會使用到的資料。

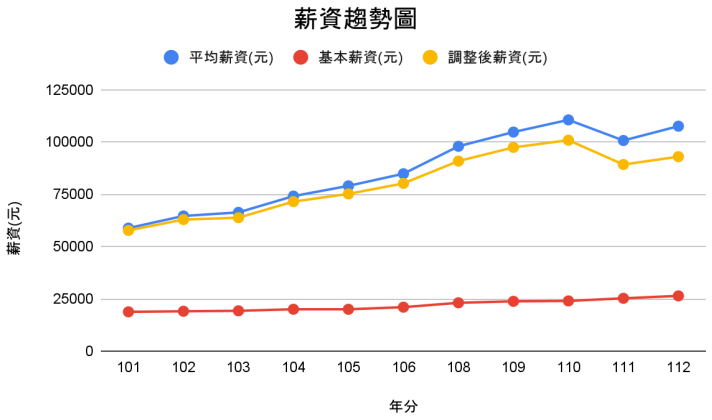
我們整理後的資料存放在下列的雲端資料夾內:

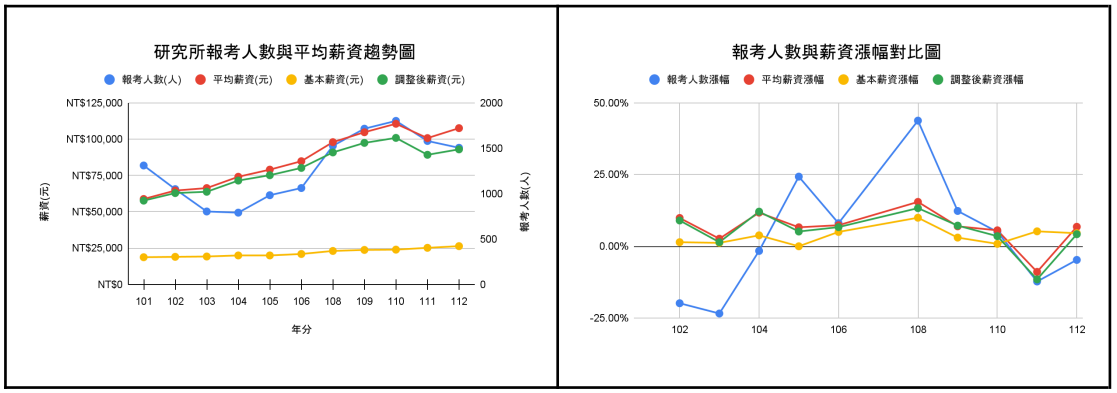
https://drive.google.com/drive/folders/18SOH605Y_CRMGXro-tw_rIsrwSo49-YN

3. Descriptive data analysis (codes & results & descriptions)

(1) 資工研究所報考人數與工程師薪資的相關性

依照資料我們能夠得到11年份的研究所報考人數與各種薪資的折線圖，並從中觀察到數據變化的趨勢。我們計算所使用的薪資已與通膨比率進行調整過。由圖可見平均薪資平穩上漲，而研究所報考人數起起落落，但整體呈增加趨勢。透過計算漲幅(今年減去前一年後除以前一年)，我們能夠更清楚的觀察趨勢。

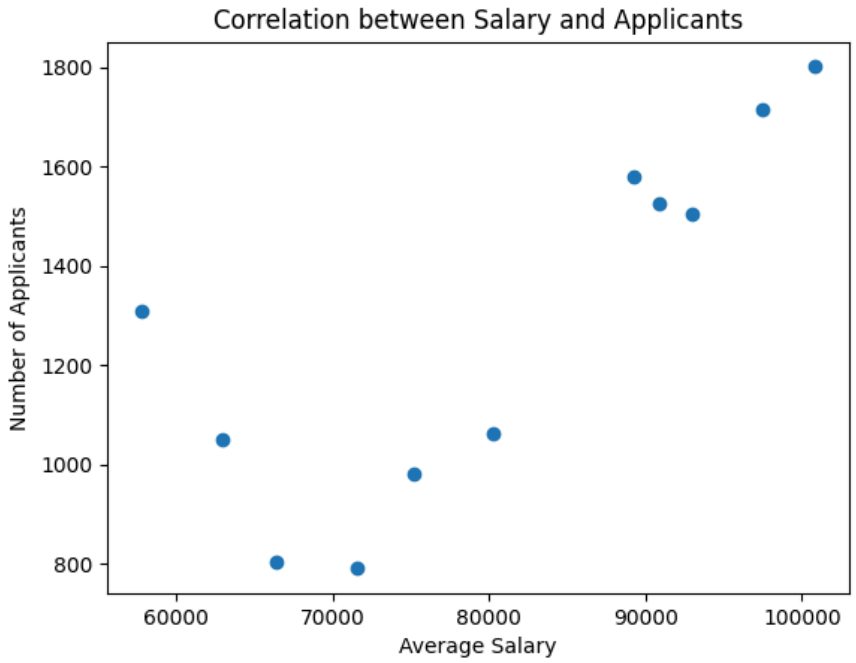




上述的資料是使用google試算表進行繪製。

而我們也將每一組資料使用scatter diagram表示以觀察報考人數與薪資的相關性。

```
import matplotlib.pyplot as plt
plt.scatter(salaries, applicants)
plt.xlabel('Average Salary')
plt.ylabel('Number of Applicants')
plt.title('Correlation between Salary and Applicants')
plt.show()
```

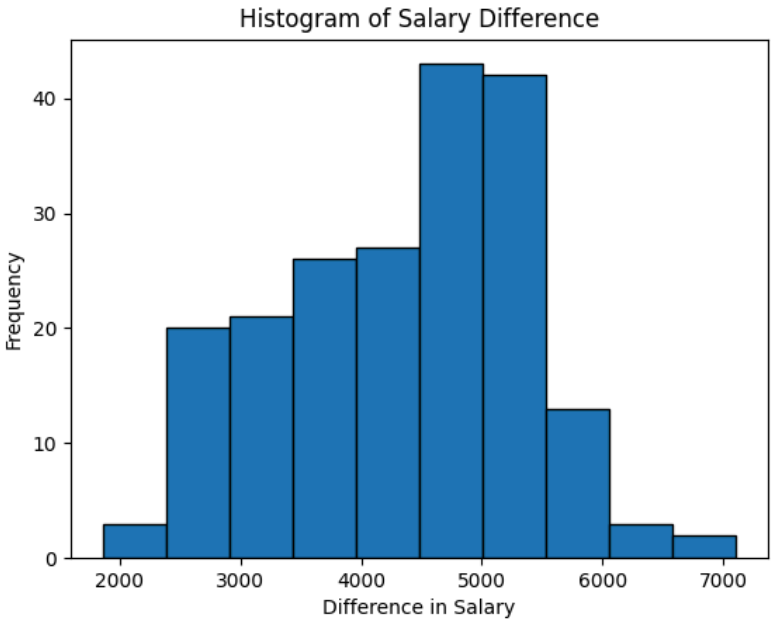


從圖中可以看出隨著薪資與報考人數大致呈現正相關。

(2) 大學與研究所學歷對社會新鮮人薪資的影響

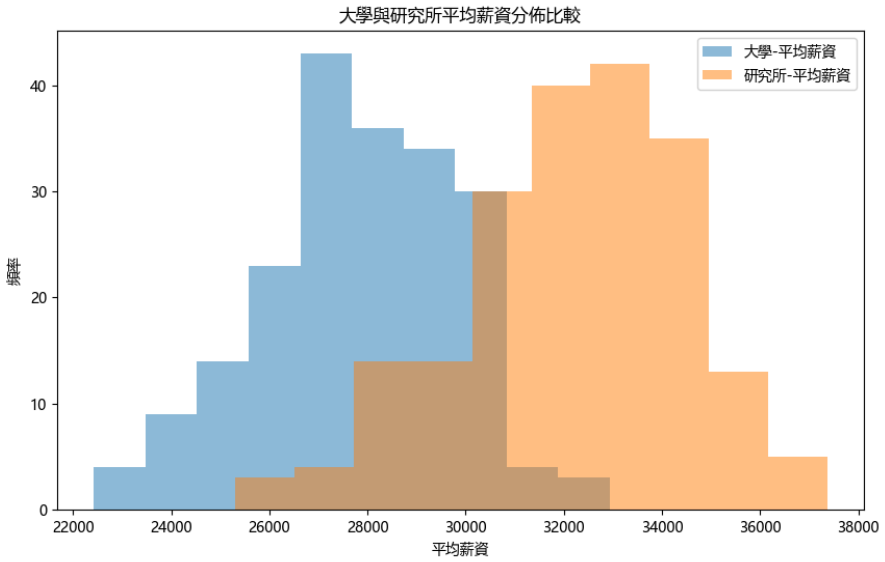
我們使用histogram將每一筆薪資差樣本的分布給表示出來。

```

plt.hist(df['Diff'], bins=10, edgecolor='black')
plt.xlabel('Difference in Salary')
plt.ylabel('Frequency')
plt.title('Histogram of Salary Difference')
plt.show()

```



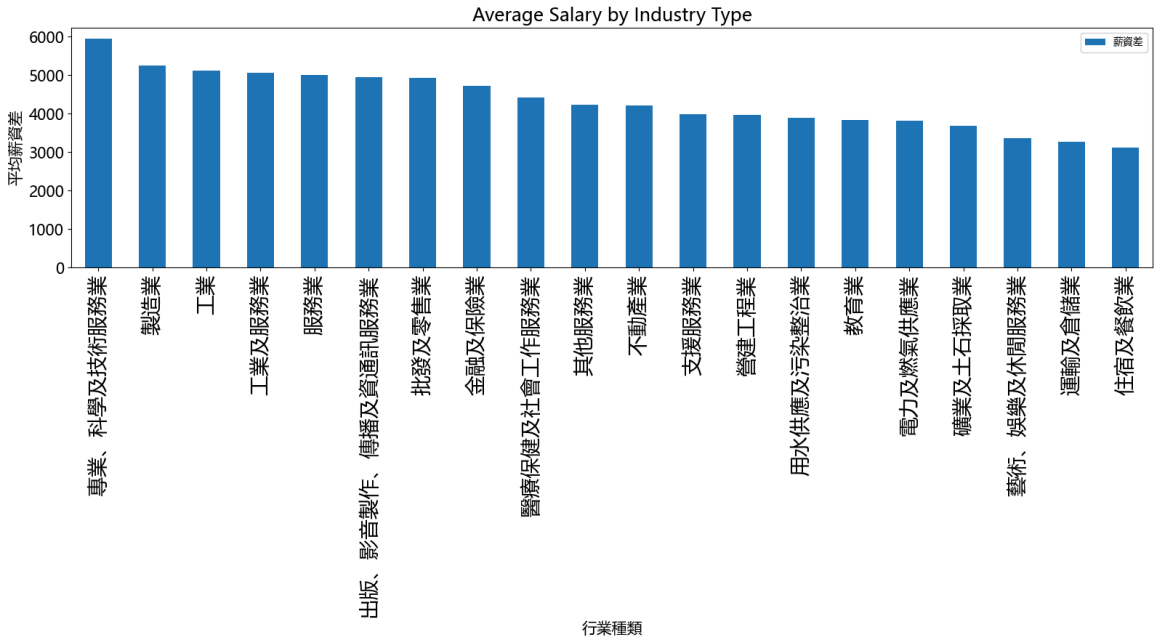
從圖中可以看出研究所學歷初任人員薪水與大學學歷初任人員薪水的差是大於零的，且大約平均值是在4000~5000之間。

```python
plt.figure(figsize=(10, 6))
plt.hist(all_data['大學-平均薪資'], bins=10, alpha=0.5, label='大學-平均薪資')
plt.hist(all_data['研究所-平均薪資'], bins=10, alpha=0.5, label='研究所-平均薪資')
plt.title('大學與研究所平均薪資分佈比較')  # 圖形標題
plt.xlabel('平均薪資')  # X軸標籤
plt.ylabel('頻率')  # Y軸標籤
plt.legend()  # 顯示圖例
plt.show()  # 顯示圖形
```



我們也計算了各個行業歷年的薪水差的平均值並使用bar chart展示。

```
# prepare the data
all_data = pd.DataFrame()
year = 109
file_path = f'./firstsalary_csv/{year}_firstsalary.csv'
df = pd.read_csv(file_path)
all_data['行業種類'] = df['行業種類']
# initialize the column with 0
all_data['大學-平均薪資'] = 0
all_data['研究所-平均薪資'] = 0
all_data['薪資差'] = 0
print(all_data)
# calculate the average salary for each industry type
for year in range(100, 110):
    file_path = f'./firstsalary_csv/{year}_firstsalary.csv'
    df = pd.read_csv(file_path)

    for col in df.columns[1:]:
        if df[col].dtype == 'object':
            df[col] = df[col].str.replace(',', '').replace('-', np.nan).astype(float)

    all_data['大學-平均薪資'] += df.iloc[:, 1]
    all_data['研究所-平均薪資'] += df.iloc[:, 5]
    all_data['薪資差'] += df.iloc[:, 5] - df.iloc[:, 1]
all_data['大學-平均薪資'] /= 10
all_data['研究所-平均薪資'] /= 10
all_data['薪資差'] /= 10

# 先將資料排序
all_data = all_data.sort_values(by='薪資差', ascending=False)

# plot as bar chart
plt.rcParams['font.family'] = 'Microsoft YaHei'  # 或 'SimHei'
# plot as bar chart
all_data.plot(x='行業種類', y=['薪資差'], kind='bar', fontsize=20)
plt.rcParams['font.size'] = 15
plt.yticks(fontsize=15)
plt.xticks(rotation=90, fontsize=20)
plt.xlabel('行業種類', fontsize=15)
plt.ylabel('平均薪資差', fontsize=15)
plt.title('Average Salary by Industry Type')
plt.show()
```



從圖中可以看出不同行業確實是會有不同的薪水差，像是以專業、科學及技術服務業為例，在研究所階段所額外獲得的知識以及研究能力或許會對接下來的工作產生較大的影響。而像是住宿及餐飲業等較為勞力密集的產業可能對於學歷則是較為不注重。

4. Statistic test (codes & results & descriptions)

(1) 資工研究所報考人數與工程師薪資的相關性

我們使用對於相關係數的假設檢定。我們想要推得的是報考人數與薪資是否是正相關。以每一年的資料為樣本，設p為真實的相關係數，r為抽樣到的相關係數，假設: H0: p = 0, Ha: p > 0，並計算樣本相關係數來檢測。

相關係數的假設檢定參考自(https://online.stat.psu.edu/stat501/lesson/1/1.9)，其中使用的公式為:

$$\text{Test statistic: } t^* = \frac{r\sqrt{n-2}}{\sqrt{1-R^2}}$$

此公式是與相關係數為0進行比較。

```python
import numpy as np
from scipy import stats
from scipy.stats import pearsonr

# 工程師薪水資料
salaries = [
    57719.112, 62910.630, 66340.650, 71510.873, 75193.414,
    80217.657, 90881.242, 97430.713, 100869.383, 89239.011, 92991.865
]
# 碩班報考人數
applicants = [
    1310, 1050, 804, 791, 983, 1062, 1527, 1715, 1801, 1580, 1505
]
# 計算皮爾森相關係數
correlation, _ = pearsonr(salaries, applicants)
print(f"The correlation coefficient is: {correlation:.4f}")

# Calculate the test statistic
n = len(salaries)
t = correlation * np.sqrt(n - 2) / np.sqrt(1 - correlation**2)
print(f"The test statistic is: {t:.4f}")

# Calculate the P-value
p = 2 * (1 - stats.t.cdf(t, df=n-2))
print(f"The P-value is: {p:.4f}")

# Set the significance level and make the decision
alpha = 0.05
if p < alpha:
    print("Reject the null hypothesis")
else:
    print("Fail to reject the null hypothesis")
```

在這段程式中，首先，我們將需要用到的歷年的薪資和報考人數放入list。

接著使用scipy.stats裡的pearsonr計算皮爾森相關係數。

考慮通膨，我們得到的結果是correlation = 0.7946

接著，使用上述test statistic可得 t* = 3.9260

最後，計算對應的P-value，可得 P-value = 0.0035

令α = 0.05，則 P-value < α，故reject H0，可知有強烈證據表明歷年資訊工程師的工資與交大研究所報考人數的皮爾森相關係數大於0，存在正相關。

另外，若不考慮通膨，以相同方式計算，可得皮爾森相關係數為0.8006，t* = 4.0089，P-value = 0.0031 < α。

一樣可知歷年工程師工資與交大研究所報考人數存在正相關。

(2) 大學與研究所學歷對社會新鮮人薪資的影響

我們使用歷年各行業的研究所與大學初任人員薪資差為一組樣本，並計算薪資差的平均值μd，假設: H0: μd <= 0, Ha: μd > 0 (我們也比較了差值在1000, 2000, 3000, 4000, 5000的情況)，並因為我們的樣本數夠多，使用paired z-test來進行檢定。

```python
import pandas as pd
import numpy as np
import os
import scipy.stats as stats
from scipy.stats import norm

# 假設所有 CSV 檔案都放在同一資料夾下
folder_path = 'firstsalary_csv'
files = os.listdir(folder_path)

# 建立一個空的 DataFrame 用於之後的資料合併
all_data = pd.DataFrame()

for file in files:
    if file.endswith('.csv'):
        # 讀取 CSV 檔案
        file_path = os.path.join(folder_path, file)
        df = pd.read_csv(file_path)

        for col in df.columns[1:]:
            if df[col].dtype == 'object':
                df[col] = df[col].str.replace(',', '').replace('-', np.nan).astype(float)
        # 從檔案名稱提取年份，假設檔案名稱格式為 "YYYY_data.csv"
        year = file.split('_')[0]

        # 新增年份列
        df['Year'] = year

        df['Diff'] = df['研究所-平均薪資'] - df['大學-平均薪資']

        # 將當前 DataFrame 加入到總 DataFrame
        all_data = pd.concat([all_data, df], ignore_index=True)

# 顯示合併後的資料
df = all_data

# use df['Diff] to do statistical test
z = (df['Diff'].mean() - 0) / (df['Diff'].std() / np.sqrt(len(df)))

# calculate p-value with 1 tail
p = 1 - stats.norm.cdf(z)
print(f"Z-score: {z}")
print(f"P-value: {p}")
```

首先是會從各個資料文件讀取數據，並合併到同一個變數下。在讀取資料的同時會計算每一對資料的差值。最後會計算出這些差值的平均值、標準差，並計算Z的值，並透過Z的值估算出P-value。

Statistic Test	Z	P-value
H0: μd <= 0, Ha: μd > 0	60.116	0.0
H0: μd <= 1000, Ha: μd > 1000	46.260	0.0
H0: μd <= 2000, Ha: μd > 2000	32.403	0.0
H0: μd <= 3000, Ha: μd > 3000	18.547	0.0
H0: μd <=4000, Ha: μd > 4000	4.690	1.365e-06
H0: μd <= 5000, Ha: μd > 5000	-9.166	1.0

令α = 0.05，當假設值在4000以下時，p-value皆小於α，而5000時，則大於α，則我們有足夠的證據表示研究所學歷的初任人員薪資是大於大學學歷的初任人員薪資，並且差距在4000以上。

接著，我們令H0: μd >= 5000, Ha: μd < 5000，將P-value的計算方式改為:

```
# calculate p-value with 1 tail
p = stats.norm.cdf(z)
```

可得 Z = -9.166，P-value = 2.444e-20。

一樣令α = 0.05，P-value < α，故reject H0。

因此，結合上述分析，我們有足夠證據說明研究所學歷的初任薪資平均大於大學學歷的初任薪資，且差距在4000~5000之間。

若是使用信賴區間分析，我們可使用：

```
print(df['Diff'].mean())
```

算出d_bar = 4338.475。

並使用：

```
print(df['Diff'].std())
```

算出sd = 1020.610

接著應用這個公式：

$$\bar{d} - t_{\alpha/2,n-1}s_d/\sqrt{n} \le \mu_D \le \bar{d} + t_{\alpha/2,n-1}s_d/\sqrt{n}$$

將式子中的t換成Z，可得到信賴區間為

4197.045 <= μd <= 4479.905

與上述分析吻合，且取得了更精確的區間。

5. **The conclusion from the analysis**

透過以上的統計分析，我們透過Descriptive Statistics了解到考研人數以及工程師薪資變化的趨勢，並透過分布圖可以大致看出兩者的相關性。再透過Inferential Statistics所進行的假設驗證，了解到人們報考資工研究所的動機與工程師的薪水的關聯。具體而言，我們透過相關係數的驗證，發現報考資工研究所的人數增加與工程師薪資水平上升呈正相關。

此外，我們也探討了大學與研究所學歷對初任就業市場薪資的影響，以評估研究所學歷對於薪資水平的加成效應。透過Descriptive Statistics，我們觀察到了研究所學歷的初任人員薪水分布是高於大學學歷的初任人員薪水分布，同時我們也觀察到不同的行業會有不一樣的薪水差距。接著我們透過Inferential Statistics，透過Paired-z-test了解到，研究所學歷的初任薪資確實大於大學學歷的初任薪資，並且平均差距在4000~5000之間。

6. Bonus

因為已知今年的報考人數卻不知道今年的平均薪資，所以改為預測今年的平均薪資，方法為使用報考人數、年份和當年的平均薪資作為特徵來預測下一年的平均薪資，以下為具體步驟:

整理數據:

- 匯總每年的報考人數和平均薪資數據。
- 將這些數據整合成特徵和目標變量。

```
# 匯總數據，將報考人數與平均薪資合併在一起
merged_df = applicants_df.merge(salary_df, left_on='年分', right_on='Year', how='inner')
merged_df = merged_df[['年分', '報考人數', 'Average Salary']]

# 準備特徵和目標變量，使用當年的平均薪資來預測下一年的平均薪資
merged_df['Next Year Salary'] = merged_df['Average Salary'].shift(-1)
merged_df = merged_df.dropna()

# 特徵包括報考人數、年份和當年的平均薪資
X = merged_df[['報考人數', '年分', 'Average Salary']]
y = merged_df['Next Year Salary']
```

訓練模型:

- 使用多變量線性回歸模型進行訓練。

進行預測:

- 使用訓練好的模型來預測113年的平均薪資。

```
# 準備113年的特徵
year_113_applicants = 2448
year_113 = 113
year_112_salary = 107571.22739745102
features_113 = np.array([[year_113_applicants, year_113, year_112_salary]])

# 預測113年的平均薪資
predicted_salary_113 = model.predict(features_113)

print(f"Predicted Salary for 113: {predicted_salary_113[0]}")
```

```
Predicted Salary for 113: 125819.66364665842
```