

**國立高雄科技大學智慧商務系一甲**

**110 學年第 2 學期期末專題**

**資料科學產業職位薪資分布**



**NKUST**

**指導教師：謝文川 老師**

**專題學生：郭粟閣**

**中華民國 110 年 6 月**

## 摘要

|                    |    |
|--------------------|----|
| 摘要 .....           | 3  |
| 第壹章、緒論 .....       | 3  |
| 前言 .....           | 3  |
| 研究動機.....          | 3  |
| 研究目的.....          | 4  |
| 研究流程.....          | 4  |
| 本系統開發使用軟硬體設備 ..... | 4  |
| 計畫時程.....          | 5  |
| 第貳章、文獻探討 .....     | 6  |
| 資料視覺化.....         | 6  |
| 資料科學.....          | 6  |
| 第參章、系統架構與設計 .....  | 7  |
| 系統架構與設計 .....      | 7  |
| 第肆章、系統實作與展示.....   | 7  |
| 抓取資料.....          | 7  |
| SQL 寫入 .....       | 9  |
| 圖表呈現.....          | 10 |

# 摘要

自從來到了智慧商務系後,接觸了許多的不同的科目,從計算到眼花的會計,到設計出漂漂亮亮的前端網頁、複雜深奧的程式後端都有涉及。在這之中印象較為深刻的就是 python 的資料處理。

因此我很好奇假如之後需要做分析大數據、資料科學的工作的話收入究竟有多少? 於是我就寫了這個程式不僅可以對之後的收入有個估值也能練習操作資料的能力,使用 Pandas 的 DataFrame 結合 matplotlib 強大的功能整合出圖表,可以清楚知道各個國家、不同職業的薪水分布,所以這次以這個來作探討

**關鍵詞: Python、資料科學、matplotlib、Pandas**

## 第壹章、緒論

### 前言

資料科學(英語: data science)又稱數據科學,是一門利用資料(數據)學習知識的學科,其目標是通過從資料中提取出有價值的部分來生產資料產品,學科範圍涵蓋了:資料取得、資料處理、資料分析等過程,舉凡與資料有關的科學均屬資料科學。透過應用這學期學習的 Python 來活用資料。

### 研究動機

要完成一個好的資料專案,靠的不能只是一個厲害的強者,需要的是一支合作無間的資料團隊。資料思維是一種跨領域宏觀視野下的資料應用。我們可以觀察近期幾個市場熱門的議題來,都不乏大數據應用的身影。其中,跨領域的整合也是另一個重要的應用關鍵。無論資料的多寡,資料專案都是建基在資

訊、統計、視覺化等不同的領域專業上面。不過現實層面上來說，很難有人可以同時具備那麼多能力，因此在資料專案中更需要團隊合作。

## 研究目的

本專題藉由這次機會,了解更多有關資料科學職位的資訊,並結合 DataFrame 及 matplotlib 以圓餅圖的方式呈現,得到每個區域的薪資分布,分析不同地區的同樣工作的收入。

期許本研究能解決下列問題:

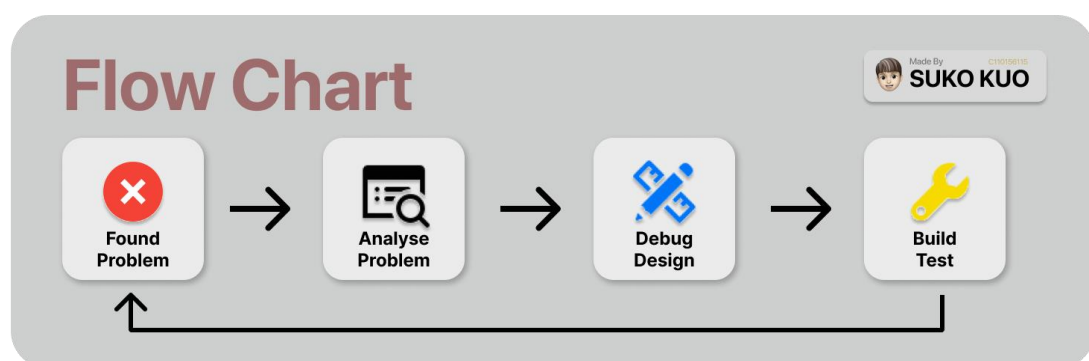
### 1. 不同國家的薪資分布:

如果能知道在各個國家的薪資分配,在之後有幸去國外工作時,可以選擇平均薪資較高的。

### 2. 各個職業的占比:

如果能從中得知各個職業的占比的話,可以去了解市場上那些職業有空缺,了解目前的趨勢。

## 研究流程



## 本系統開發使用軟硬體設備



Edition: **Windows 11 Pro**



Browser: **Chrome** v102 64bit

資料庫軟體



DB:

**mysqlnd**  
8.01.5



PHP Version:

**5.7.32**

開發軟體/語言



Dev Env:

**VsCode**  
v1.68.1



Language:

**Python**  
v3.9.7 64bit

## 計畫時程

| 計畫       | 2022 |      |      |      |      |      |
|----------|------|------|------|------|------|------|
|          | 6/15 | 6/17 | 6/18 | 6/19 | 6/20 | 6/21 |
| 專題<br>時程 |      |      |      |      |      |      |
| 定義<br>問題 |      |      |      |      |      |      |
| 分析<br>問題 |      |      |      |      |      |      |
| 實體<br>設計 |      |      |      |      |      |      |
| 建構<br>測試 |      |      |      |      |      |      |

# 第貳章、文獻探討

## 資料視覺化

為了清晰有效地傳遞資訊，資料視覺化將資料此用統計圖形、資訊、圖表、和其他工具呈現，方便於分析。

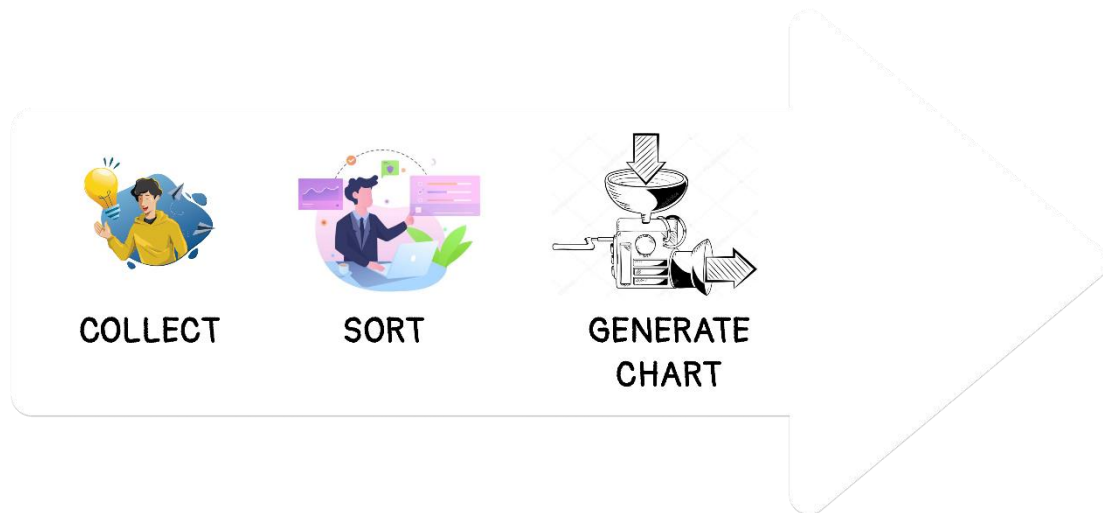
## 資料科學

資料科學橫跨多個領域，包括統計、科學方法、人工智慧（AI）和資料分析，目的在於從資料中發現價值。實作資料科學的人員稱為資料科學家。他們會搭配運用一系列的技能，分析從網路、智慧型手機、客戶、感應器和其他來源收集到的資料，並從中獲取可行的見解。



# 第參章、系統架構與設計

## 系統架構與設計



### 1. 蒐集:

利用 Python 去獲取網路上的開放數據

### 2. 整理:

運用 pandas 的 DataFrame 篩選資料

### 3. 製圖:

以 matplotlib.pyplot 將資料製成圓餅圖

# 第肆章、系統實作與展示

## 抓取資料

運用 Python 抓取網路上的檔案並印出。

(因資料來源 Kaggle 需要登入，所以先處理完放在自己的網域裡。)

資料來源:[Kaggle](#)

```
GetData.py

import requests
data_url="https://sususu.su/python/salary.json"
data=requests.get(data_url)
print(data.json())
```

發送請求並以 json 格式列印出來

```
python.json

[
  {
    "": 0,
    "work_year": 2020,
    "experience_level": "MI",
    "employment_type": "FT",
    "job_title": "Data Scientist",
    "salary": 70000,
    "salary_currency": "EUR",
    "salary_in_usd": 79833,
    "employee_residence": "DE",
    "remote_ratio": 0,
    "company_location": "DE",
    "company_size": "L"
  },
  {
    "": 1,
    "work_year": 2020,
    "experience_level": "SE",
    "employment_type": "FT",
    "job_title": "Machine Learning Scientist",
    "salary": 260000,
    "salary_currency": "USD",
    "salary_in_usd": 260000,
    "employee_residence": "JP",
    "remote_ratio": 0,
    "company_location": "JP",
    "company_size": "S"
  }
  .
  .... 600+ data
```

以上為 json 資料



# SQL 寫入

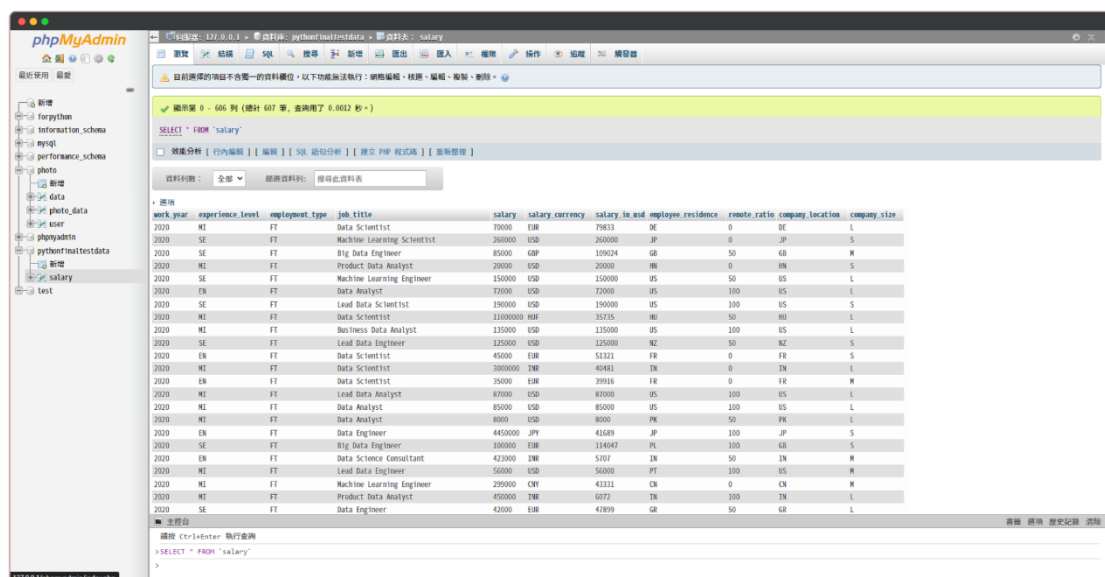
```
SQL.py

import pymysql,pandas as pd,requests
conn=pymysql.connect(host='localhost',user='root',passwd='',db='pythonfinaltestdata',port=3306,charset='utf8')

cursor=conn.cursor(pymysql.cursors.DictCursor)
data_url="https://sususu.su/python/salary.json"
data=requests.get(data_url)
data=pd.DataFrame(data.json())
data.drop(data.columns[0],axis=1,inplace=True)
for i in range(len(data)):
    cursor.execute("INSERT INTO `salary`('work_year', `experience_level`, `employment_type`, `job_title`,
`salary`, `salary_currency`, `salary_in_usd`, `employee_residence`, `remote_ratio`, `company_location`,
`company_size`) VALUES ('{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}').format(data.iloc[i]
['work_year'],data.iloc[i]['experience_level'],data.iloc[i]['employment_type'],data.iloc[i]
['job_title'],data.iloc[i]['salary'],data.iloc[i]['salary_currency'],data.iloc[i]['salary_in_usd'],data.iloc[i]
['employee_residence'],data.iloc[i]['remote_ratio'],data.iloc[i]['company_location'],data.iloc[i]
['company_size']))

cursor.execute("SELECT * FROM `salary`")
a=cursor.fetchall()
conn.commit()
```

## 寫入 SQL 程式碼

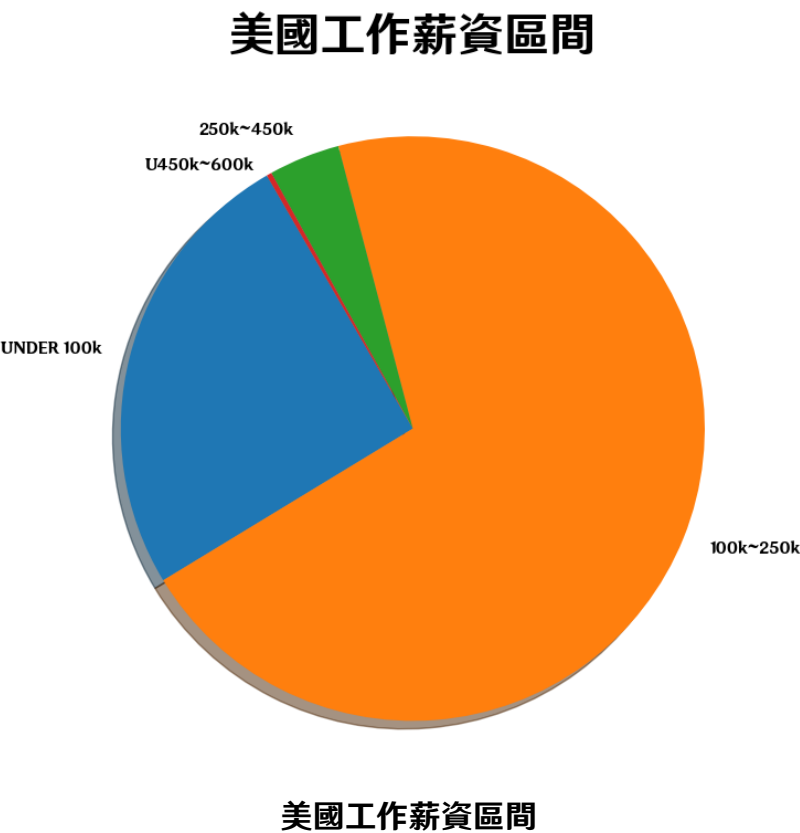
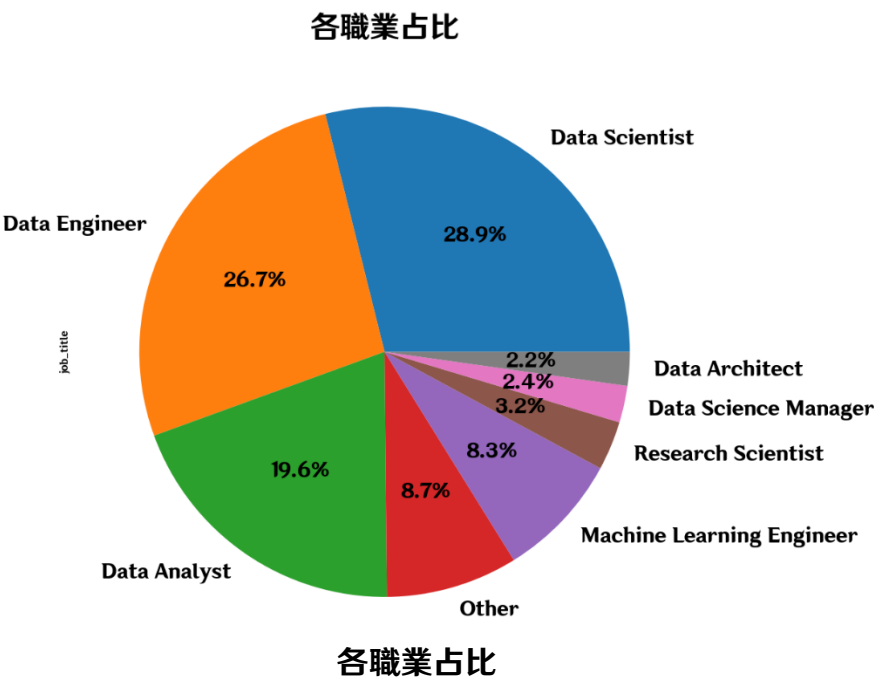


The screenshot shows the phpMyAdmin web interface. The left sidebar displays a database structure with 'pythonfinaltestdata' selected, showing a 'salary' table. The main panel shows the 'salary' table with 605 rows. The table columns are: work\_year, experience\_level, employment\_type, job\_title, salary, salary\_currency, salary\_in\_usd, employee\_residence, remote\_ratio, company\_location, and company\_size. The data is displayed in a grid format with alternating row colors.

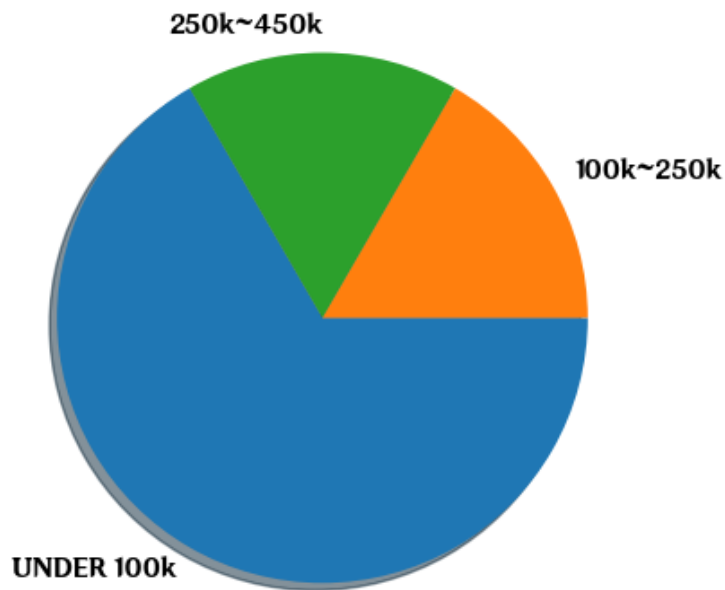
| work_year | experience_level | employment_type | job_title                  | salary   | salary_currency | salary_in_usd | employee_residence | remote_ratio | company_location | company_size |
|-----------|------------------|-----------------|----------------------------|----------|-----------------|---------------|--------------------|--------------|------------------|--------------|
| 2020      | MI               | FT              | Data Scientist             | 70000    | EUR             | 79833         | DE                 | 0            | DE               | L            |
| 2020      | SE               | FT              | Machine Learning Scientist | 260000   | USD             | 260000        | JP                 | 0            | JP               | S            |
| 2020      | SE               | FT              | Big Data Engineer          | 85000    | GBP             | 109024        | GB                 | 50           | GB               | M            |
| 2020      | MI               | FT              | Product Data Analyst       | 20000    | USD             | 20000         | IN                 | 0            | IN               | S            |
| 2020      | SE               | FT              | Machine Learning Engineer  | 150000   | USD             | 150000        | US                 | 50           | US               | L            |
| 2020      | EN               | FT              | Data Analyst               | 72000    | USD             | 72000         | US                 | 100          | US               | L            |
| 2020      | SE               | FT              | Lead Data Scientist        | 190000   | USD             | 190000        | US                 | 100          | US               | S            |
| 2020      | MI               | FT              | Data Scientist             | 11000000 | RUB             | 23735         | RU                 | 50           | RU               | L            |
| 2020      | MI               | FT              | Business Data Analyst      | 125000   | USD             | 125000        | US                 | 100          | US               | L            |
| 2020      | SE               | FT              | Lead Data Engineer         | 125000   | USD             | 125000        | NZ                 | 50           | NZ               | S            |
| 2020      | EN               | FT              | Data Scientist             | 45000    | EUR             | 51321         | FR                 | 0            | FR               | S            |
| 2020      | MI               | FT              | Data Scientist             | 3000000  | TWD             | 60481         | TW                 | 0            | TW               | L            |
| 2020      | EN               | FT              | Data Scientist             | 35000    | EUR             | 39916         | FR                 | 0            | FR               | M            |
| 2020      | MI               | FT              | Lead Data Analyst          | 80000    | USD             | 80000         | US                 | 100          | US               | L            |
| 2020      | MI               | FT              | Data Analyst               | 85000    | USD             | 85000         | US                 | 100          | US               | L            |
| 2020      | MI               | FT              | Data Analyst               | 8000     | USD             | 8000          | PK                 | 50           | PK               | L            |
| 2020      | EN               | FT              | Data Engineer              | 4450000  | JPY             | 41649         | JP                 | 100          | JP               | S            |
| 2020      | SE               | FT              | Big Data Engineer          | 100000   | EUR             | 118147        | PL                 | 100          | GB               | S            |
| 2020      | EN               | FT              | Data Science Consultant    | 420000   | TWD             | 5107          | TW                 | 50           | TW               | M            |
| 2020      | MI               | FT              | Lead Data Engineer         | 56000    | USD             | 56000         | PT                 | 100          | US               | M            |
| 2020      | MI               | FT              | Machine Learning Engineer  | 290000   | CNY             | 43331         | CN                 | 0            | CN               | M            |
| 2020      | MI               | FT              | Product Data Analyst       | 450000   | TWD             | 6072          | TW                 | 100          | TW               | L            |
| 2020      | SE               | FT              | Data Engineer              | 42000    | EUR             | 47899         | GB                 | 50           | GB               | L            |

## 存入數據庫的數據

圖表呈現

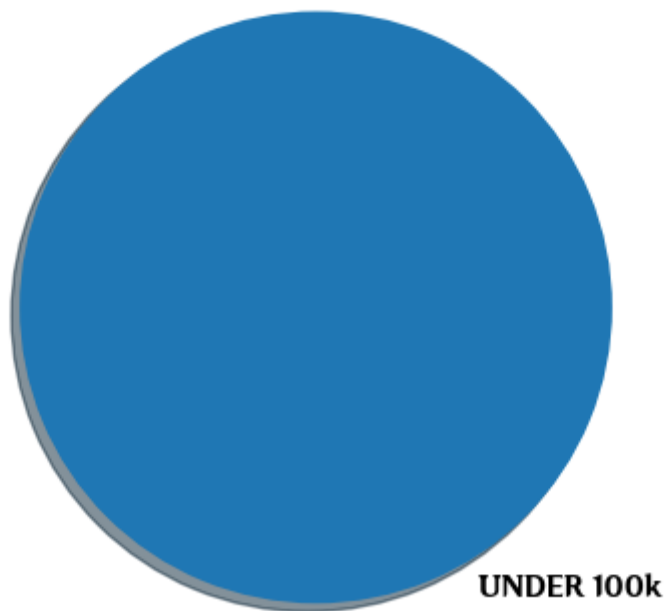


# 日本工作薪資區間



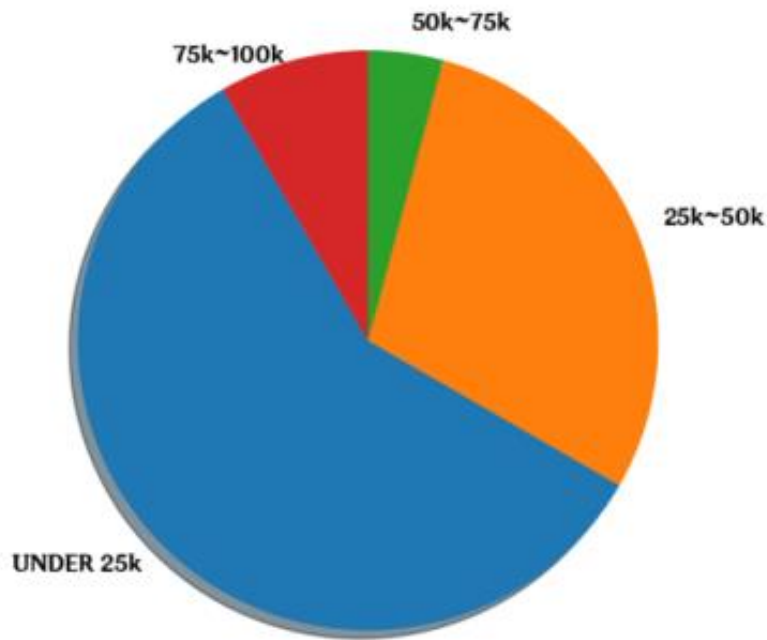
日本工作薪資區間

# 印度工作薪資區間



印度工作薪資區間

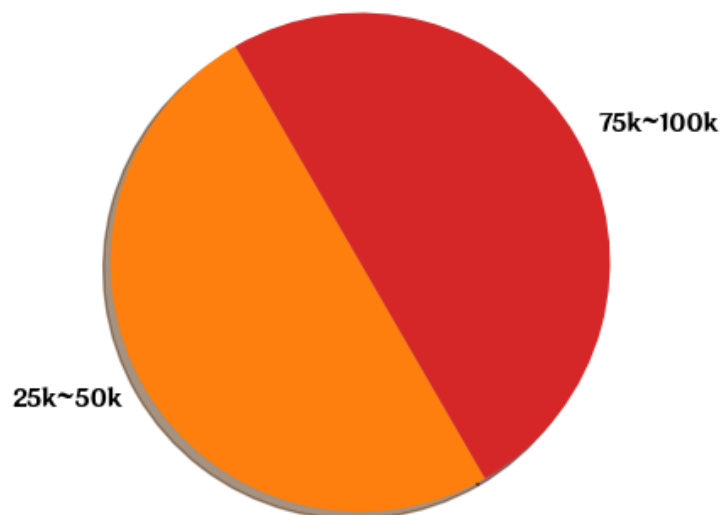
## 印度工作薪資區間



印度工作薪資區間 (因全低於 100k 所以降低篩選條件重新產生)

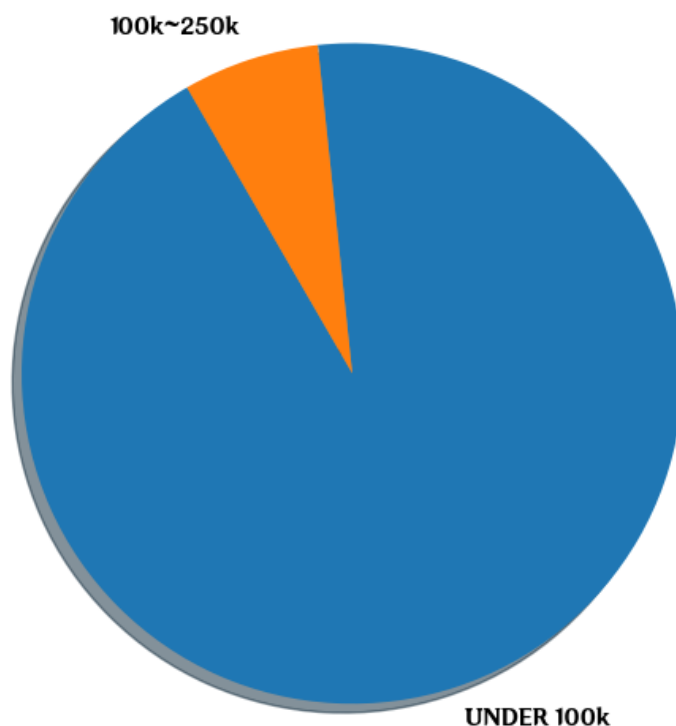
---

## 中國大陸工作薪資區間



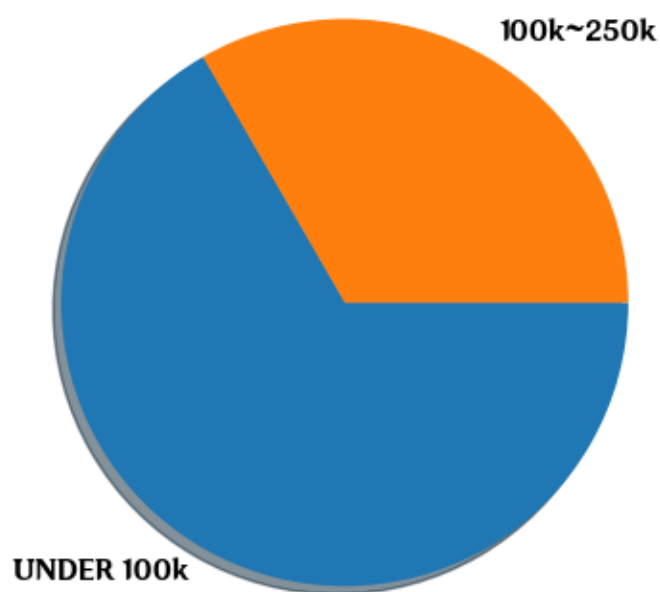
中國大陸工作薪資區間

## 法國工作薪資區間



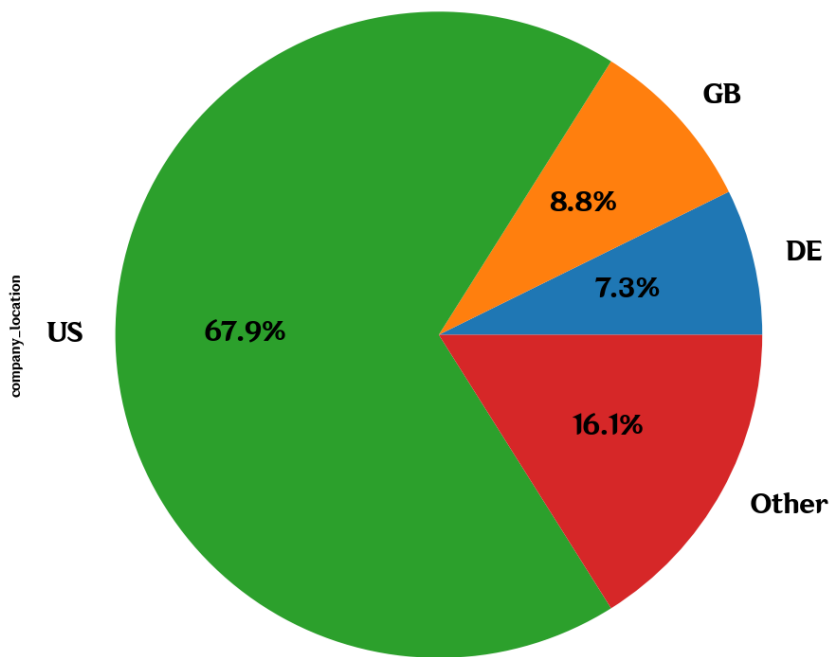
法國工作薪資區間

## 加拿大工作薪資區間



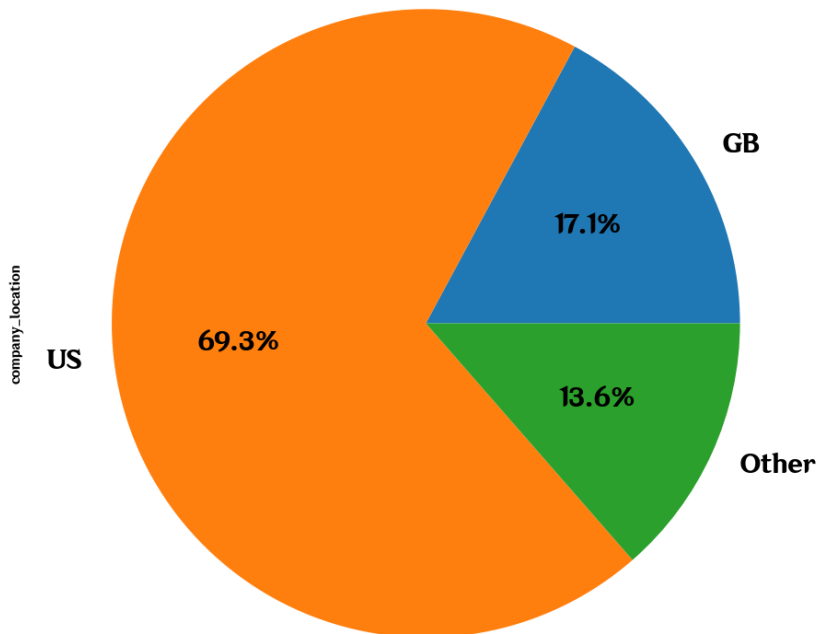
加拿大工作薪資

**Data Scientist 各國公司占比**



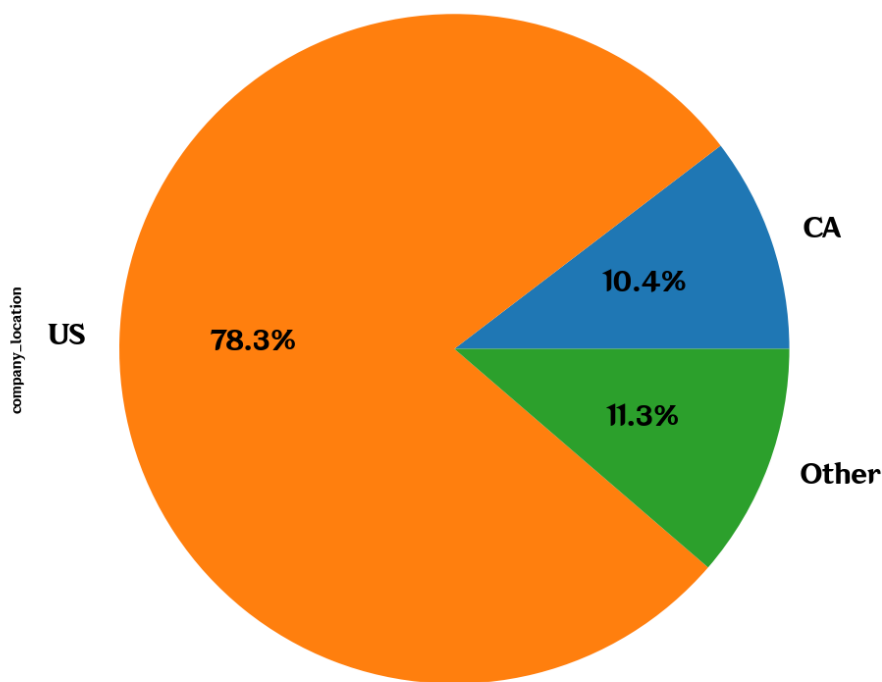
**Data Scientist 各國公司占比(GB=英國,DE=德國)**

**Data Engineer 各國公司占比**



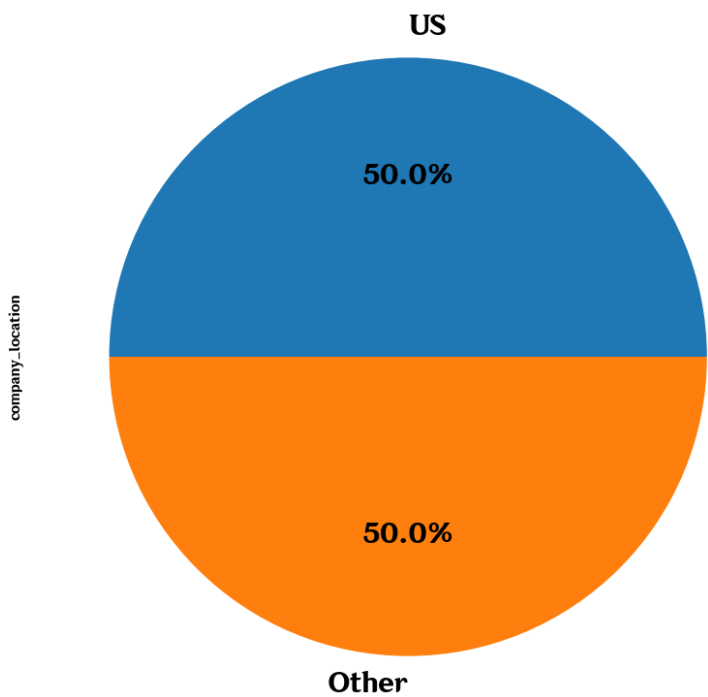
**Data Engineer 各國占比(GB=英國)**

Data Analyst 各國公司占比



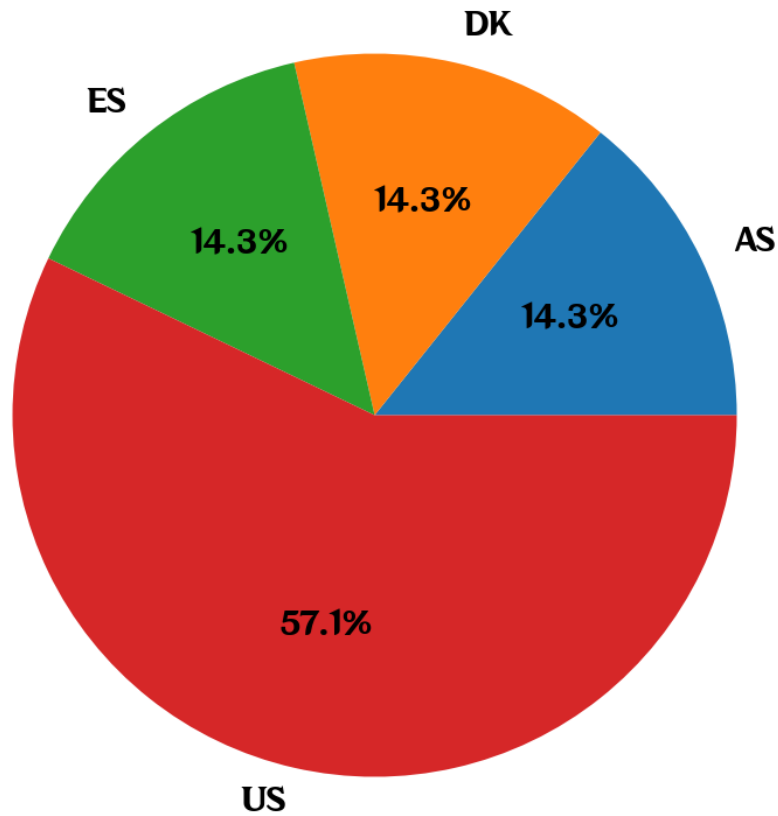
Data Analyst 各國公司占比

Machine Learning Engineer 各國公司占比



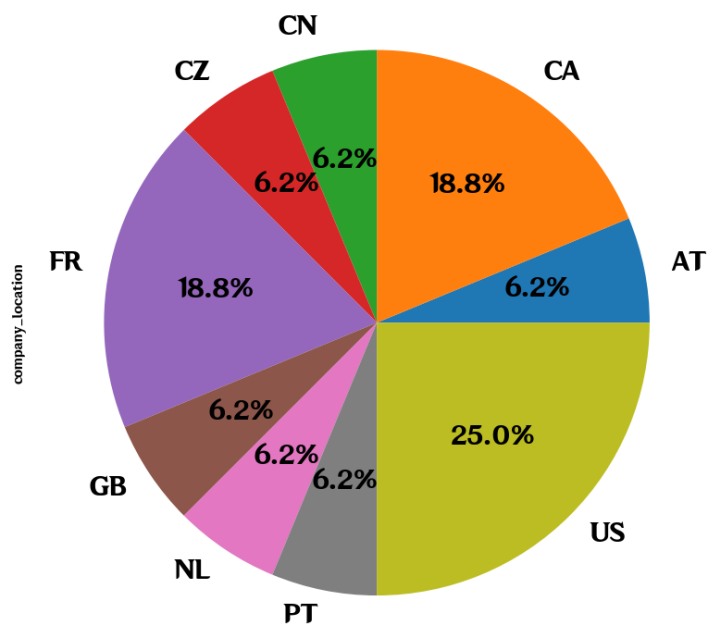
Machine Learning Engineer 各國公司占比

## AI Scientist 各國公司占比



## AI Scientist 各國公司占比

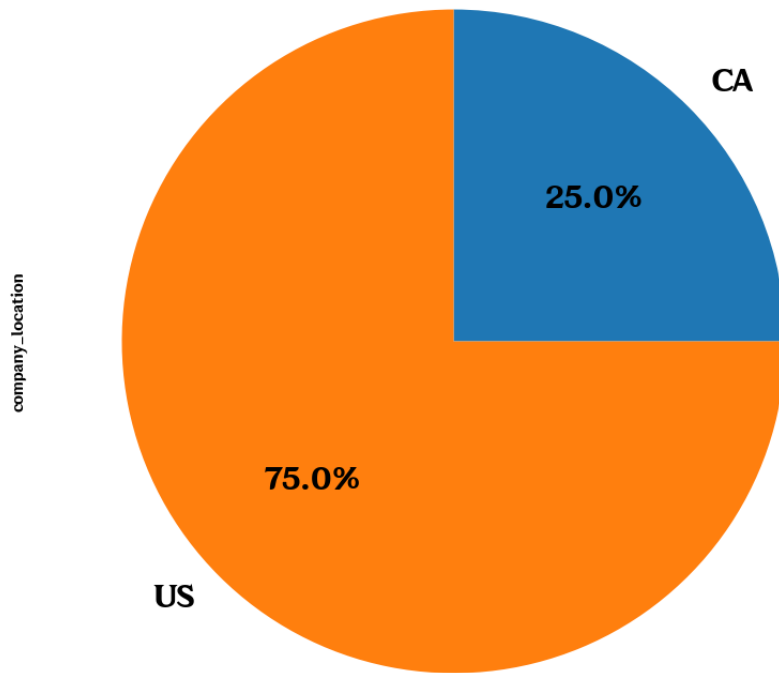
## Research Scientist 各國公司占比



## Research Scientist 各國公司比

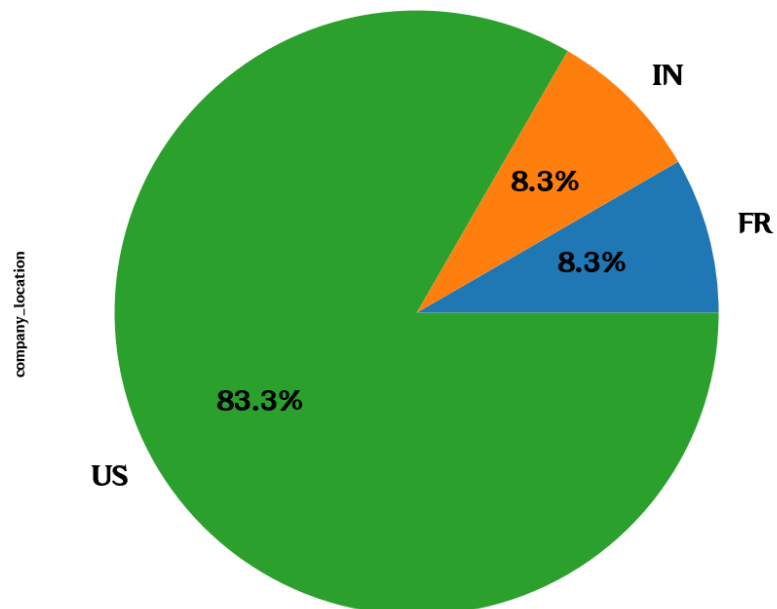


## Data Architect 各國公司占比



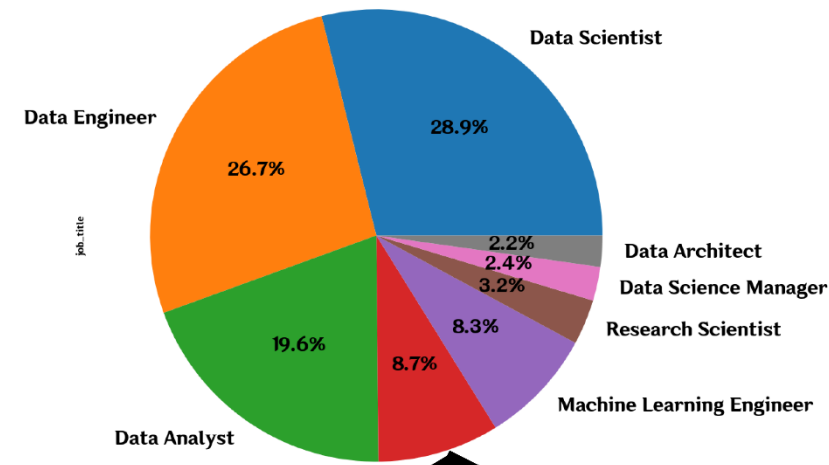
## Data Architect 各國公司占比

## Data Science Manager 各國公司占比

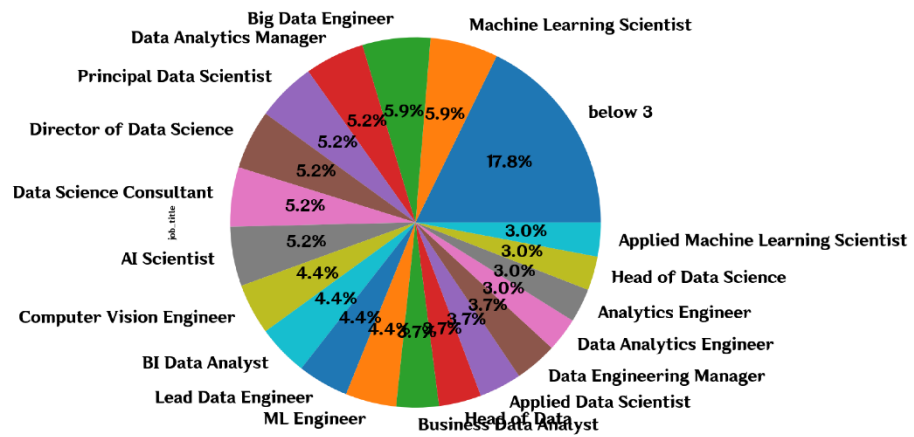


## Data Science Manager 各國公司占比

各職業占比



各職業占比



更詳細的職業占比

## 第伍章、 結論與未來展望

目前在分析完資料後，可以明確的發現，未來假如需要走向 **Data Science** 方面的話，在美國絕對有最好的潛力，薪資優越、工作機會多。希望自己在未來能夠更佳的精進自己的能力，讓自己有能力去美國工作。

# 第陸章、參考資料

## 附錄一 程式碼

```
Font Setting

from matplotlib import rcParams
#字體設定
rcParams['font.family'] = 'SDK_SC_Web'
gsfont = {'fontname': 'SDK_SC_Web'}
```

### 設置圖表字體

```
各個職業占比.py

import pandas as pd
#各個職業的占比
job_title=tmp.groupby(['job_title'])['job_title'].count()
del_list=[]
other=0
for i in job_title.index:
    if job_title[i]<10:
        del_list.append(i)
        other+=1
for i in del_list:
    job_title.drop(i,inplace=True)
#job_title add other
job_title.loc['Other']=other
job_title=job_title.sort_values(ascending=False)
chart=job_title.plot(kind='pie',figsize=(10,14),autopct='%1.1f%%',fontsize=20)
plt.title('各職業占比',fontsize=30,**gsfont)
plt.show()
plt.clf()
```

### 各個職業占比圖表生成

```

X國工作薪資區間.py

#X國工作薪資區間
Country='CA'#填入你要查詢的國家
US_SAL=tmp
US_SAL.drop(['work_year','experience_level','employment_type','salary_currency','salary','remote_ratio','employee_residence','company_size'],axis=1,inplace=True)
#get all value in company location contain "US"
US_SAL=US_SAL[US_SAL['company_location'].str.contains(Country)]
US_SAL=US_SAL.sort_values(by='salary_in_usd',ascending=True)
US_SAL.drop(US_SAL.columns[0],axis=1,inplace=True)
#UNDER 100k
US_SAL_100k=US_SAL[US_SAL['salary_in_usd']<100001].shape[0]#UNDER 250k
US_SAL_250k=US_SAL[US_SAL['salary_in_usd']<250001].shape[0]-US_SAL_100k#UNDER 450k
US_SAL_450k=US_SAL[US_SAL['salary_in_usd']<450001].shape[0]-US_SAL[US_SAL['salary_in_usd']<250001].shape[0]#UNDER 600k
US_SAL_600k=US_SAL[US_SAL['salary_in_usd']<600001].shape[0]-US_SAL[US_SAL['salary_in_usd']<450001].shape[0]

#made a pie chart
labels = ['UNDER 100k','100k~250k','250k~450k','450k~600k']
value=[US_SAL_100k,US_SAL_250k,US_SAL_450k,US_SAL_600k]
plt.pie(value,labels=labels,shadow=True,startangle=120)
plt.title('加拿大工作薪資區間',fontsize=30,**gsfont)
plt.show()
plt.clf()

```

## XX 國工作薪資區間圖表

```

不同工作各國國家佔比.py

job='Data Science Manager'
jobCount=tmp
jobCount=jobCount[jobCount['job_title'].str.contains(job)]
#count every company_location
jobCount=jobCount.groupby(['company_location'])['company_location'].count()
del_list=[]
other=0
#----丟棄10位以下的----
#for i in jobCount.index:
#    if jobCount[i]<10:
#        del_list.append(i)
#        other+=1
#for i in del_list:
#    jobCount.drop(i,inplace=True)
#job_title add other
#jobCount.loc['Other']=other
chart=jobCount.plot(kind='pie',figsize=(10,14),autopct='%1.1f%%',fontsize=20)
plt.title('{} 各國公司占比'.format(job),fontsize=30,**gsfont)
plt.show()
plt.clf()

```

```
SQL.py

import pymysql,pandas as pd,requests
conn=pymysql.connect(host='localhost',user='root',passwd='',db='pythonfinaltestdata',port=3306,charset='utf8')

cursor=conn.cursor(pymysql.cursors.DictCursor)
data_url="https://sususu.su/python/salary.json"
data=requests.get(data_url)
data=pd.DataFrame(data.json())
data.drop(data.columns[0],axis=1,inplace=True)
for i in range(len(data)):
    cursor.execute("INSERT INTO `salary`(`work_year`, `experience_level`, `employment_type`, `job_title`,
`salary`, `salary_currency`, `salary_in_usd`, `employee_residence`, `remote_ratio`, `company_location`,
`company_size`) VALUES ('{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}').format(data.iloc[i]
['work_year'],data.iloc[i]['experience_level'],data.iloc[i]['employment_type'],data.iloc[i]
['job_title'],data.iloc[i]['salary'],data.iloc[i]['salary_currency'],data.iloc[i]['salary_in_usd'],data.iloc[i]
['employee_residence'],data.iloc[i]['remote_ratio'],data.iloc[i]['company_location'],data.iloc[i]
['company_size']))

cursor.execute("SELECT * FROM `salary`")
a=cursor.fetchall()
conn.commit()
```

## SQL 寫入