API Pull and Analysis

In [5]:
```
import requests
import json
response_API = requests.get('https://www.reed.co.uk/api/1.0/search?gra
```

In [33]:
```
data = response_API.text
```

In [34]:
```
pip install reed
```

```
Requirement already satisfied: reed in c:\users\alexi\anaconda3\lib\s
ite-packages (0.0.4)
Note: you may need to restart the kernel to use updated packages.
```

In [11]:
```
from reed import ReedClient

client = ReedClient(api_key= '7269a1b6-da90-4144-9cec-c1c8b93326a0')
```

In [35]:
```
params = {
    'graduate' : "true" "false",
    'minimumSalary': 1
}
```

In [54]:
```
response = client.search(**params)
```

In [55]:
```
print(response)
```

```
[{'jobId': 48696813, 'employerId': 451880, 'employerName': 'Corri
culo Ltd', 'employerProfileId': None, 'employerProfileName': Non
e, 'jobTitle': 'Data Scientist, Forecasting Analyst, Python, R, F
inance, Remote, 4658', 'locationName': 'London', 'minimumSalary':
50000.0, 'maximumSalary': 60000.0, 'currency': 'GBP', 'expiration
Date': '12/12/2022', 'date': '31/10/2022', 'jobDescription': ' Da
ta Scientist, Forecasting Analyst, Python, R, Finance, Remot
eAn exciting opportunity has arisen for a skilled Data Scientist&
nbsp;/ Forecasting Analyst to join a thriving and rapidly gr
owing software house within the games industry. This Data Scienti
st / Forecasting Analyst position is operating on a fully re
mote basis with  opportunities to work in various locations globa
lly, with office space i... ', 'applications': 24, 'jobUrl': 'htt
ps://www.reed.co.uk/jobs/data-scientist-forecasting-analyst-pytho
n-r-finance-remote-4658/48696813'}, {'jobId': 48648132, 'employer
Id': 589273, 'employerName': 'Method Resourcing', 'employerProfil
eId': None, 'employerProfileName': None, 'jobTitle': 'Python Data
Analyst Remote', 'locationName': 'London', 'minimumSalary': 5000
0.0, 'maximumSalary': 65000.0, 'currency': 'GBP', 'expirationDat
```

In [10]:
```
df = pd.DataFrame(response)
```

In [12]:
```
df.head()
```

Out[12]:

	jobId	employerId	employerName	employerProfileId	employerProfileName	jobTit
0	48809236	331522	Harnham	None	None	Sanctio Screeni Clie Sal Lea
1	48928712	653512	LHH Recruitment Solutions	None	None	Marketi Direct
2	48809156	578236	Hansen Filler	None	None	Interim H Proces Lea
3	48928625	391063	Opus Recruitment Solutions Ltd	None	None	Seni Accou Executi
4	48809060	331522	Harnham	None	None	Sal Consulta

In [13]:
```
len(df.index)
```

Out[13]:
```
100
```

Transformation 1

The first step is to drop all of the unecessary columns from this df. In order to join to the other two datasets, I think it would be most helpful to just aggregate the API table in order to give context to # of jobs available in the market. If I leave unique rows, there is no true connection to the flat file.

In [14]:
```
df = df.drop(['jobId', 'employerId', 'employerName', 'employerProfileI
```

In [15]:
```
df.head()
```

Out[15]:

	jobTitle	locationName	minimumSalary	maximumSalary	currency	applications
0	Sanctions Screening Client Sales Lead	London	100000.0	130000.0	GBP	3
1	Marketing Director	London	110000.0	120000.0	GBP	25
2	Interim HR Process Lead	London	450.0	550.0	GBP	35
3	Senior Account Executive	London	100000.0	150000.0	GBP	3
4	Sales Consultant	London	90000.0	130000.0	GBP	7

In [16]:
```
print(df['locationName'].unique())
```

```
['London' 'Southport' 'Leeds' 'Birmingham' 'Oxford' 'Barnet' 'Cambrid
ge'
 'Blyth' 'City of London' 'Dublin' 'Manchester' 'Parrett Works' 'Ches
hire'
 'Penn' 'Bournemouth' 'Milan' 'Glasgow' 'Basingstoke' 'Wrexham' 'Camd
en'
 'United Kingdom' 'Nursling' 'Reading' 'Derby' 'Cardiff' 'Dubai' 'Cro
ydon'
 'Chelmsford' 'Fulham' 'Malvern' 'USA' 'Bristol' 'West Midlands (Regi
on)'
 'Sheffield' 'Peterborough' 'Swanley' 'South East England' 'Vienna'
 'Ascot' 'Milton Keynes']
```

In [19]:
```
df = df.drop('locationName', axis=1)
```

Transformation 2

Second transformation is to drop any results with no salary or currency information.

In [17]:
```
print(df['currency'].unique())
```

```
['GBP' None 'EUR' 'USD']
```

In [20]:
```
df = df[df.currency.notnull()]
```

In [21]:
```
df = df[df.maximumSalary.notnull()]
```

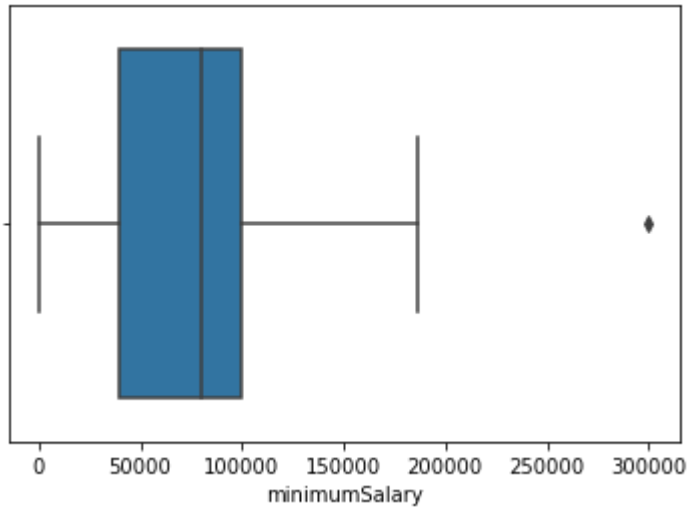
Transformation 3

Third transformation is to check for any outliers in the salary.

In [84]:
```
import numpy as np
```

In [23]:
```
import seaborn as sns
import matplotlib.pyplot as plt

sns.boxplot(x = df['minimumSalary'])
plt.show()
```



In [24]:
```
df['minimumSalary'].describe()
```

Out[24]:
```
count        75.000000
mean      72961.846667
std       52761.942584
min          38.500000
25%       40000.000000
50%       80000.000000
75%      100000.000000
max      300000.000000
Name: minimumSalary, dtype: float64
```

In [27]:
```
df['minimumSalary'].quantile(0.25)
```

Out[27]:
```
40000.0
```

In [28]:
```
df['minimumSalary'].quantile(0.75)
```

Out[28]:
```
100000.0
```

In [44]:
```
Q1 = df['minimumSalary'].quantile(0.25)
Q3 = df['minimumSalary'].quantile(0.75)
IQR = Q3 - Q1
```

In [45]:
```
IQR
```

Out[45]:
```
60000.0
```

In [32]:
```
lower_lim = Q1 - 1.5 * IQR
upper_lim = Q3 + 1.5 * IQR
```

In [33]:
```
lower_lim
```

Out[33]:
```
-50000.0
```

In [34]:
```
upper_lim
```

Out[34]:
```
190000.0
```

In [49]:
```
outliers_15_low = (df['minimumSalary'] < lower_lim)
```

In [50]:
```
outliers_15_up = (df['minimumSalary'] > upper_lim)
```

In [51]:
```
len(df['minimumSalary']) - len(df['minimumSalary'][outliers_15_low] +
```

Out[51]:
```
75
```

In [52]:
```
df['minimumSalary'][(outliers_15_low | outliers_15_up)]
```

Out[52]:
```
23    300000.0
Name: minimumSalary, dtype: float64
```

In [53]:
```
df['minimumSalary'][-(outliers_15_low | outliers_15_up)]
```

Out[53]:
```
0     100000.0
1     110000.0
2        450.0
3     100000.0
4      90000.0
         ...
91     80000.0
92    100000.0
93     80000.0
94     51000.0
98       300.0
Name: minimumSalary, Length: 74, dtype: float64
```

In [54]:
```
df = df[-(outliers_15_low | outliers_15_up)]
```

In [55]:
```
df
```

Out[55]:

	jobTitle	minimumSalary	maximumSalary	currency	applications
0	Sanctions Screening Client Sales Lead	100000.0	130000.0	GBP	3
1	Marketing Director	110000.0	120000.0	GBP	25
2	Interim HR Process Lead	450.0	550.0	GBP	35
3	Senior Account Executive	100000.0	150000.0	GBP	3
4	Sales Consultant	90000.0	130000.0	GBP	7
...	...	...	...	...	...
91	Sales & Marketing Director	80000.0	100000.0	GBP	21
92	Head of Consumer Insights & Strategic Marketing	100000.0	100000.0	GBP	17
93	Chief Marketing Officer	80000.0	110000.0	GBP	2
94	Regional Sales Director - Ascot	51000.0	120000.0	GBP	0
98	HR Systems Manager	300.0	450.0	GBP	0

74 rows × 5 columns

Transformation 4

Validate that there are no duplicate jobs postings pulled from the API.

In [58]:
```
boolean = not df["jobTitle"].is_unique
boolean
```

Out[58]: True

In [60]:
```
boolean = df['jobTitle'].duplicated().any()
boolean
```

Out[60]: True

Transformation 5

The final step is to categorize the jobs by department and aggregate the data so that we can provide insights to the existing employee data.

In [18]:

```
print(df['jobTitle'].unique())
```

['Sanctions Screening Client Sales Lead' 'Marketing Director'
 'Interim HR Process Lead ' 'Senior Account Executive' 'Sales Consult
ant'
 'National Sales Manager / Sales Director - LegalTech/PropertyTech'
 'Marketing Campaign Manager - SC Cleared - Contract' 'Sales Executiv
e '
 'Specification Sales Manager - Wall Partition Systems'
 'Sales Closer - Financial Services' 'Interim Senior HR Business Part
ner'
 'Cloud Pre-sales Architect' 'Marketing & Brand Consultant'
 'Sales Apprentice' 'Digital Marketing Coordinator'
 'Head of Sales Operations' 'Senior Medical Writer pharmacovigilance'
 'UK Sales Manager Intralogistics' 'Cloud Security Sales Specialist'
 'Head of Sales - Middle East' 'B2B Sales Manager'
 'Sales / Business Development - DLT / Blockchain, ESG, Sustainable F
inance - EMEA - Remote'
 'Client Biller Marketing Agency' 'HR Officer'
 'Solution Architect - HR Systems' 'Sales Manager' 'Institutional Sal
es'
 'Sales Director'
 'Enterprise Sales\xa0- Cloud, Digital Transformation, UC & Security
Solutions'
 'Medical Scientist' 'Sales and Marketing Director - Work From Home'
 'Salesforce Marketing Cloud Specialist'
 'General Manager - Sales & Distribution' 'Head of HR'
 'Digital Marketing Director ' 'Chief Marketing Officer'
 'Insurance Sales Executive' 'Manager of Territory Accounts '
 'HR Change Assistant Manager' 'Digital Growth Marketing Manager cont
ract'
 'Interim Director of Marketing Exco-1 - Listed Business'
 'Graduate Sales Executive' 'HR Process Mapping Lead Workday'
 'Consultant Psychiatrist' 'Head of Product Marketing - Loyalty'
 'HR Advisor' 'Future Talent Team HR Advisor'
 'Director of Customer & Marketing'
 'Head of Digital Marketing (Hybrid Working)'
 'Head of Marketing (Hybrid Working)'
 'Sales & Business Development Lead - AI/Data/Remote' 'Nurse Practiti
oner'
 'Sales Advisor' 'Sales and Leasing Agent' 'Head of Marketing'
 'Sales and Business Partner - Recruitment Resourcer Remote'
 'European Sales Manager ' 'Senior HR Manager, Operations and Benefit
s'
 'Interim Director, People and Organisational Development'
 'Sales - Broker Investments £150K OTE'
 'Sales Closer Investment Broker - V High Income'
 'Chief Marketing Officer ' 'Sales Broker Executive Investments'
 'Luxury Car Dealership Sales Executive' 'Head of Sales Contact Centr
e'
 'Head of Business Development / Institutional Sales'
 'Head of Operational Human Resources'
 'New Business Sales - Digital Transformation, Cloud, Uc & Security S
olutions'
 'Software Solutions Sales Executives / Account Managers'
 'Head of Residential Sales'
 'Digital Marketing and Communications Manager'
 'Sales Ledger Manager 12 month FTC'
 'Solution Architect - HR Transformation Programme, ERP'

```
 'Head Of People, HR Director' 'Marketing & Engagement Manager - Phar
ma'
 'HR Technology Framework Lead'
 'Medical Writer - clinical and regulatory documents'
 'Sales & Marketing Director'
 'Head of Consumer Insights & Strategic Marketing'
 'Regional Sales Director - Ascot' 'Associate Product Marketing Manag
er'
 'Regional Sales Lead' 'Head of Brand & Marketing Creative (Remote)'
 'HR Systems Manager' 'HR Transformation Assistant Manager']
```

In [64]:
```
## Create input dataframe using copy to clipboard

department = ['Sales', 'Sales', 'Human Resources', 'Sales', 'Sales', '
```

In [65]:
```
df['Department'] = department
```

In [66]:
```
df
```

Out[66]:

	jobTitle	minimumSalary	maximumSalary	currency	applications	Department
0	Sanctions Screening Client Sales Lead	100000.0	130000.0	GBP	3	Sales
1	Marketing Director	110000.0	120000.0	GBP	25	Sales
2	Interim HR Process Lead	450.0	550.0	GBP	35	Human Resources
3	Senior Account Executive	100000.0	150000.0	GBP	3	Sales
4	Sales Consultant	90000.0	130000.0	GBP	7	Sales
...	...	...	...	...	...	...
91	Sales & Marketing Director	80000.0	100000.0	GBP	21	Sales
92	Head of Consumer Insights & Strategic Marketing	100000.0	100000.0	GBP	17	Sales
93	Chief Marketing Officer	80000.0	110000.0	GBP	2	Sales
94	Regional Sales Director - Ascot	51000.0	120000.0	GBP	0	Sales
98	HR Systems Manager	300.0	450.0	GBP	0	Human Resources

74 rows × 6 columns

In [81]:
```
df['Department'].value_counts()
```

Out[81]:
```
Sales                    55
Human Resources          10
Healthcare                5
Acounting and Finance     3
Other                     1
Name: Department, dtype: int64
```

In [85]:
```
df.pivot_table(index='Department', values='applications', aggfunc=np.s
```

Out[85]:

Department	applications
Acounting and Finance	18
Healthcare	5
Human Resources	185
Other	4
Sales	1794

In [86]:
```
df.pivot_table(index='Department', values='minimumSalary', aggfunc=np.
```

Out[86]:

Department	minimumSalary
Acounting and Finance	34333.333333
Healthcare	53907.700000
Human Resources	38260.000000
Other	100000.000000
Sales	78490.909091

In [87]:
```
df.pivot_table(index='Department', values='maximumSalary', aggfunc=np.
```

Out[87]:

Department	maximumSalary
Acounting and Finance	145000.000000
Healthcare	62125.400000
Human Resources	47307.700000
Other	120000.000000
Sales	118537.454545

In [88]:
```
Data = {'Department': ['Sales', 'Human Resources', 'Healthcare', 'Acco
        'Job_Listings': [55, 10, 5, 3, 1],
        'Applications': [1794, 185, 5, 15, 4],
        'Average_minimumSalary': [78490, 38260, 53907, 34333, 100000],
        'Average_maximumSalary': [118537, 47307, 62125, 145000, 120000

df_final = pd.DataFrame(Data)

df_final
```

Out[88]:

	Department	Job_Listings	Applications	Average_minimumSalary	Average_maximumSala
0	Sales	55	1794	78490	1185
1	Human Resources	10	185	38260	473
2	Healthcare	5	5	53907	621
3	Accounting and Finance	3	15	34333	1450
4	Other	1	4	100000	1200

Implications and Considerations

This final table has a significantly smaller sample size than the original flat file. Sales had the most job listings and applications submitted, which makes it the most reliable department out of the sample. However, for real-world application, a longer processing time and pull from the API would be required to get the most comprehensive averages and look at the current job market available. Ideally, this would also be pulled from an American job postings board (or wherever the source data originates). Since the data in the project is fabricated, no additional social considerations can be made of the sample population. In a real-world scenario, engagement with key HR business partners, Legal, and Privacy would be required in order to facilitate conversations regarding employee data and talent acquisition and retention.

In [89]:
```
df_final.to_excel("milestone_4.xlsx")
```