DATA ANALYSIS AND VISUALIZATION
Project Report

On

Data Analysis Job Market in the United States

SUBMITTED BY :

Shivam Gupta (19519)
Kushagra Kaushik (19540)
Vaishnavi Khare (19545)

SUBMITTED TO :

Mrs. Anamika Gupta

DECLARATION

We hereby declare that this project report, submitted to Shaheed Sukhdev College of Business Studies, University of Delhi is a record of an official work done by us under the guidance of Mrs. Anamika Gupta (Course Instructor). This project is submitted in the partial fulfillment of the requirements for the completion of the "Data Analysis and Visualization" course. The results embodied have not been submitted to any other University or Institute for the award of any degree or diploma.

Mrs. Anamika Gupta

Shivam Gupta (19519)
Kushagra Kaushik (19540)
Vaishnavi Khare (19545)

ACKNOWLEDGMENT

The project report was a great chance for learning and professional development. Therefore, we consider ourselves very lucky as we were provided with an opportunity to be a part of it. The success and outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of the project.

Bearing in mind previous we are using this opportunity to express our deepest gratitude and special thanks to Mrs. Anamika Gupta who in spite of being extraordinarily busy with her duties, took time out to hear, guide and keep us on the correct path and allowing us to carry out this project.

TABLE OF CONTENTS

Data Analysis Job Market in the United States

Data grows faster and faster every second, it seems to be a good idea to employ the crew which will handle this data efficiently (especially when your company processes a lot of data every day).

Here come, data scientists, data analysts, data engineers and many many more specialists who know how to deal with data. But this project will show data analyst jobs based on salaries, companies and sectors where this job is the most common.

A). Data Description

This data is about the different intricate details about jobs being offered by different companies in the United States. This data can be used to speed up or filter the job search process for data analysis enthusiasts. The data offers various different criteria to look for in a job. The columns with their description in the data are:

1. Job Title - The actual job profile being offered.
2. Salary Estimate - The salary range suggested by glassdoor.co.in for the profile being offered.
3. Job Description - This column describes the whole concept of the job alongwith a brief explanation of the company's ideology.
4. Rating - Rating out of 5 given to the corresponding companies.
5. Company Name - Name of the company offering the job.
6. Location - Location where the job is being offered.
7. Headquarters - Location of the headquarters of the company offering the job.
8. Size - Size of the firm based on the number of employees.
9. Founded - The year the company was founded in.
10. Type of Ownership - Type of business ownership of the firm.
11. Industry - What industry is the company associated with.
12. Sector - Which sector does the company fall under.
13. Revenue - Annual Revenue of the company.
14. Competitors - Most visible or obvious competitors of the company.
15. Easy Apply - Is Easy Apply available for the job on sites like Linkedin.

The link to the data being used in the project:

https://www.kaggle.com/andrewmvd/data-analyst-jobs (https://www.kaggle.com/andrewmvd/data-analyst-jobs)

B). Importing Libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Importing modules is an important part of working with Python that allows you to call functions that are not part of your main program.

We have imported four libraries which are :

1.Numpy provides support for large multidimensional array objects and various tools to work with them. 2.Pandas is a package of python which allows us to perform various operations on datasets such as reshaping,merging,sorting ,slicing and so on. 3..Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. 4.Seaborn is an open-source Python library built on top of matplotlib. It is used for data visualization and exploratory data analysis. Seaborn works easily with dataframes and the Pandas library

The Actual Data

In [2]:
```
data = pd.read_csv('DataAnalyst.csv')
data
```

Out[2]:

	Unnamed: 0	Job Title	Salary Estimate	Job Description	Rating	Company Name
0	0	Data Analyst, Center on Immigration and Justic...	$37K-66K (Glassdoor est.)	Are you eager to roll up your sleeves and harn...	3.2	Vera Institute of Justice\n3.2
1	1	Quality Data Analyst	$37K-66K (Glassdoor est.)	Overview\n\nProvides analytical and technical ...	3.8	Visiting Nurse Service of New York\n3.8
2	2	Senior Data Analyst, Insights & Analytics Team...	$37K-66K (Glassdoor est.)	We're looking for a Senior Data Analyst who ha...	3.4	Squarespace\n3.4
3	3	Data Analyst	$37K-66K (Glassdoor est.)	Requisition NumberRR-0001939\nRemote:Yes\nWe c...	4.1	Celerity\n4.1
4	4	Reporting Data Analyst	$37K-66K (Glassdoor est.)	ABOUT FANDUEL GROUP\n\nFanDuel Group is a worl...	3.9	FanDuel\n3.9
...	...	...	...	...	...	...
2248	2248	RQS - IHHA - 201900004460 -1q Data Security An...	$78K-104K (Glassdoor est.)	Maintains systems to protect data from unautho...	2.5	Avacend, Inc.\n2.5
2249	2249	Senior Data Analyst (Corporate Audit)	$78K-104K (Glassdoor est.)	Position:\nSenior Data Analyst (Corporate Audi...	2.9	Arrow Electronics\n2.9
2250	2250	Technical Business Analyst (SQL, Data analytic...	$78K-104K (Glassdoor est.)	Title: Technical Business Analyst (SQL, Data a...	-1.0	Spiceorb
2251	2251	Data Analyst 3, Customer Experience	$78K-104K (Glassdoor est.)	Summary\n\nResponsible for working cross-funct...	3.1	Contingent Network Services\n3.1
2252	2252	Senior Quality Data Analyst	$78K-104K (Glassdoor est.)	You.\n\nYou bring your body, mind, heart and s...	3.4	SCL Health\n3.4

2253 rows × 16 columns

In [3]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2253 entries, 0 to 2252
Data columns (total 16 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Unnamed: 0         2253 non-null   int64  
 1   Job Title          2253 non-null   object 
 2   Salary Estimate    2253 non-null   object 
 3   Job Description    2253 non-null   object 
 4   Rating             2253 non-null   float64
 5   Company Name       2252 non-null   object 
 6   Location           2253 non-null   object 
 7   Headquarters       2253 non-null   object 
 8   Size               2253 non-null   object 
 9   Founded            2253 non-null   int64  
 10  Type of ownership  2253 non-null   object 
 11  Industry           2253 non-null   object 
 12  Sector             2253 non-null   object 
 13  Revenue            2253 non-null   object 
 14  Competitors        2253 non-null   object 
 15  Easy Apply         2253 non-null   object 
dtypes: float64(1), int64(2), object(13)
memory usage: 281.8+ KB
```

C). Cleaning of Data

In [4]:
```
#checking for null values
```

In [5]:
```
data.isnull().sum()
```

Out[5]:
```
Unnamed: 0           0
Job Title            0
Salary Estimate      0
Job Description      0
Rating               0
Company Name         1
Location             0
Headquarters         0
Size                 0
Founded              0
Type of ownership    0
Industry             0
Sector               0
Revenue              0
Competitors          0
Easy Apply           0
dtype: int64
```

The missing data in this dataset is marked as '-1'. So, to turn this into a more data analysis friendly format, we will convert all -1s to NaN (Not a Number).

In [6]:
```
data.replace('-1', np.nan, inplace=True)
data.replace(-1, np.nan, inplace=True)
data.replace(-1.0, np.nan, inplace=True)
```

In [7]:
```
data.isnull().sum()
```

Out[7]:
```
Unnamed: 0             0
Job Title              0
Salary Estimate        1
Job Description        0
Rating               272
Company Name           1
Location               0
Headquarters         172
Size                 163
Founded              660
Type of ownership    163
Industry             353
Sector               353
Revenue              163
Competitors         1732
Easy Apply          2173
dtype: int64
```

In [8]:
```
data['Job Title'].value_counts()
```

Out[8]:
```
Data Analyst                                              405
Senior Data Analyst                                        90
Junior Data Analyst                                        30
Business Data Analyst                                      28
Sr. Data Analyst                                           21
                                                         ...
Data Management Analyst - Dallas, TX/ Tempe, FL             1
Senior BI and Data Analyst                                  1
Data Analyst/SQL/Healthcare Or Medical OR Clinical          1
Senior Data Analyst, Tax Technology (Level 6)               1
Data Governance Business Analyst                            1
Name: Job Title, Length: 1272, dtype: int64
```

In [9]:
```
same_job_name = {'Sr. Data Analyst' : 'Senior Data Analyst',
                 'Sr Data Analyst' : 'Senior Data Analyst',
                 'Data Analyst Junior' : 'Junior Data Analyst',
                'Data Analyst Senior' : 'Senior Data Analyst'}

data['Job Title'] = data['Job Title'].map(same_job_name).fillna(data['Job Titl
e'])
```

In [10]:
```
data['Job Title'].value_counts()
```

Out[10]:
```
Data Analyst                                              405
Senior Data Analyst                                       121
Junior Data Analyst                                        47
Business Data Analyst                                      28
Data Quality Analyst                                       17
                                                         ...
Data Management Analyst - Dallas, TX/ Tempe, FL             1
Senior BI and Data Analyst                                  1
Data Analyst/SQL/Healthcare Or Medical OR Clinical          1
Senior Data Analyst, Tax Technology (Level 6)               1
Data Governance Business Analyst                            1
Name: Job Title, Length: 1268, dtype: int64
```

Converting salary data into numeric form

In [11]:
```
data['min_salary'] = data['Salary Estimate'].str.replace('$', '').str[:3].str.
replace('K', '').str.strip().astype('float')
data['max_salary'] = data['Salary Estimate'].str[6:10].str.replace('K','').str
.lstrip('$').str.strip().astype('float')
```

```
<ipython-input-11-a085343c0aa3>:1: FutureWarning: The default value of regex
will change from True to False in a future version. In addition, single chara
cter regular expressions will*not* be treated as literal strings when regex=T
rue.
  data['min_salary'] = data['Salary Estimate'].str.replace('$', '').str[:3].s
tr.replace('K', '').str.strip().astype('float')
```

In [12]:
```
data['average_salary'] = (data['min_salary'] + data['max_salary']) / 2
# Thus, the resulting columns will be:
data[['average_salary', 'min_salary', 'max_salary']].head(n = 10)
```

Out[12]:

	average_salary	min_salary	max_salary
0	51.5	37.0	66.0
1	51.5	37.0	66.0
2	51.5	37.0	66.0
3	51.5	37.0	66.0
4	51.5	37.0	66.0
5	51.5	37.0	66.0
6	51.5	37.0	66.0
7	51.5	37.0	66.0
8	51.5	37.0	66.0
9	51.5	37.0	66.0

In [ ]:
```
#We divided the salary estimate column in in minimum and maximum salary ie in
 two different columns and creating a new column
# that calculates the average of salary along with reomving the extra string v
alues.
```

In [13]:
```
data['Size'].unique()
```

Out[13]:
```
array(['201 to 500 employees', '10000+ employees',
       '1001 to 5000 employees', '501 to 1000 employees',
       '5001 to 10000 employees', '1 to 50 employees',
       '51 to 200 employees', 'Unknown', nan], dtype=object)
```

In [14]:
```
#Changing the values of size column to solid numeeric figures and removing str
ings for better numeric analysis:
ind = 0
sizes = []
for i in data['Size']:
    if 'to' in str(i):
        sizes.append(((int(i.split(' to ')[0])) + int(i.split(' to ')[1][:-10
])) / 2)
    if '+' in str(i):
        sizes.append(int(i[:-11]))
    if pd.isnull(i):
        sizes.append(i)
    if str(i) == 'Unknown':
        sizes.append(np.nan)
```

In [15]:
```
data['Size'] = sizes
```

In [16]:
```
data['Size'].unique()
```

Out[16]:
```
array([  350.5, 10000. ,  3000.5,   750.5,  7500.5,    25.5,   125.5,
          nan])
```

In [17]:
```
data.head()
```

Out[17]:

	Unnamed: 0	Job Title	Salary Estimate	Job Description	Rating	Company Name	Locatic
0	0	Data Analyst, Center on Immigration and Justic...	$37K-66K (Glassdoor est.)	Are you eager to roll up your sleeves and harn...	3.2	Vera Institute of Justice\n3.2	Ne York, N
1	1	Quality Data Analyst	$37K-66K (Glassdoor est.)	Overview\n\nProvides analytical and technical ...	3.8	Visiting Nurse Service of New York\n3.8	Ne York, N
2	2	Senior Data Analyst, Insights & Analytics Team...	$37K-66K (Glassdoor est.)	We're looking for a Senior Data Analyst who ha...	3.4	Squarespace\n3.4	Ne York, N
3	3	Data Analyst	$37K-66K (Glassdoor est.)	Requisition NumberRR-0001939\nRemote:Yes\nWe c...	4.1	Celerity\n4.1	Ne York, N
4	4	Reporting Data Analyst	$37K-66K (Glassdoor est.)	ABOUT FANDUEL GROUP\n\nFanDuel Group is a worl...	3.9	FanDuel\n3.9	Ne York, N

Top-10 jobs being offered by frequency.

In [18]:
```
pd.value_counts(data['Job Title']).head(n = 10)
```

Out[18]:
```
Data Analyst                405
Senior Data Analyst         121
Junior Data Analyst          47
Business Data Analyst        28
Data Quality Analyst         17
Data Analyst II              17
Data Governance Analyst      16
Lead Data Analyst            15
Data Reporting Analyst       13
Financial Data Analyst       12
Name: Job Title, dtype: int64
```

405 companies offer Data Analyst position i.e. the position of data analyst in highest in demand.

Dropping columns that are of not much use

In [19]:
```
data.drop(['Unnamed: 0','Salary Estimate','Competitors','Easy Apply'], axis =
1, inplace = True)
```

Renaming columns to a more readable and informative format.

In [20]:
```
data.rename(columns = {'min_salary' : 'Min Salary (k USD)', 'max_salary' : 'Ma
x Salary (k USD)',
                       'average_salary' : 'Avg Salary (k USD)', 'Size' : 'Avg
 No of employees'}, inplace = True)
```

In [21]:
```
data = data.reindex(columns = ['Job Title','Job Description', 'Min Salary (k U
SD)','Max Salary (k USD)',
                               'Avg Salary (k USD)', 'Rating',  'Avg No of emp
loyees','Company Name', 'Location',
                               'Headquarters', 'Industry', 'Sector', 'Type of
 ownership', 'Founded'])
#Rearranging the columns for better data represntation
```

In [22]:
```
data.head()
```

Out[22]:

	Job Title	Job Description	Min Salary (k USD)	Max Salary (k USD)	Avg Salary (k USD)	Rating	Avg No of employees	Company N
0	Data Analyst, Center on Immigration and Justic...	Are you eager to roll up your sleeves and harn...	37.0	66.0	51.5	3.2	350.5	Vera Institu Justice
1	Quality Data Analyst	Overview\n\nProvides analytical and technical ...	37.0	66.0	51.5	3.8	10000.0	Visiting N Service of York
2	Senior Data Analyst, Insights & Analytics Team...	We're looking for a Senior Data Analyst who ha...	37.0	66.0	51.5	3.4	3000.5	Squarespace
3	Data Analyst	Requisition NumberRR-0001939\nRemote:Yes\nWe c...	37.0	66.0	51.5	4.1	350.5	Celerity
4	Reporting Data Analyst	ABOUT FANDUEL GROUP\n\nFanDuel Group is a worl...	37.0	66.0	51.5	3.9	750.5	FanDuel

Removing the rating part from the Company Name as we have a separate column for that already.

In [23]:
```
data['Company Name'] = data['Company Name'].str.split('\n').str[0]
```

In [24]:
```
data.isnull().sum()
```

Out[24]:
```
Job Title               0
Job Description         0
Min Salary (k USD)      1
Max Salary (k USD)      1
Avg Salary (k USD)      1
Rating                272
Avg No of employees   205
Company Name            1
Location                0
Headquarters          172
Industry              353
Sector                353
Type of ownership     163
Founded               660
dtype: int64
```

We have 14 columns in total now. So, we will remove the rows having more than 9 columns i.e. more than half with null values.

In [25]:
```
clean_data = data.dropna(thresh = 8)
clean_data.shape
```

Out[25]:
```
(2090, 14)
```

Picking out specific skills from job description which could be of our use

In [26]:
```
job_desc = pd.DataFrame()
skills = ['python', 'sql', 'power bi', 'excel']
for option in skills:
    job_desc[option] = clean_data['Job Description'].apply(lambda text: option
in text.lower())
```

In [27]:
```
job_desc
```

Out[27]:

	python	sql	power bi	excel
0	True	True	False	False
1	False	True	False	True
2	True	True	False	True
3	False	True	False	False
4	True	True	False	True
...	...	...	...	...
2247	False	True	False	True
2248	False	False	False	False
2249	False	True	False	False
2251	True	True	False	False
2252	False	False	False	True

2090 rows × 4 columns

In [28]:
```
clean_data.shape
```

Out[28]:
```
(2090, 14)
```

In [29]:
```
#Droppping the job description column because it is not much of our use since
 we have already found the important parts from it
clean_data.drop(['Job Description'], axis=1,inplace =True)
```

```
C:\Users\kusha\anaconda3\lib\site-packages\pandas\core\frame.py:4308: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy
  return super().drop(
```

In [30]:
```
#Cleaned data :
clean_data
```

Out[30]:

	Job Title	Min Salary (k USD)	Max Salary (k USD)	Avg Salary (k USD)	Rating	Avg No of employees	Company Name	Locati
0	Data Analyst, Center on Immigration and Justic...	37.0	66.0	51.5	3.2	350.5	Vera Institute of Justice	New Yo I
1	Quality Data Analyst	37.0	66.0	51.5	3.8	10000.0	Visiting Nurse Service of New York	New Yo I
2	Senior Data Analyst, Insights & Analytics Team...	37.0	66.0	51.5	3.4	3000.5	Squarespace	New Yo I
3	Data Analyst	37.0	66.0	51.5	4.1	350.5	Celerity	New Yo I
4	Reporting Data Analyst	37.0	66.0	51.5	3.9	750.5	FanDuel	New Yo I
...	...	...	...	...	...	...	...	
2247	Marketing/Communications - Data Analyst-Marketing	78.0	104.0	91.0	4.1	125.5	APN Software Services Inc.	Broomfie C
2248	RQS - IHHA - 201900004460 -1q Data Security An...	78.0	104.0	91.0	2.5	125.5	Avacend, Inc.	Denv C
2249	Senior Data Analyst (Corporate Audit)	78.0	104.0	91.0	2.9	10000.0	Arrow Electronics	Centenn C
2251	Data Analyst 3, Customer Experience	78.0	104.0	91.0	3.1	350.5	Contingent Network Services	Centenn C
2252	Senior Quality Data Analyst	78.0	104.0	91.0	3.4	10000.0	SCL Health	Broomfie C

2090 rows × 13 columns

In [31]:
```
clean_data.head()
```

Out[31]:

	Job Title	Min Salary (k USD)	Max Salary (k USD)	Avg Salary (k USD)	Rating	Avg No of employees	Company Name	Location	Headquarters
0	Data Analyst, Center on Immigration and Justic...	37.0	66.0	51.5	3.2	350.5	Vera Institute of Justice	New York, NY	New York, NY
1	Quality Data Analyst	37.0	66.0	51.5	3.8	10000.0	Visiting Nurse Service of New York	New York, NY	New York, NY
2	Senior Data Analyst, Insights & Analytics Team...	37.0	66.0	51.5	3.4	3000.5	Squarespace	New York, NY	New York, NY
3	Data Analyst	37.0	66.0	51.5	4.1	350.5	Celerity	New York, NY	McLean, VA
4	Reporting Data Analyst	37.0	66.0	51.5	3.9	750.5	FanDuel	New York, NY	New York, NY

In [32]:
```
clean_data.shape
```

Out[32]: (2090, 13)

In [33]:
```
clean_data.describe()
```

Out[33]:

	Min Salary (k USD)	Max Salary (k USD)	Avg Salary (k USD)	Rating	Avg No of employees	Founded
count	2089.000000	2089.000000	2089.000000	1981.000000	2048.000000	1593.000000
mean	54.167066	89.868837	72.017951	3.731903	2846.209229	1978.362837
std	19.542672	29.404971	23.640123	0.670332	3806.228417	47.822289
min	24.000000	38.000000	33.500000	1.000000	25.500000	1682.000000
25%	41.000000	70.000000	58.000000	3.300000	125.500000	1970.000000
50%	50.000000	87.000000	68.500000	3.700000	750.500000	1997.000000
75%	63.000000	104.000000	80.500000	4.100000	3000.500000	2006.000000
max	113.000000	190.000000	150.000000	5.000000	10000.000000	2019.000000

In [34]:
```
clean_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2090 entries, 0 to 2252
Data columns (total 13 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Job Title            2090 non-null   object 
 1   Min Salary (k USD)   2089 non-null   float64
 2   Max Salary (k USD)   2089 non-null   float64
 3   Avg Salary (k USD)   2089 non-null   float64
 4   Rating               1981 non-null   float64
 5   Avg No of employees  2048 non-null   float64
 6   Company Name         2090 non-null   object 
 7   Location             2090 non-null   object 
 8   Headquarters         2081 non-null   object 
 9   Industry             1900 non-null   object 
 10  Sector               1900 non-null   object 
 11  Type of ownership    2090 non-null   object 
 12  Founded              1593 non-null   float64
dtypes: float64(6), object(7)
memory usage: 228.6+ KB
```

Leftover null values in columns.

In [35]:
```
clean_data.isnull().sum()
```

Out[35]:
```
Job Title               0
Min Salary (k USD)      1
Max Salary (k USD)      1
Avg Salary (k USD)      1
Rating                109
Avg No of employees    42
Company Name            0
Location                0
Headquarters            9
Industry              190
Sector                190
Type of ownership       0
Founded               497
dtype: int64
```

Finding the correlation between all the numeric columns.

In [36]:
```
clean_data.corr()
```

Out[36]:

	Min Salary (k USD)	Max Salary (k USD)	Avg Salary (k USD)	Rating	Avg No of employees	Founded
Min Salary (k USD)	1.000000	0.860394	0.948441	0.025664	-0.036020	0.082436
Max Salary (k USD)	0.860394	1.000000	0.977562	0.051387	-0.021723	0.099208
Avg Salary (k USD)	0.948441	0.977562	1.000000	0.042560	-0.028382	0.095733
Rating	0.025664	0.051387	0.042560	1.000000	-0.131874	0.176102
Avg No of employees	-0.036020	-0.021723	-0.028382	-0.131874	1.000000	-0.580917
Founded	0.082436	0.099208	0.095733	0.176102	-0.580917	1.000000

In [93]:
```
# Not much correlation other than the negative correlation between 'Founded' a
nd 'No of Employees' which is understandable
# as it was founded later on. Also, there is negative correlation between sala
ry and no of employees. Not much but it is there.
```

In [37]:
```
clean_data
```

Out[37]:

	Job Title	Min Salary (k USD)	Max Salary (k USD)	Avg Salary (k USD)	Rating	Avg No of employees	Company Name	Locati
0	Data Analyst, Center on Immigration and Justic...	37.0	66.0	51.5	3.2	350.5	Vera Institute of Justice	New Yo
1	Quality Data Analyst	37.0	66.0	51.5	3.8	10000.0	Visiting Nurse Service of New York	New Yo
2	Senior Data Analyst, Insights & Analytics Team...	37.0	66.0	51.5	3.4	3000.5	Squarespace	New Yo
3	Data Analyst	37.0	66.0	51.5	4.1	350.5	Celerity	New Yo
4	Reporting Data Analyst	37.0	66.0	51.5	3.9	750.5	FanDuel	New Yo
...	...	...	...	...	...	...	...	
2247	Marketing/Communications - Data Analyst-Marketing	78.0	104.0	91.0	4.1	125.5	APN Software Services Inc.	Broomfie
2248	RQS - IHHA - 201900004460 -1q Data Security An...	78.0	104.0	91.0	2.5	125.5	Avacend, Inc.	Denv
2249	Senior Data Analyst (Corporate Audit)	78.0	104.0	91.0	2.9	10000.0	Arrow Electronics	Centenn
2251	Data Analyst 3, Customer Experience	78.0	104.0	91.0	3.1	350.5	Contingent Network Services	Centenn
2252	Senior Quality Data Analyst	78.0	104.0	91.0	3.4	10000.0	SCL Health	Broomfie

2090 rows × 13 columns

1). For a starting example subset, we will find out the average salary being offered by companies that were founded in the 21st century.

In [38]:
```
founded_after_2000 = clean_data[clean_data['Founded']>=2000]
founded_after_2000['Avg Salary (k USD)'].mean()
```

Out[38]: 74.44233687405159

2).Finding all the top rated companies

In [39]:
```
top_rated_companies = pd.DataFrame(clean_data[clean_data['Rating']==5]['Company Name'].unique())
top_rated_companies.columns = ["Company Name"]
top_rated_companies
```

Out[39]:

	Company Name
0	Duolingo
1	Whip Media Group
2	Mulberry Technology
3	EDO, Inc.
4	Crossfire Consulting Corporation
...	...
56	Concept Software & Services Inc
57	Strive Health
58	Edison Software, Inc.
59	xentity corporation
60	Goldstone Partners

61 rows × 1 columns

In [ ]:
```
#Duolingo is highest rated company
```

In [40]:
```
clean_data.loc[:,['Job Title','Avg Salary (k USD)','Rating']]
```

Out[40]:

	Job Title	Avg Salary (k USD)	Rating
0	Data Analyst, Center on Immigration and Justic...	51.5	3.2
1	Quality Data Analyst	51.5	3.8
2	Senior Data Analyst, Insights & Analytics Team...	51.5	3.4
3	Data Analyst	51.5	4.1
4	Reporting Data Analyst	51.5	3.9
...	...	...	...
2247	Marketing/Communications - Data Analyst-Marketing	91.0	4.1
2248	RQS - IHHA - 201900004460 -1q Data Security An...	91.0	2.5
2249	Senior Data Analyst (Corporate Audit)	91.0	2.9
2251	Data Analyst 3, Customer Experience	91.0	3.1
2252	Senior Quality Data Analyst	91.0	3.4

2090 rows × 3 columns

3).Finding the number of jobs with respect to the location

In [41]:
```
clean_data['Location'].value_counts()
```

Out[41]:
```
New York, NY             292
Chicago, IL              119
San Francisco, CA        112
Austin, TX                77
Los Angeles, CA           73
                        ...
Arlington Heights, IL      1
Northridge, CA             1
Exton, PA                  1
Conshohocken, PA           1
Elk Grove Village, IL      1
Name: Location, Length: 242, dtype: int64
```

In [94]:
```
# There is a big difference between the first 2 companies in the list. Howeve
r, after that there is a slow but continuous
# decline in the no. of jobs.
```

4).Finding the average of salary offered in each sector in descending order

In [42]:
```
clean_data.groupby('Sector', as_index=False)['Avg Salary (k USD)'].mean().sort_values(by="Avg Salary (k USD)",ascending=False)
```

Out[42]:

	Sector	Avg Salary (k USD)
3	Biotech & Pharmaceuticals	83.106061
18	Real Estate	80.916667
2	Arts, Entertainment & Recreation	80.642857
0	Accounting & Legal	75.220930
11	Information Technology	74.247368
10	Health Care	72.807947
4	Business Services	72.273423
13	Manufacturing	72.075000
6	Consumer Services	71.500000
5	Construction, Repair & Maintenance	70.968750
21	Telecommunications	70.363636
12	Insurance	70.254902
7	Education	69.567308
22	Transportation & Logistics	69.350000
1	Aerospace & Defense	68.500000
14	Media	68.476190
8	Finance	67.644970
20	Retail	66.684211
16	Non-Profit	65.769231
9	Government	64.319444
17	Oil, Gas, Energy & Utilities	63.269231
23	Travel & Tourism	62.000000
15	Mining & Metals	59.500000
19	Restaurants, Bars & Food Services	54.000000

In [1]:
```
#Biotech & Pharmaceuticals,Real Estate and Arts - Entertainment & Recreation sectors
#provides the highest average salary
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

D). Discretization and Binning

In [43]:
```
#on founded column
clean_data.Founded.unique()
```

Out[43]:
```
array([1961., 1893., 2003., 2002., 2009., 2014., 2001., 1914., 1896.,
       2013., 1841.,   nan, 1884., 2004., 2016., 1996., 1958., 2006.,
       1998., 1969., 2011., 2017., 2005., 1904., 2018., 1956., 2007.,
       1990., 1923., 2015., 1951., 1981., 2000., 1966., 1939., 1999.,
       1812., 1945., 1985., 1984., 1962., 1995., 2012., 1967., 1799.,
       1918., 1948., 1994., 1895., 1845., 1968., 2008., 1911., 1997.,
       1832., 1934., 1938., 1963., 1992., 1977., 1886., 1862., 1906.,
       1993., 1935., 1971., 1989., 1859., 1946., 1818., 1932., 1784.,
       1856., 1978., 1912., 1957., 1937., 1982., 1964., 1850., 1983.,
       1869., 1947., 1868., 1936., 1853., 2010., 1902., 1944., 1888.,
       1880., 1973., 1931., 1987., 2019., 1919., 1979., 1954., 1980.,
       1976., 1872., 1926., 1991., 1855., 1959., 1986., 1890., 1882.,
       1883., 1837., 1817., 1908., 1988., 1965., 1828., 1889., 1898.,
       1928., 1913., 1901., 1916., 1857., 1955., 1949., 1970., 1822.,
       1972., 1864., 1975., 1960., 1885., 1922., 1740., 1830., 1887.,
       1974., 1905., 1953., 1952., 1897., 1865., 1921., 1867., 1933.,
       1874., 1891., 1807., 1870., 1892., 1852., 1698., 1924., 1907.,
       1682., 1875., 1917., 1876., 1785., 1909., 1789., 1930.])
```

In [44]:
```
#max value in Founded
clean_data['Founded'].max()
```

Out[44]:
```
2019.0
```

In [45]:
```
#min value in Founded
clean_data['Founded'].min()
```

Out[45]:
```
1682.0
```

In [46]:
```
#binning on Founded column
df_founded = clean_data.loc[clean_data['Founded'].notnull()]
df_founded['Founded']
```

Out[46]:
```
0       1961.0
1       1893.0
2       2003.0
3       2002.0
4       2009.0
         ...
2243    1930.0
2245    2004.0
2249    1935.0
2251    1984.0
2252    1864.0
Name: Founded, Length: 1593, dtype: float64
```

In [47]:
```
#counting the number of companies for each unique year of founding
df_founded.Founded.value_counts().sort_index()
```

Out[47]:
```
1682.0     1
1698.0     1
1740.0     5
1784.0     9
1785.0     1
          ..
2015.0    28
2016.0    22
2017.0    18
2018.0    19
2019.0     4
Name: Founded, Length: 160, dtype: int64
```

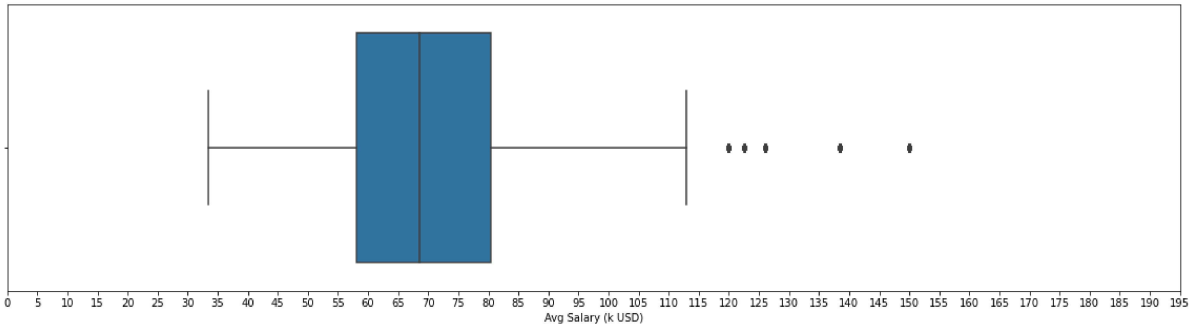
In [48]:
```
founded = df_founded['Founded']
bins = [1650, 1700, 1750, 1800, 1850, 1900, 1950, 2000, 2020]
cats = pd.cut(founded,bins)
print(pd.value_counts(cats).sort_index())
```

```
(1650, 1700]      2
(1700, 1750]      5
(1750, 1800]     21
(1800, 1850]     24
(1850, 1900]     72
(1900, 1950]    148
(1950, 2000]    715
(2000, 2020]    606
Name: Founded, dtype: int64
```

E).Detecting outliers

In [49]:
```
plt.figure(figsize = (20, 5))
sns.boxplot(clean_data['Avg Salary (k USD)'])
plt.xticks(range(0, 200, 5))
plt.show()
```

```
C:\Users\kusha\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



Since the average salaray is very high after $115K so taking it as the limit and seeing the percentage of data having higher average salary than that

In [50]:
```
percentage = len(clean_data[clean_data['Avg Salary (k USD)'] > 115.0 ])/len(cl
ean_data['Avg Salary (k USD)']) * 100
print("The percentage of the companies giving salary higher than $115k is {:.2
f}%".format(percentage))
```

```
The percentage of the companies giving salary higher than $115k is 6.22%
```

Right now we're keeping these values since they are not few

Finding the number of jobs available in each sector

In [51]:
```
jobs_in_sector = pd.DataFrame(clean_data['Sector'].value_counts())
jobs_in_sector.reset_index(inplace = True)
jobs_in_sector
```

Out[51]:

	index	Sector
0	Information Technology	570
1	Business Services	524
2	Finance	169
3	Health Care	151
4	Education	52
5	Insurance	51
6	Accounting & Legal	43
7	Media	42
8	Manufacturing	40
9	Retail	38
10	Government	36
11	Biotech & Pharmaceuticals	33
12	Non-Profit	26
13	Aerospace & Defense	22
14	Transportation & Logistics	20
15	Construction, Repair & Maintenance	16
16	Consumer Services	14
17	Oil, Gas, Energy & Utilities	13
18	Real Estate	12
19	Telecommunications	11
20	Restaurants, Bars & Food Services	8
21	Arts, Entertainment & Recreation	7
22	Travel & Tourism	1
23	Mining & Metals	1

In [95]:
```
# After IT and Business Services, the no. of jobs drastically fell.
```

F). Data Visualisation

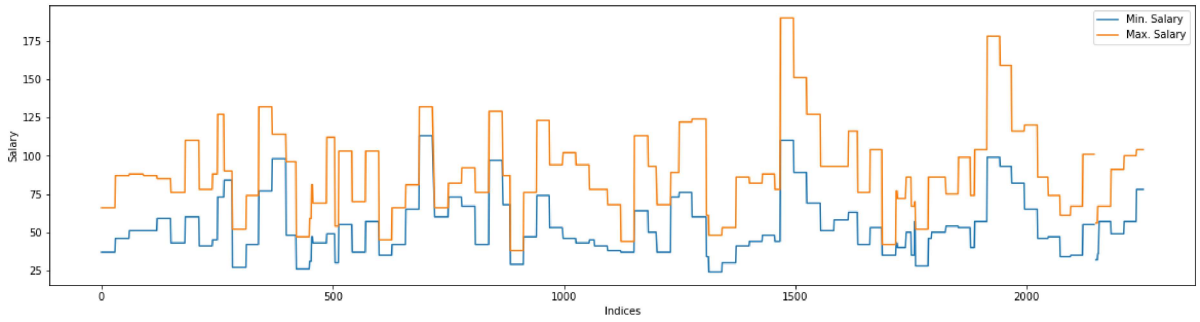
In [52]:
```
#making a copy of the dataframe

df_original = clean_data.copy()
```

1. Plotting the minimum and maximum salary provided by the companies

In [53]:
```
plt.figure(figsize = (20, 5))
plt.plot(clean_data['Min Salary (k USD)'],  label = 'Min. Salary')
plt.plot(clean_data['Max Salary (k USD)'],  label = 'Max. Salary')
plt.xlabel('Indices')
plt.ylabel('Salary')
plt.legend(loc = 'best')
```

Out[53]: <matplotlib.legend.Legend at 0x1ccfd6c5fd0>



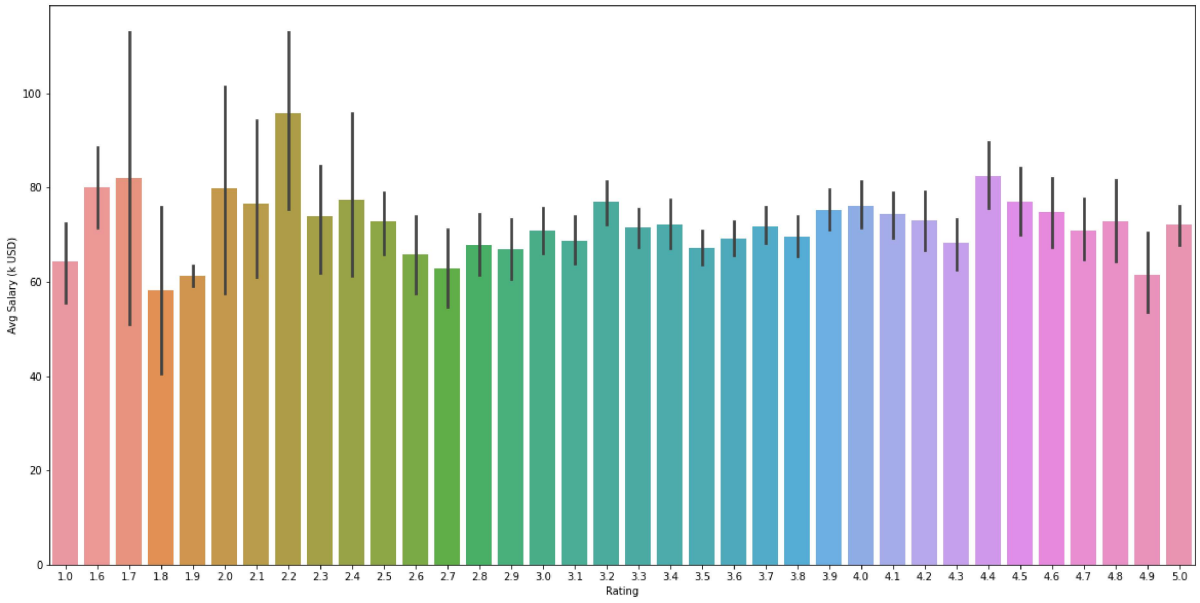
We can notice that there are many points in the above graph where min and max salary overlap. Hence there is no "fixed min or max salary" in analysis business.

2. Finding the average salaries provided by different rated companies

In [54]:
```
# Barplot for the rating and average salaries.
plt.figure(figsize = (20, 10))
sns.barplot(clean_data['Rating'], clean_data['Avg Salary (k USD)'])
```

```
C:\Users\kusha\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variables as keyword args: x, y. From version 0.1
2, the only valid positional argument will be `data`, and passing other argum
ents without an explicit keyword will result in an error or misinterpretatio
n.
  warnings.warn(
```

Out[54]: <AxesSubplot:xlabel='Rating', ylabel='Avg Salary (k USD)'>



The minimum average salary provided by diiferent rated companies is above than 50K(USD)

3. Finding the average salary across each Sector

In [55]:
```
#average salary per sector
avg_salary_sector = clean_data.groupby('Sector')['Avg Salary (k USD)'].mean()
avg_salary_sector = pd.DataFrame(avg_salary_sector)
avg_salary_sector.reset_index(inplace = True)
avg_salary_sector.sort_values(by = "Avg Salary (k USD)", ascending = False)
```

Out[55]:

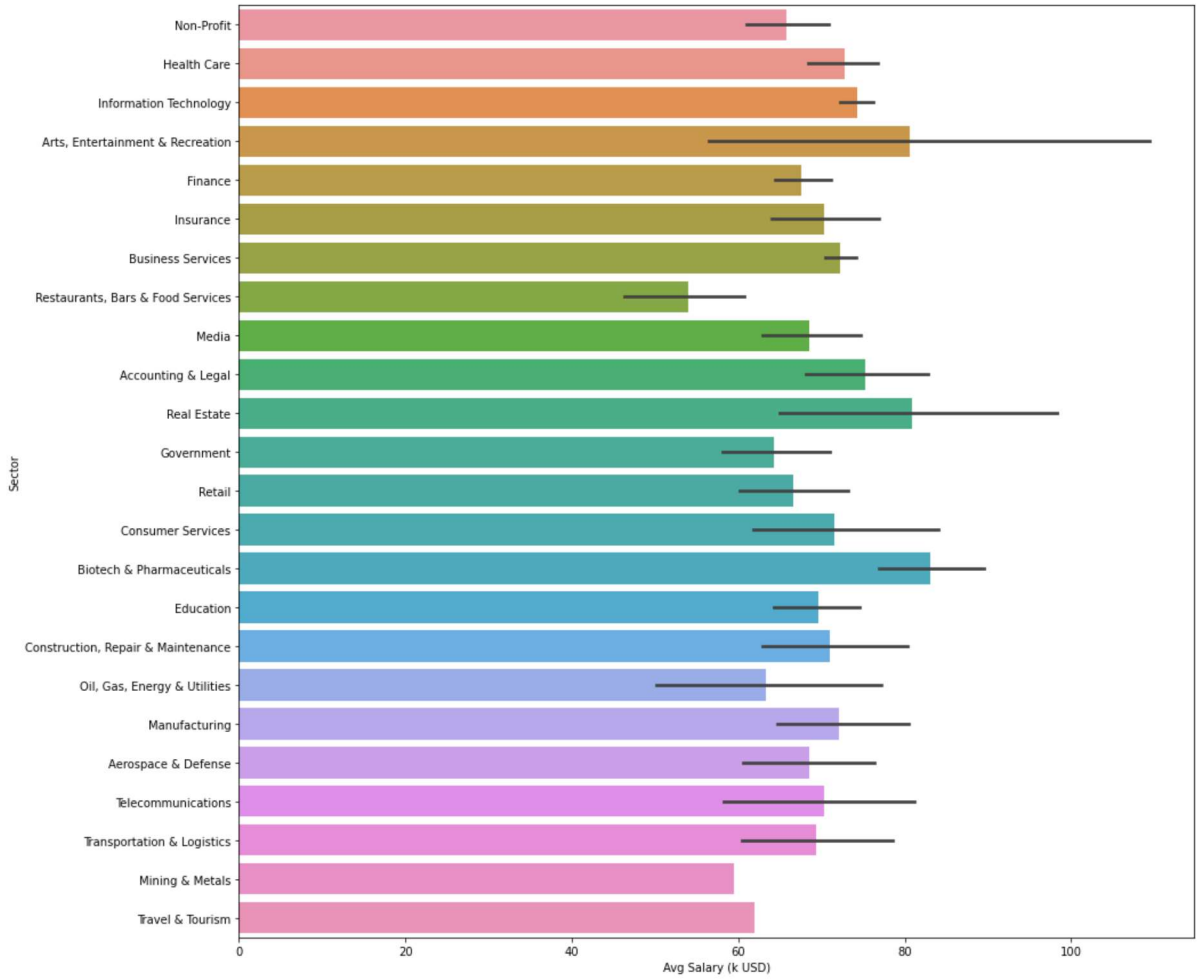
	Sector	Avg Salary (k USD)
3	Biotech & Pharmaceuticals	83.106061
18	Real Estate	80.916667
2	Arts, Entertainment & Recreation	80.642857
0	Accounting & Legal	75.220930
11	Information Technology	74.247368
10	Health Care	72.807947
4	Business Services	72.273423
13	Manufacturing	72.075000
6	Consumer Services	71.500000
5	Construction, Repair & Maintenance	70.968750
21	Telecommunications	70.363636
12	Insurance	70.254902
7	Education	69.567308
22	Transportation & Logistics	69.350000
1	Aerospace & Defense	68.500000
14	Media	68.476190
8	Finance	67.644970
20	Retail	66.684211
16	Non-Profit	65.769231
9	Government	64.319444
17	Oil, Gas, Energy & Utilities	63.269231
23	Travel & Tourism	62.000000
15	Mining & Metals	59.500000
19	Restaurants, Bars & Food Services	54.000000

The Biotechnical and Pharmaceuticals Sector provides the highest average salary, followed by Real Estate

In [56]:
```
#plotting the average_salaries across each sector
plt.figure(figsize = (15, 15))
sns.barplot(clean_data['Avg Salary (k USD)'],clean_data['Sector'])
```

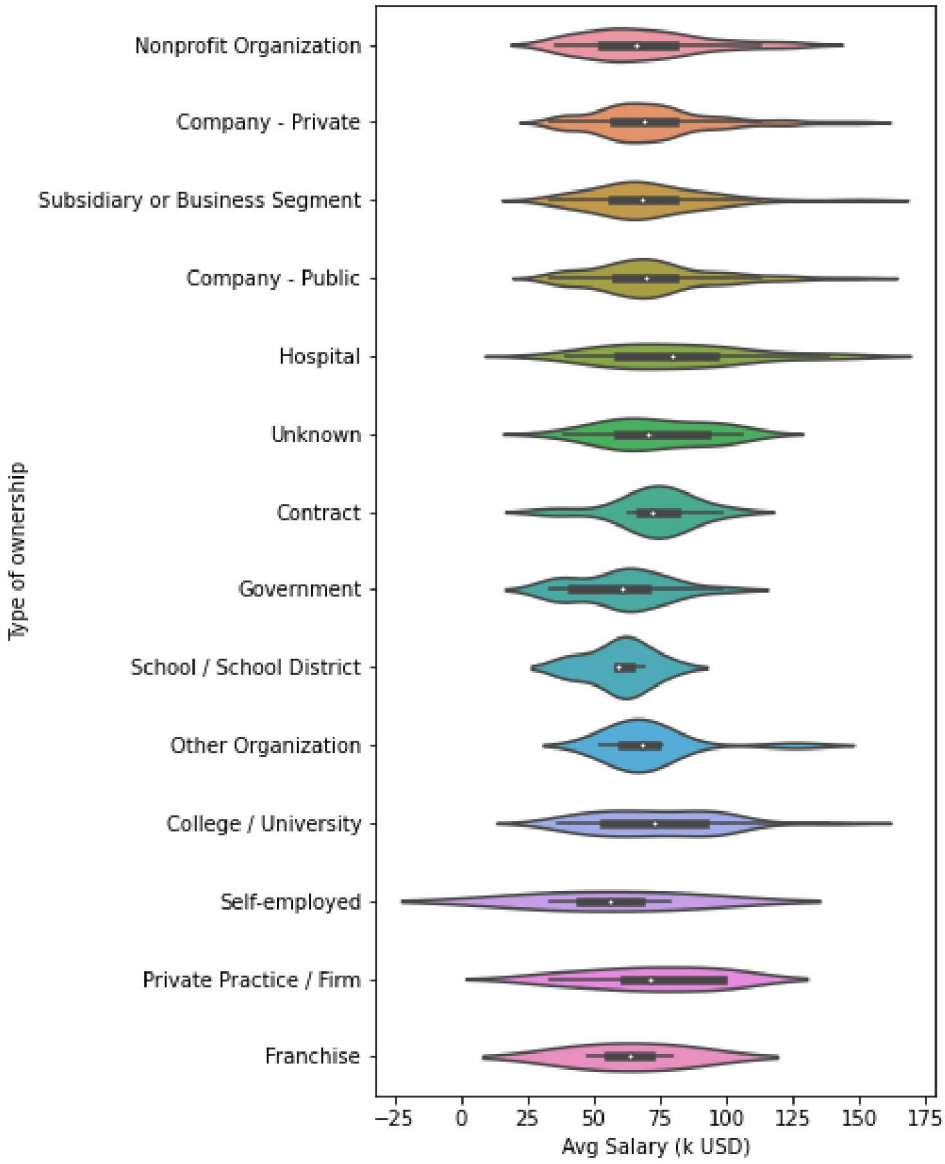
```
C:\Users\kusha\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variables as keyword args: x, y. From version 0.1
2, the only valid positional argument will be `data`, and passing other argum
ents without an explicit keyword will result in an error or misinterpretatio
n.
  warnings.warn(
```

Out[56]: <AxesSubplot:xlabel='Avg Salary (k USD)', ylabel='Sector'>



4. Visualising the average salary across different types of ownerships

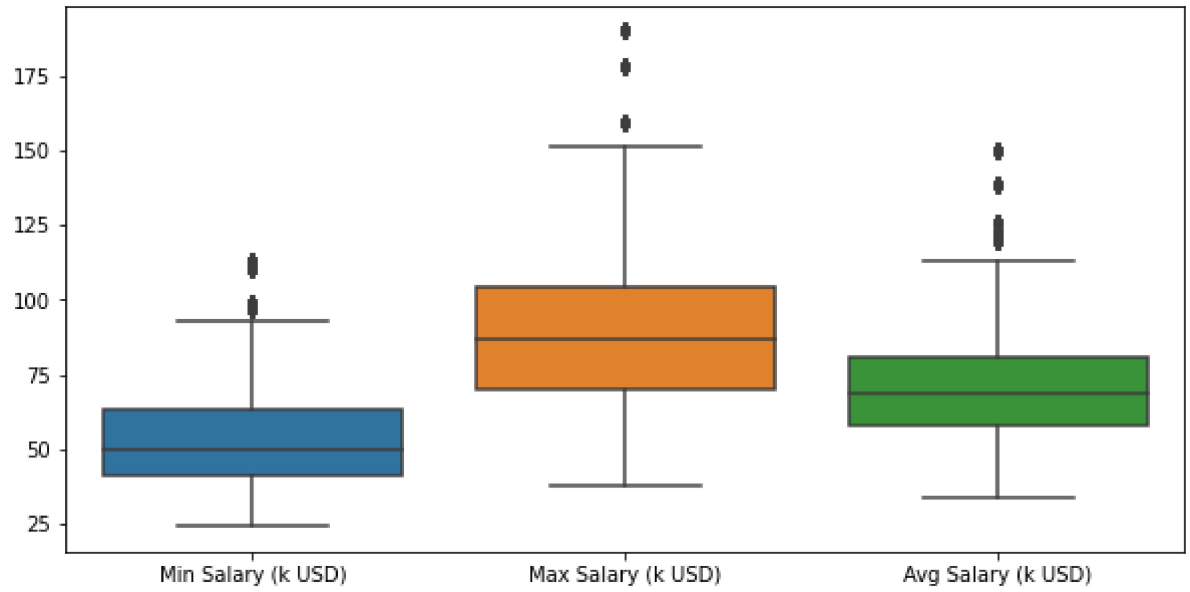
In [57]:
```
#organisation wise size of the company
plt.figure(figsize = (5, 10))
sns.violinplot(data = clean_data, x= "Avg Salary (k USD)", y="Type of ownership")
plt.show()
```



5. Seeing how minimum, maximum and average salaries varies

In [58]:
```
plt.figure(figsize = (10, 5))
salaries = clean_data[['Min Salary (k USD)','Max Salary (k USD)','Avg Salary
 (k USD)']]
sns.boxplot(data = salaries)
```

Out[58]: <AxesSubplot:>



6. Finding the Top 15 Profiles for Jobs

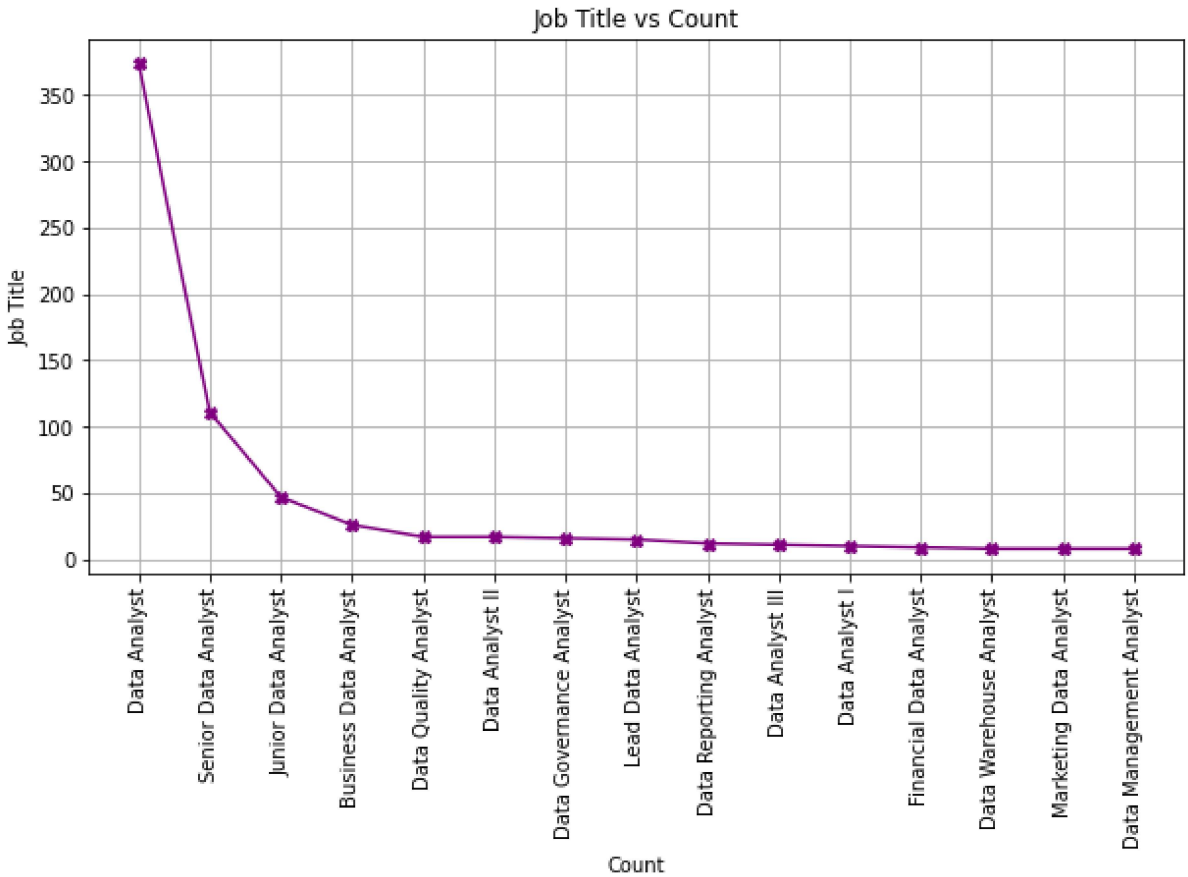
In [59]:
```
top15_job_title = clean_data['Job Title'].value_counts().head(15)
top15_job_title
```

Out[59]:
```
Data Analyst                374
Senior Data Analyst         111
Junior Data Analyst          47
Business Data Analyst        26
Data Quality Analyst         17
Data Analyst II              17
Data Governance Analyst      16
Lead Data Analyst            15
Data Reporting Analyst       12
Data Analyst III             11
Data Analyst I               10
Financial Data Analyst        9
Data Warehouse Analyst        8
Marketing Data Analyst        8
Data Management Analyst       8
Name: Job Title, dtype: int64
```

7. Job Profiles vs Count

In [60]:
```
#for top15 most demanded job titles
plt.figure(figsize=(10,5))
plt.grid()
plt.title("Job Title vs Count")
plt.xlabel("Count")
plt.ylabel("Job Title")
plt.plot(top15_job_title ,color='purple', marker='X')
plt.xticks(labels = None, rotation=90)
```

Out[60]:
```
([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14],
 [Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, '')])
```



Data analyst being high in demand followed by Senior then junior data analyst. The curve is quite constant starting from business analyst showing the almost equal demand in all other kind of analyst.

8). Finding the Top 15 Locations for available Jobs

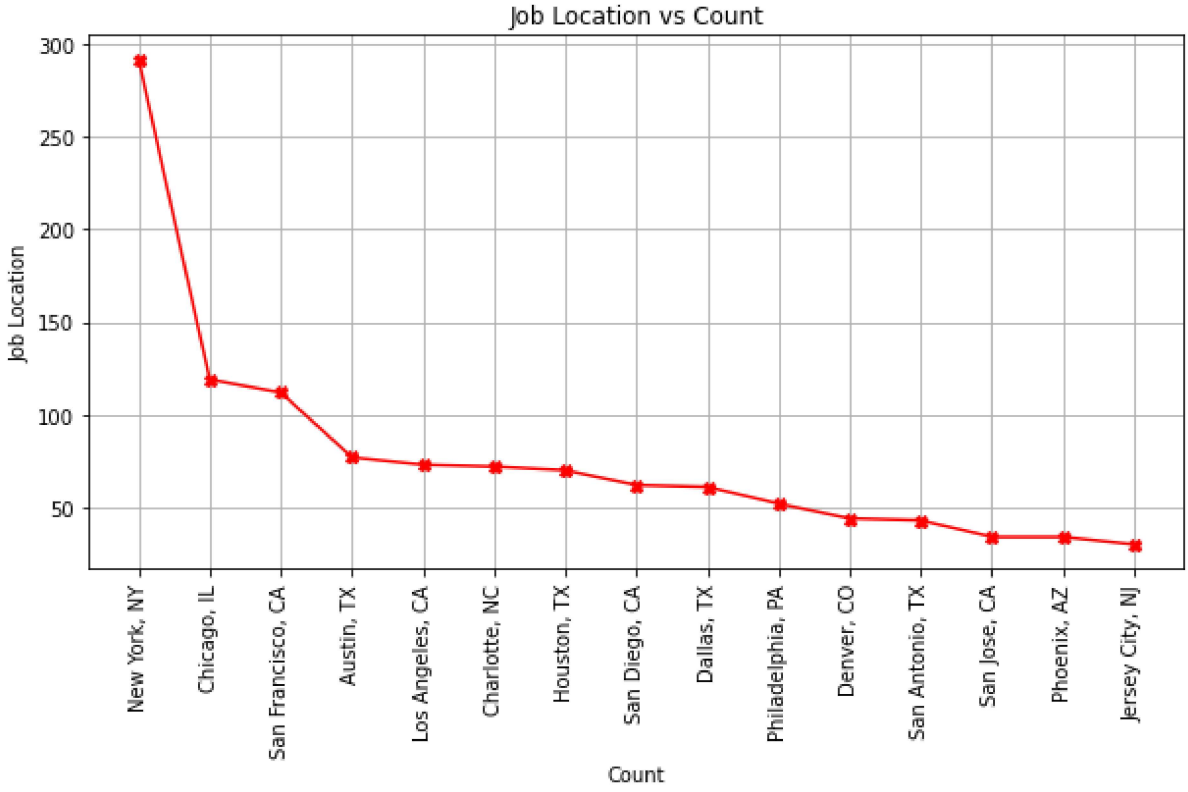
In [61]:
```
top15_job_location = clean_data['Location'].value_counts().head(15)
top15_job_location
```

Out[61]:
```
New York, NY         292
Chicago, IL          119
San Francisco, CA    112
Austin, TX            77
Los Angeles, CA       73
Charlotte, NC         72
Houston, TX           70
San Diego, CA         62
Dallas, TX            61
Philadelphia, PA      52
Denver, CO            44
San Antonio, TX       43
San Jose, CA          34
Phoenix, AZ           34
Jersey City, NJ       30
Name: Location, dtype: int64
```

Job Location vs Count

In [62]:
```
plt.figure(figsize=(10,5))
plt.grid()
plt.title("Job Location vs Count")
plt.xlabel("Count")
plt.ylabel("Job Location")
plt.plot(top15_job_location ,color='red', marker='X')
plt.xticks(labels = None, rotation=90)
```

Out[62]:
```
([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14],
 [Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, '')])
```



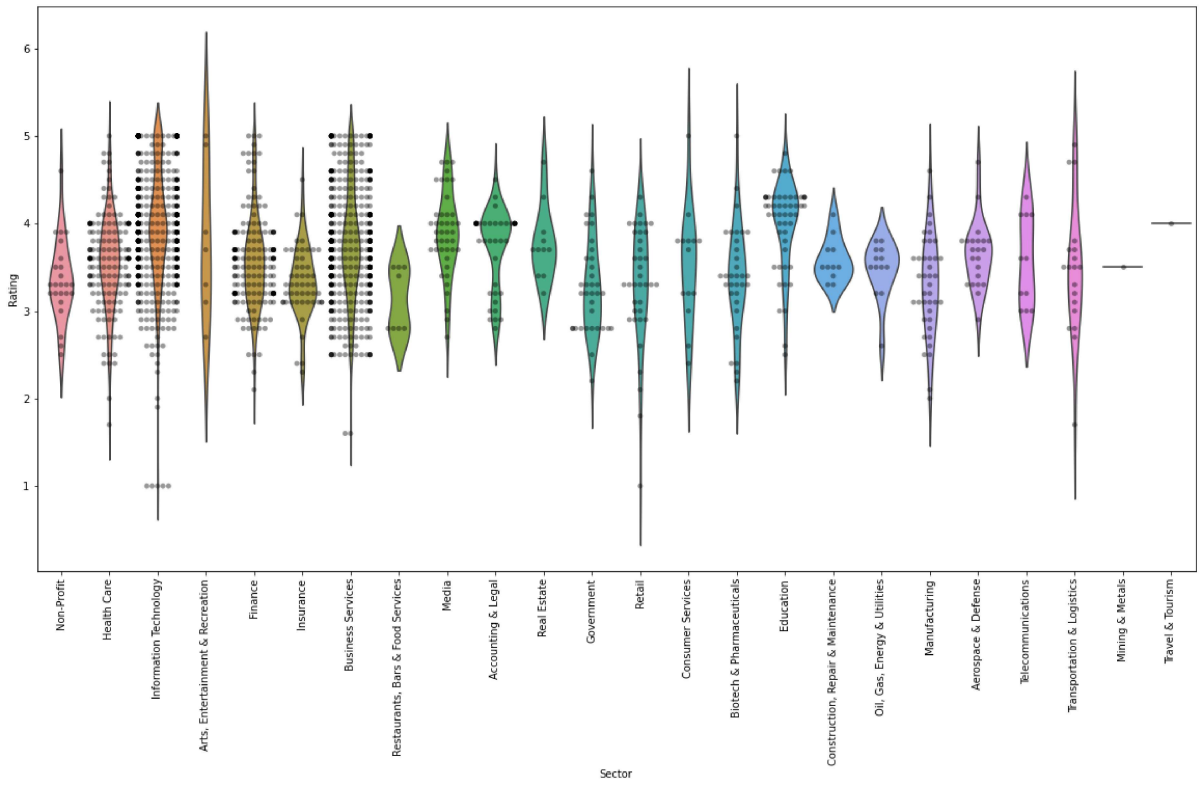
New York providing the highest Number of jobs and Jersey city offering the lowest number of analysis jobs

9. Visualising the Sector Wise rating of companies to check which sectors have what rated companies

In [63]:
```
#sector wise rating of companies
plt.figure(figsize=(20,10))
sns.violinplot(x= 'Sector', y='Rating' , data =clean_data, inner =None)
sns.swarmplot(x= 'Sector', y='Rating', data = clean_data, color= 'k',alpha =
0.4)
plt.xticks(rotation = 90)
```

```
C:\Users\kusha\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserW
arning: 27.8% of the points cannot be placed; you may want to decrease the si
ze of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\kusha\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserW
arning: 67.7% of the points cannot be placed; you may want to decrease the si
ze of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\kusha\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserW
arning: 36.1% of the points cannot be placed; you may want to decrease the si
ze of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\kusha\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserW
arning: 7.8% of the points cannot be placed; you may want to decrease the siz
e of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\kusha\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserW
arning: 63.9% of the points cannot be placed; you may want to decrease the si
ze of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\kusha\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserW
arning: 34.9% of the points cannot be placed; you may want to decrease the si
ze of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\kusha\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserW
arning: 8.3% of the points cannot be placed; you may want to decrease the siz
e of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\kusha\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserW
arning: 15.4% of the points cannot be placed; you may want to decrease the si
ze of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

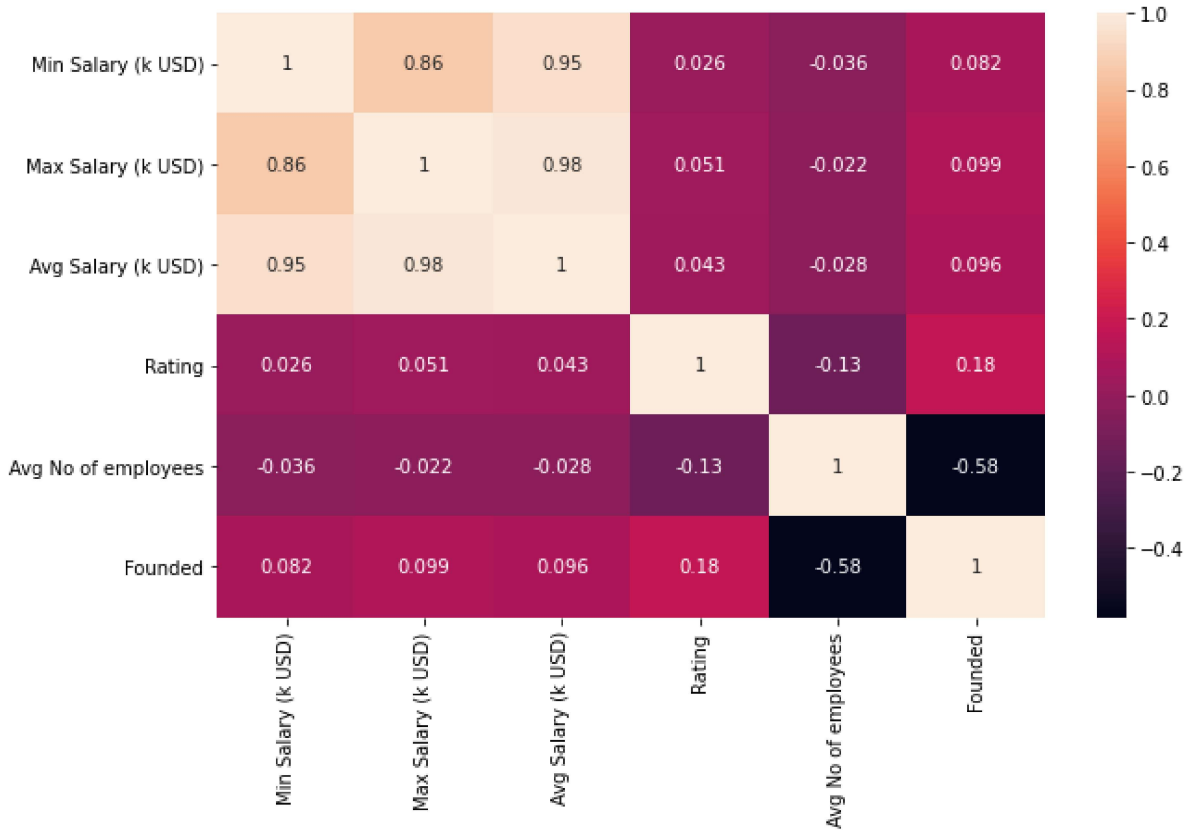
```
Out[63]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
                 17, 18, 19, 20, 21, 22, 23]),
          [Text(0, 0, 'Non-Profit'),
           Text(1, 0, 'Health Care'),
           Text(2, 0, 'Information Technology'),
           Text(3, 0, 'Arts, Entertainment & Recreation'),
           Text(4, 0, 'Finance'),
           Text(5, 0, 'Insurance'),
           Text(6, 0, 'Business Services'),
           Text(7, 0, 'Restaurants, Bars & Food Services'),
           Text(8, 0, 'Media'),
           Text(9, 0, 'Accounting & Legal'),
           Text(10, 0, 'Real Estate'),
           Text(11, 0, 'Government'),
           Text(12, 0, 'Retail'),
           Text(13, 0, 'Consumer Services'),
           Text(14, 0, 'Biotech & Pharmaceuticals'),
           Text(15, 0, 'Education'),
           Text(16, 0, 'Construction, Repair & Maintenance'),
           Text(17, 0, 'Oil, Gas, Energy & Utilities'),
           Text(18, 0, 'Manufacturing'),
           Text(19, 0, 'Aerospace & Defense'),
           Text(20, 0, 'Telecommunications'),
           Text(21, 0, 'Transportation & Logistics'),
           Text(22, 0, 'Mining & Metals'),
           Text(23, 0, 'Travel & Tourism')])
```



10. Correlation between the numeric columns of out dataset

In [64]:
```
plt.figure(figsize=(10,6))
corr = clean_data.corr()
sns.heatmap(corr, annot = True)
```

Out[64]: <AxesSubplot:>



11. Using the job description and merging it with the Job Titles to see what job profile require which skills

In [65]:
```
#jobs with skills
job_titles = pd.DataFrame(clean_data['Job Title'])
job_titles_with_skills = pd.concat([job_titles,job_desc],axis=1)
job_titles_with_skills
```

Out[65]:

	Job Title	python	sql	power bi	excel
0	Data Analyst, Center on Immigration and Justic...	True	True	False	False
1	Quality Data Analyst	False	True	False	True
2	Senior Data Analyst, Insights & Analytics Team...	True	True	False	True
3	Data Analyst	False	True	False	False
4	Reporting Data Analyst	True	True	False	True
...	...	...	...	...	...
2247	Marketing/Communications - Data Analyst-Marketing	False	True	False	True
2248	RQS - IHHA - 201900004460 -1q Data Security An...	False	False	False	False
2249	Senior Data Analyst (Corporate Audit)	False	True	False	False
2251	Data Analyst 3, Customer Experience	True	True	False	False
2252	Senior Quality Data Analyst	False	False	False	True

2090 rows × 5 columns

In [66]:
```
job_titles_with_skills = job_titles_with_skills.groupby('Job Title')[['python'
, 'sql', 'power bi','excel']].sum().sort_values(by='python', ascending=False).
head(10)
```

In [67]:
```
job_titles_with_skills.reset_index(inplace=True)
job_titles_with_skills
```

Out[67]:

	Job Title	python	sql	power bi	excel
0	Data Analyst	118	279	37	222
1	Senior Data Analyst	62	94	16	61
2	Business Data Analyst	10	20	3	15
3	NY Healthcare Data/Reporting Analyst	5	5	0	5
4	Marketing Data Analyst	5	8	0	5
5	Data Analyst III (Healthcare Analytics)	5	5	0	5
6	Lead Data Analyst	5	13	6	7
7	TX Healthcare Data/Reporting Analyst	4	4	0	4
8	Healthcare Data/Reporting Analyst	4	4	0	4
9	Data Analyst II	4	8	2	10

In [68]:
```
job_titles_with_skills['no_of_jobs'] = clean_data['Job Title'].value_counts()
[:10].values
```

In [69]:
```
job_titles_with_skills.columns
job_titles_with_skills
```

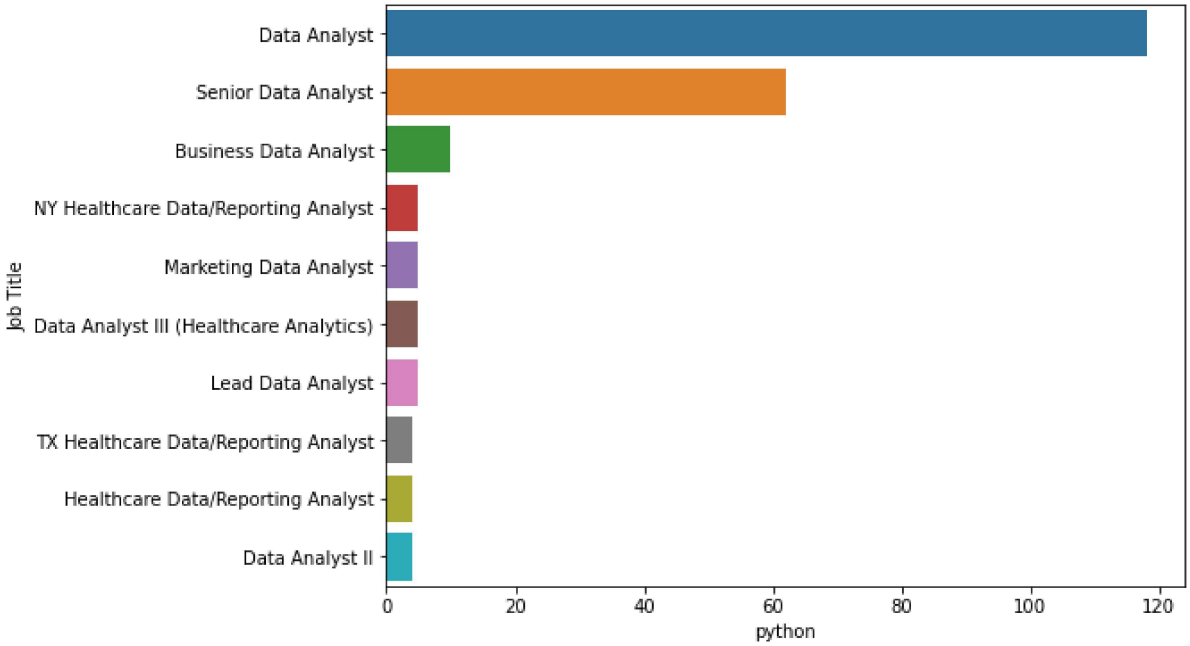
Out[69]:

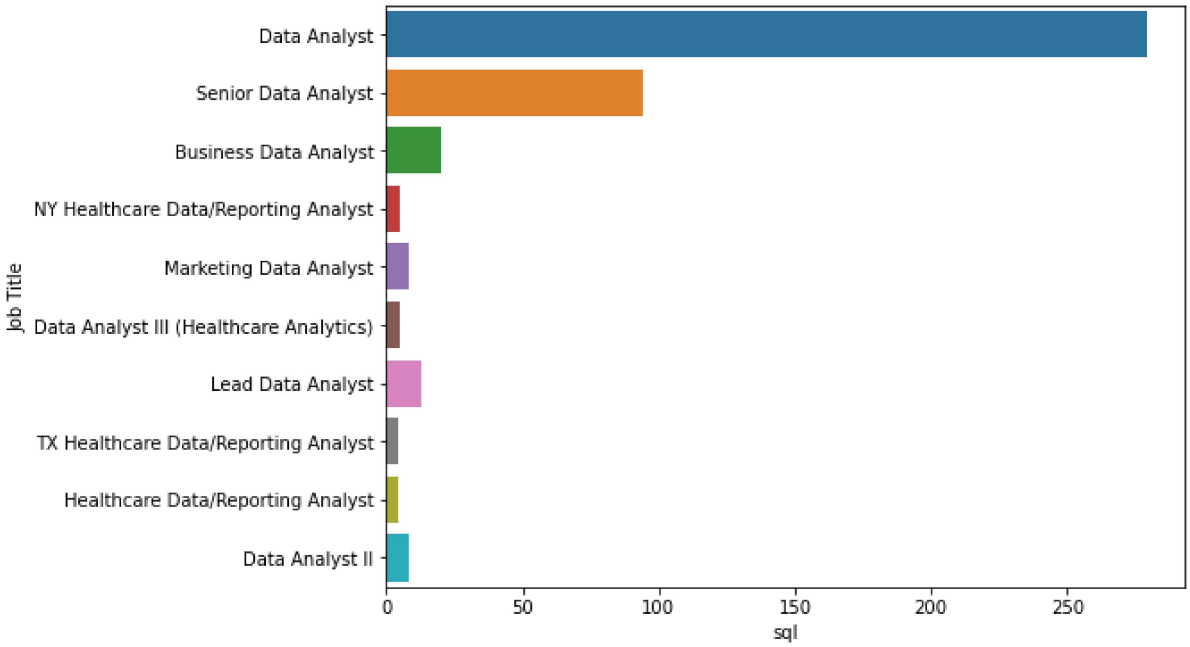
	Job Title	python	sql	power bi	excel	no_of_jobs
0	Data Analyst	118	279	37	222	374
1	Senior Data Analyst	62	94	16	61	111
2	Business Data Analyst	10	20	3	15	47
3	NY Healthcare Data/Reporting Analyst	5	5	0	5	26
4	Marketing Data Analyst	5	8	0	5	17
5	Data Analyst III (Healthcare Analytics)	5	5	0	5	17
6	Lead Data Analyst	5	13	6	7	16
7	TX Healthcare Data/Reporting Analyst	4	4	0	4	15
8	Healthcare Data/Reporting Analyst	4	4	0	4	12
9	Data Analyst II	4	8	2	10	11

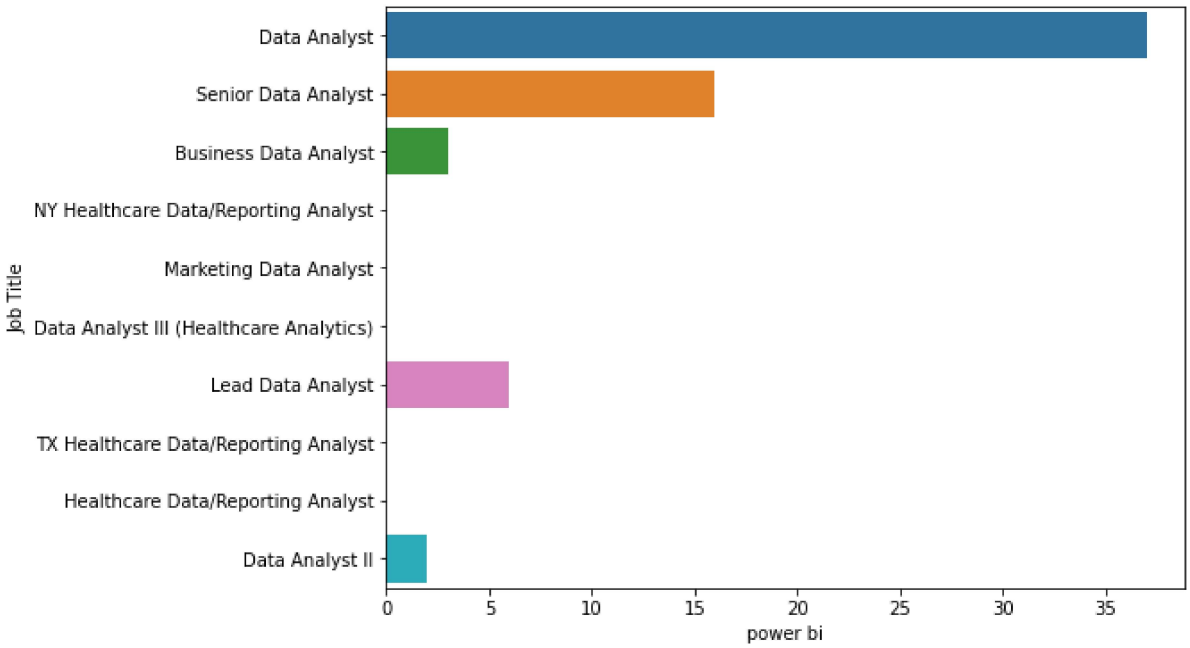
Visualising each skill required for the Job Titles

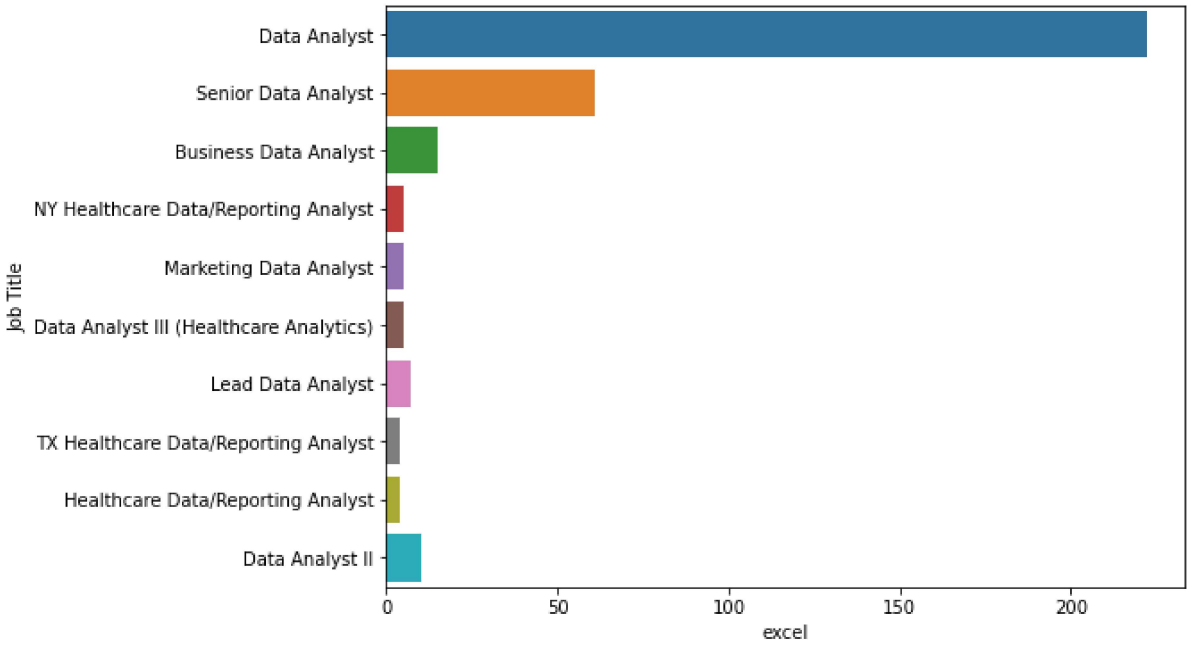
In [70]:
```
def job_title_vs_skills_required(dataset, x_axis, y_axis ):
    plt.figure(figsize=(8,6))
    sns.barplot(x = x_axis, y = y_axis, data = dataset )

#for python
job_title_vs_skills_required(job_titles_with_skills, 'python','Job Title')
job_title_vs_skills_required(job_titles_with_skills, 'sql','Job Title')
job_title_vs_skills_required(job_titles_with_skills, 'power bi','Job Title')
job_title_vs_skills_required(job_titles_with_skills, 'excel','Job Title')
```


Data Analyst
Senior Data Analyst
Business Data Analyst
NY Healthcare Data/Reporting Analyst
Marketing Data Analyst
Data Analyst III (Healthcare Analytics)
Lead Data Analyst
TX Healthcare Data/Reporting Analyst
Healthcare Data/Reporting Analyst
Data Analyst II
Job Title
0
20
40
60
80
100
120
python


Data Analyst
Senior Data Analyst
Business Data Analyst
NY Healthcare Data/Reporting Analyst
Marketing Data Analyst
Data Analyst III (Healthcare Analytics)
Lead Data Analyst
TX Healthcare Data/Reporting Analyst
Healthcare Data/Reporting Analyst
Data Analyst II
Job Title
0
50
100
150
200
250
sql





In all the above four graphs job profile of Data Analyst requires all the four kind of skills i.e. python, sql, power bi and excel

12. Melting the dataframe for more clear representation

In [71]:
```
melted_df = pd.melt(job_titles_with_skills, id_vars =['Job Title'], value_vars
= ['python', 'sql', 'power bi','excel','no_of_jobs'], var_name = 'skills', val
ue_name = 'count')
melted_df
```

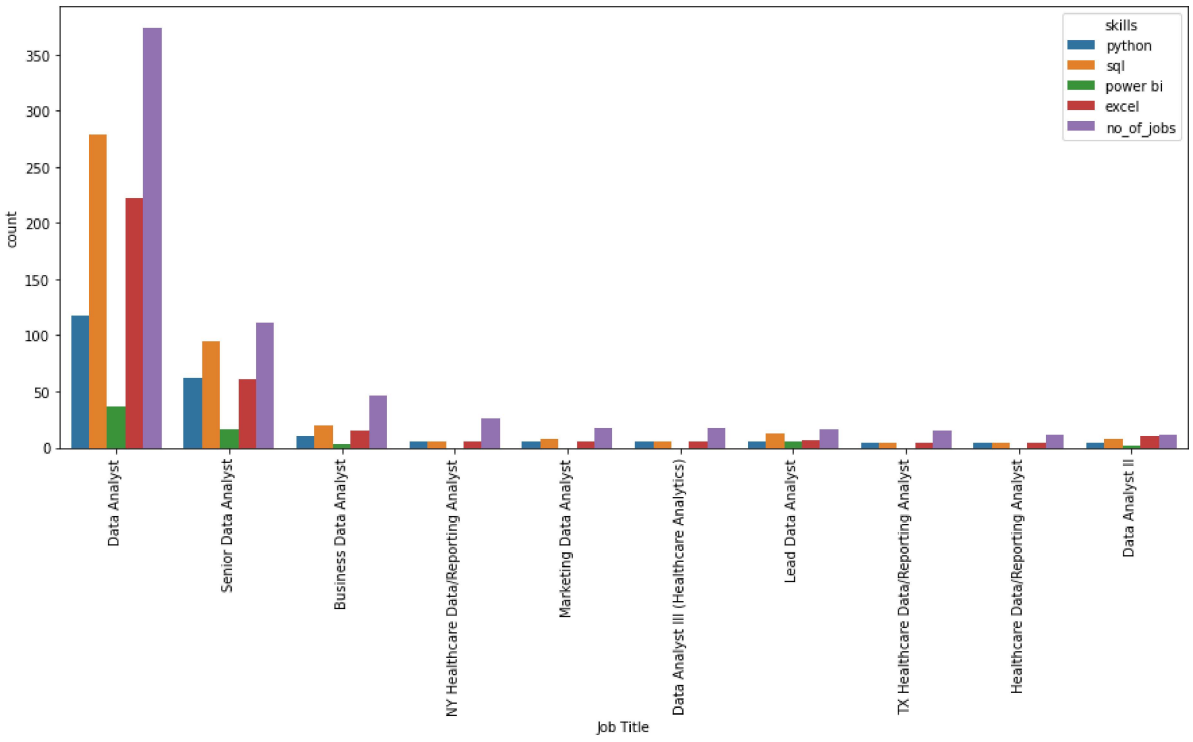
Out[71]:

	Job Title	skills	count
0	Data Analyst	python	118
1	Senior Data Analyst	python	62
2	Business Data Analyst	python	10
3	NY Healthcare Data/Reporting Analyst	python	5
4	Marketing Data Analyst	python	5
5	Data Analyst III (Healthcare Analytics)	python	5
6	Lead Data Analyst	python	5
7	TX Healthcare Data/Reporting Analyst	python	4
8	Healthcare Data/Reporting Analyst	python	4
9	Data Analyst II	python	4
10	Data Analyst	sql	279
11	Senior Data Analyst	sql	94
12	Business Data Analyst	sql	20
13	NY Healthcare Data/Reporting Analyst	sql	5
14	Marketing Data Analyst	sql	8
15	Data Analyst III (Healthcare Analytics)	sql	5
16	Lead Data Analyst	sql	13
17	TX Healthcare Data/Reporting Analyst	sql	4
18	Healthcare Data/Reporting Analyst	sql	4
19	Data Analyst II	sql	8
20	Data Analyst	power bi	37
21	Senior Data Analyst	power bi	16
22	Business Data Analyst	power bi	3
23	NY Healthcare Data/Reporting Analyst	power bi	0
24	Marketing Data Analyst	power bi	0
25	Data Analyst III (Healthcare Analytics)	power bi	0
26	Lead Data Analyst	power bi	6
27	TX Healthcare Data/Reporting Analyst	power bi	0
28	Healthcare Data/Reporting Analyst	power bi	0
29	Data Analyst II	power bi	2
30	Data Analyst	excel	222
31	Senior Data Analyst	excel	61
32	Business Data Analyst	excel	15
33	NY Healthcare Data/Reporting Analyst	excel	5
34	Marketing Data Analyst	excel	5

	Job Title	skills	count
35	Data Analyst III (Healthcare Analytics)	excel	5
36	Lead Data Analyst	excel	7
37	TX Healthcare Data/Reporting Analyst	excel	4
38	Healthcare Data/Reporting Analyst	excel	4
39	Data Analyst II	excel	10
40	Data Analyst	no_of_jobs	374
41	Senior Data Analyst	no_of_jobs	111
42	Business Data Analyst	no_of_jobs	47
43	NY Healthcare Data/Reporting Analyst	no_of_jobs	26
44	Marketing Data Analyst	no_of_jobs	17
45	Data Analyst III (Healthcare Analytics)	no_of_jobs	17
46	Lead Data Analyst	no_of_jobs	16
47	TX Healthcare Data/Reporting Analyst	no_of_jobs	15
48	Healthcare Data/Reporting Analyst	no_of_jobs	12
49	Data Analyst II	no_of_jobs	11

In [72]:
```
plt.figure(figsize=(15,6))
sns.barplot(data = melted_df , x='Job Title',y='count',hue='skills')
plt.xticks(rotation = 90)
```

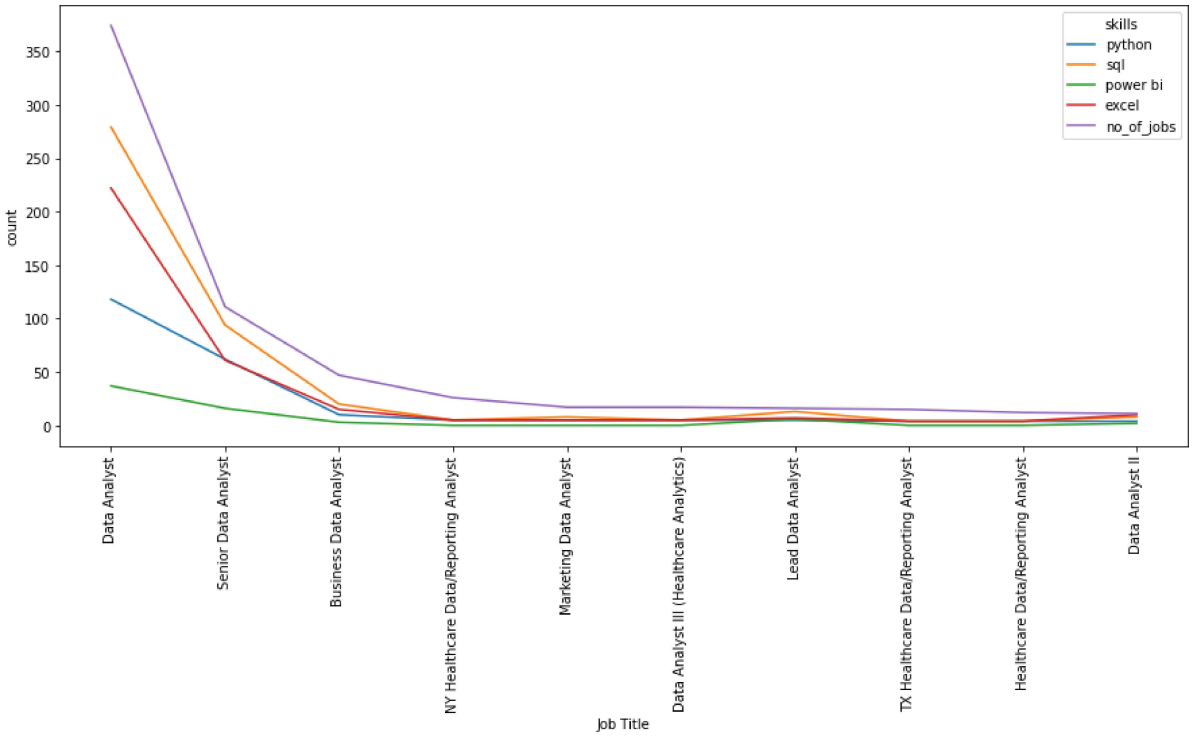
Out[72]:
```
(array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
 [Text(0, 0, 'Data Analyst'),
  Text(1, 0, 'Senior Data Analyst'),
  Text(2, 0, 'Business Data Analyst'),
  Text(3, 0, 'NY Healthcare Data/Reporting Analyst'),
  Text(4, 0, 'Marketing Data Analyst'),
  Text(5, 0, 'Data Analyst III (Healthcare Analytics)'),
  Text(6, 0, 'Lead Data Analyst'),
  Text(7, 0, 'TX Healthcare Data/Reporting Analyst'),
  Text(8, 0, 'Healthcare Data/Reporting Analyst'),
  Text(9, 0, 'Data Analyst II')])
```



We observe that Excel and SQL are the most popular skills required for each job profile, followed by python and power bi

In [73]:
```
plt.figure(figsize=(15,6))
sns.lineplot(x="Job Title", y="count",
             hue="skills",
             data=melted_df)
plt.xticks(rotation =90)
```

Out[73]:
```
([0, 1, 2, 3, 4, 5, 6, 7, 8, 9],
 [Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, '')])
```



Finding the percentage of each skill for the Job Profiles to confirm

In [74]:
```
#pie plots for these (??)
```

In [75]:
```
job_titles_with_skills.python.sum()/job_titles_with_skills.no_of_jobs.sum()*100
```

Out[75]:
```
34.36532507739938
```

In [76]:
```
job_titles_with_skills.sql.sum()/job_titles_with_skills.no_of_jobs.sum()*100
```

Out[76]:
```
68.11145510835914
```

In [77]:
```
job_titles_with_skills.excel.sum()/job_titles_with_skills.no_of_jobs.sum()*100
```

Out[77]: 52.32198142414861

In [78]:
```
job_titles_with_skills['power bi'].sum()/job_titles_with_skills.no_of_jobs.sum
()*100
```

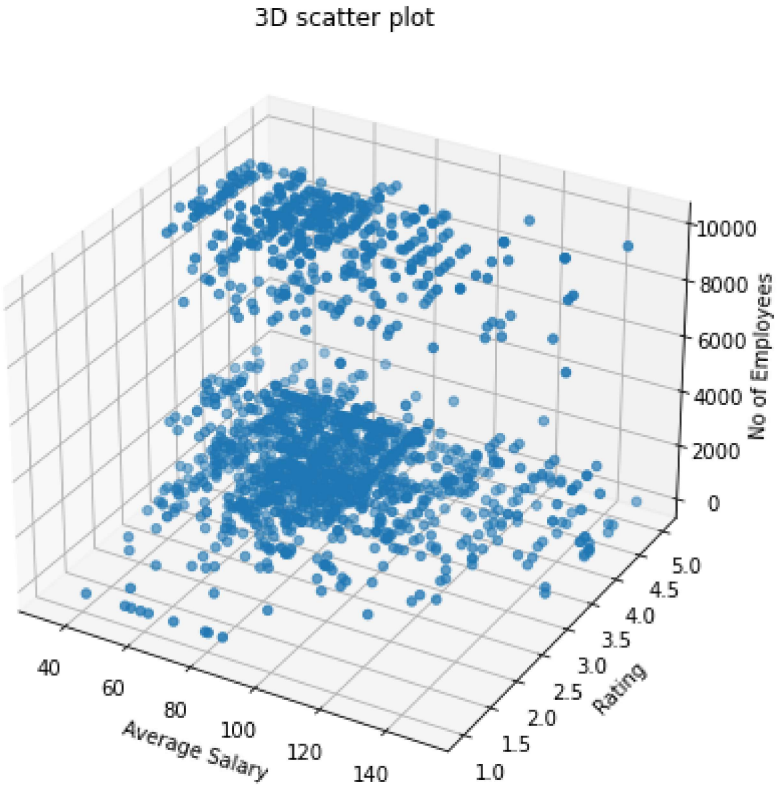
Out[78]: 9.907120743034056

Thus SQL, Excel and Python are the most demanded skills

13. Plot for Average Salary, No of Employees and Rating of the Job Profiles

In [96]:
```
from mpl_toolkits import mplot3d

x = clean_data['Avg Salary (k USD)']
y = clean_data['Rating']
z = clean_data['Avg No of employees']
fig = plt.figure(figsize = (10, 7))
ax = plt.axes(projection ="3d")
ax.scatter3D(x, y, z)
ax.set_xlabel("Average Salary")
ax.set_ylabel("Rating")
ax.set_zlabel("No of Employees")
plt.title("3D scatter plot")
plt.show()
```



Plotting a 3D plot for Average Salary, No of Employees and Rating of the Job Profiles, we notice how good rated companies have a smaller size and good average salary whereas the average salary decreases when the size is increased

14. Plotting scatter plot for average salary and Rating

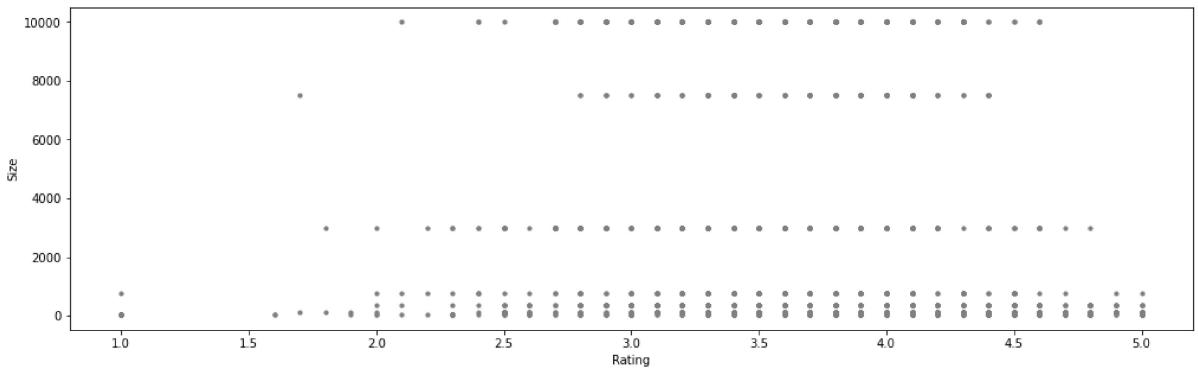
In [80]:
```
#average salary with rating
plt.figure(figsize=(17,5))
plt.scatter('Avg Salary (k USD)','Rating', data = clean_data, s=10, c = "purpl
e")
plt.xlabel("Average Salary")
plt.ylabel("Rating")
```

Out[80]: Text(0, 0.5, 'Rating')

15. Plotting average salary with Size

In [81]:
```
plt.figure(figsize=(17,5))
plt.scatter('Rating','Avg No of employees', data = clean_data, s=10, c = "gra
y")
plt.xlabel("Rating")
plt.ylabel("Size")
```

Out[81]: Text(0, 0.5, 'Size')



16. Plotting Average Salary, Ratings And Size for Companies founded before and after 2015

In [82]:
```
plt.figure(figsize = (20, 5))
salary_rating_size = clean_data.loc[:,['Avg Salary (k USD)','Rating','Avg No of employees','Founded']]
salary_rating_size['Founded'] = salary_rating_size.Founded.apply(lambda x:"After 2015" if x>2015 else "Before 2015")
salary_rating_size
```

Out[82]:

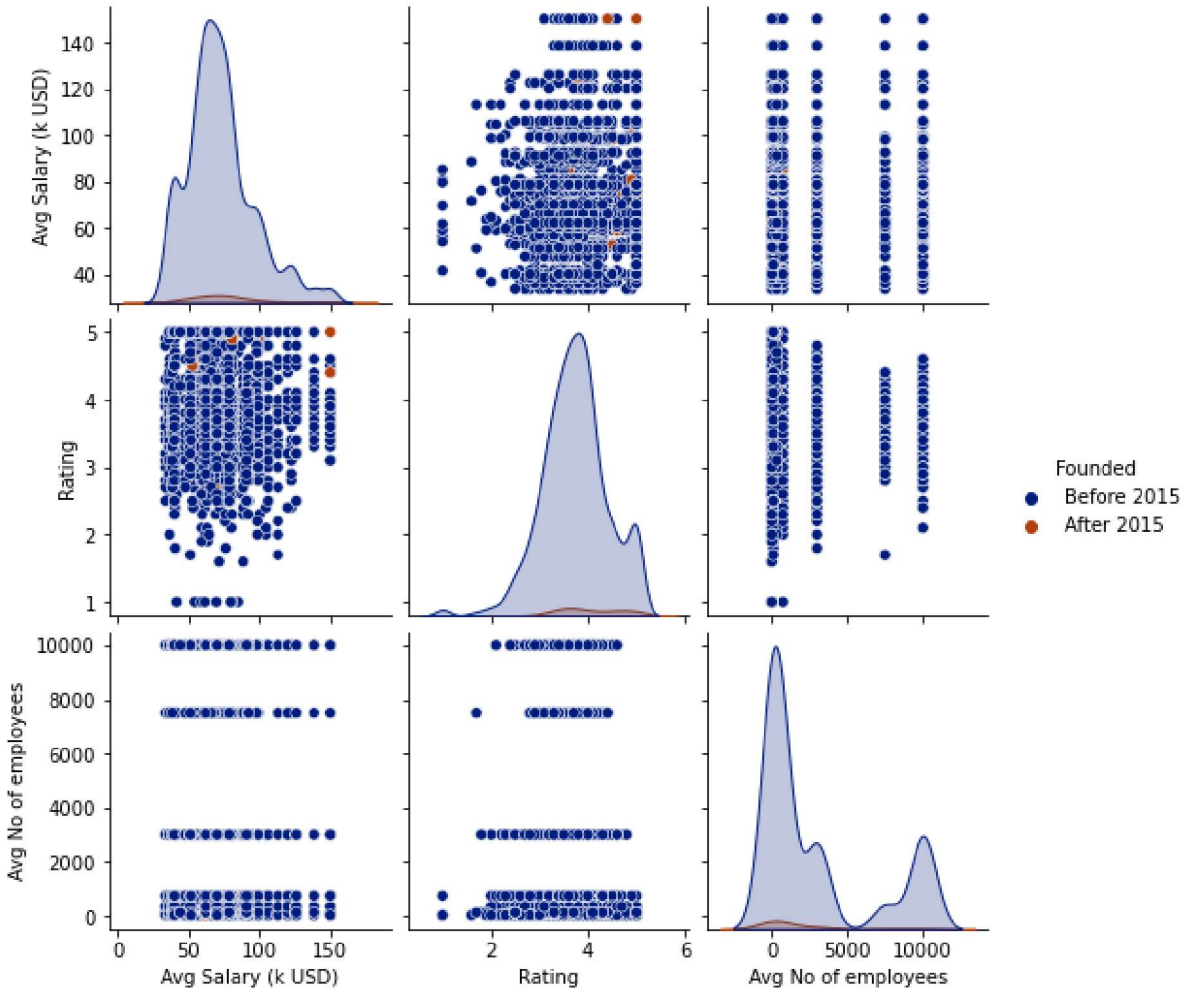
	Avg Salary (k USD)	Rating	Avg No of employees	Founded
0	51.5	3.2	350.5	Before 2015
1	51.5	3.8	10000.0	Before 2015
2	51.5	3.4	3000.5	Before 2015
3	51.5	4.1	350.5	Before 2015
4	51.5	3.9	750.5	Before 2015
...	...	...	...	...
2247	91.0	4.1	125.5	Before 2015
2248	91.0	2.5	125.5	Before 2015
2249	91.0	2.9	10000.0	Before 2015
2251	91.0	3.1	350.5	Before 2015
2252	91.0	3.4	10000.0	Before 2015

2090 rows × 4 columns

```
<Figure size 1440x360 with 0 Axes>
```

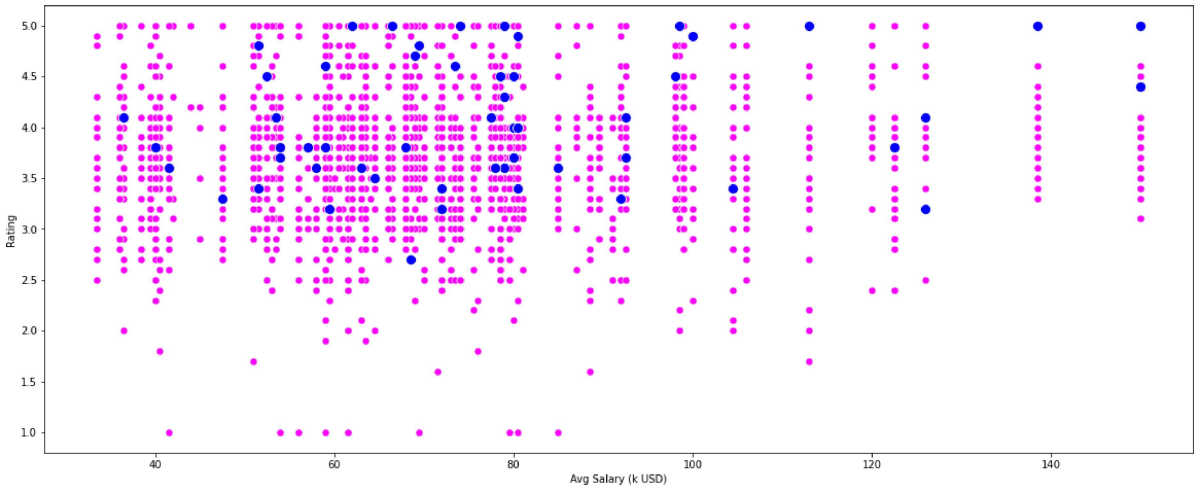
In [83]:
```
sns.pairplot(salary_rating_size , hue = 'Founded', palette = "dark")
```

Out[83]: <seaborn.axisgrid.PairGrid at 0x1cc801fb9a0>



In [84]:
```
plt.figure(figsize = (20, 8))
sns.scatterplot(data = salary_rating_size[salary_rating_size['Founded'] == "Be
fore 2015"], x ="Avg Salary (k USD)", y ="Rating" , color="magenta" , s=50)
sns.scatterplot(data = salary_rating_size[salary_rating_size['Founded'] == "Af
ter 2015"], x ="Avg Salary (k USD)", y ="Rating", color="blue" ,s=100)
```

Out[84]: <AxesSubplot:xlabel='Avg Salary (k USD)', ylabel='Rating'>



17. Merging Dataframes to include the skills dataset creating a new one

In [85]:
```
#merging dataframes to see skills required in each sector
df_with_skills = pd.merge(left = clean_data, right = job_desc,left_index = Tru
e, right_index = True)
```

In [86]:
```
sector_with_skills = df_with_skills.groupby('Sector')[['python', 'sql','power
 bi','excel']].sum().sort_values(by = 'sql', ascending = False)
sector_with_skills.reset_index(inplace= True)
#in top 5
sector_with_skills
```

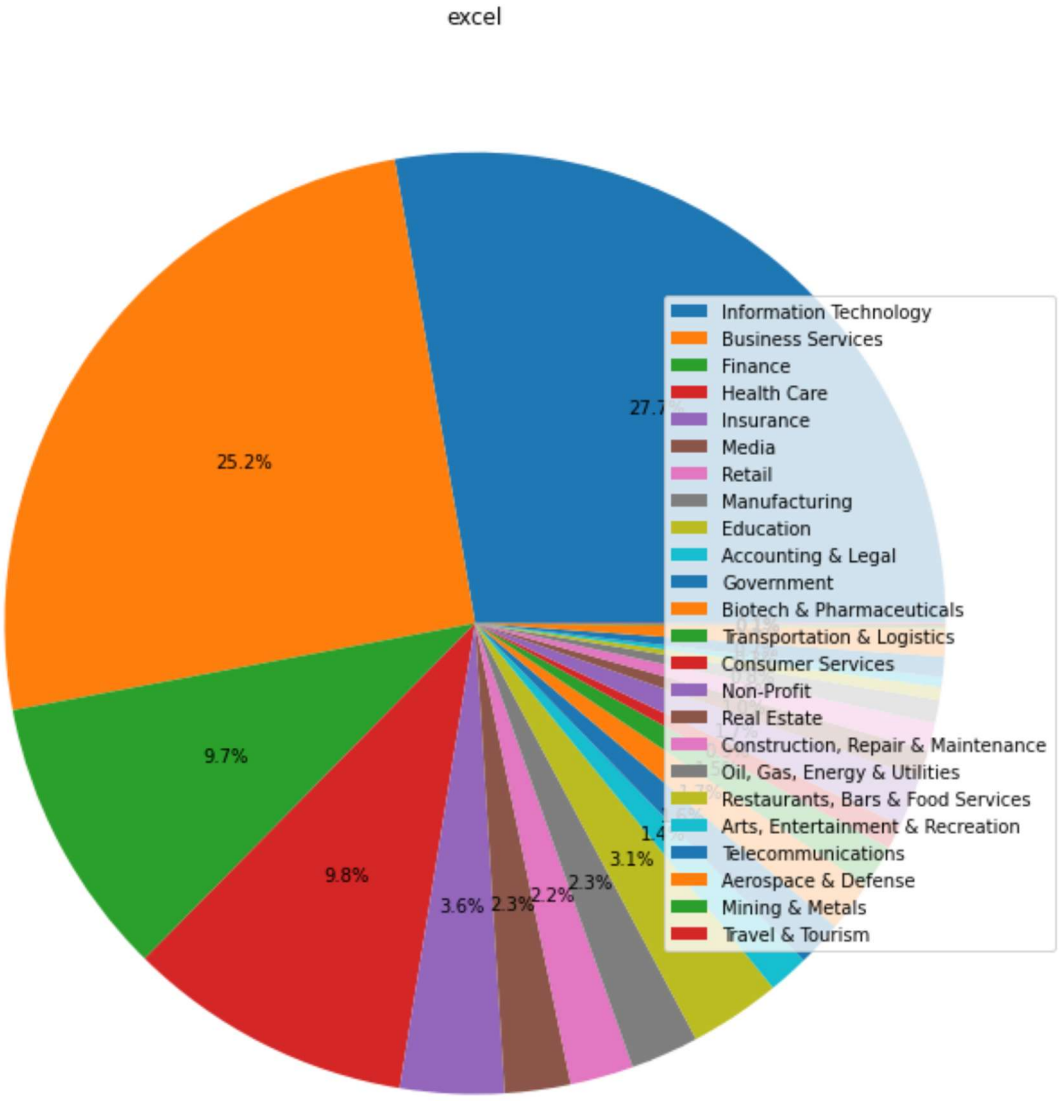
Out[86]:

	Sector	python	sql	power bi	excel
0	Information Technology	167	396	31	319
1	Business Services	125	314	42	290
2	Finance	63	98	17	111
3	Health Care	37	97	11	113
4	Insurance	12	38	4	41
5	Media	20	33	5	26
6	Retail	19	24	4	25
7	Manufacturing	14	24	6	27
8	Education	17	23	3	36
9	Accounting & Legal	5	20	5	16
10	Government	8	17	2	18
11	Biotech & Pharmaceuticals	12	15	4	20
12	Transportation & Logistics	10	14	2	17
13	Consumer Services	5	12	1	10
14	Non-Profit	7	12	5	20
15	Real Estate	7	10	3	11
16	Construction, Repair & Maintenance	2	8	3	11
17	Oil, Gas, Energy & Utilities	5	6	0	9
18	Restaurants, Bars & Food Services	3	6	1	5
19	Arts, Entertainment & Recreation	3	6	1	4
20	Telecommunications	5	6	4	8
21	Aerospace & Defense	4	4	0	11
22	Mining & Metals	1	1	0	1
23	Travel & Tourism	1	1	0	1

Sector wise division for "Excel" as their skill requirement

In [87]:
```
labels = sector_with_skills['Sector']
fig, ax1 = plt.subplots(figsize = (12,12))
plt.title('excel')
ax1.pie(sector_with_skills['excel'], autopct='%1.1f%%')
ax1.legend(labels, loc ="right")
```

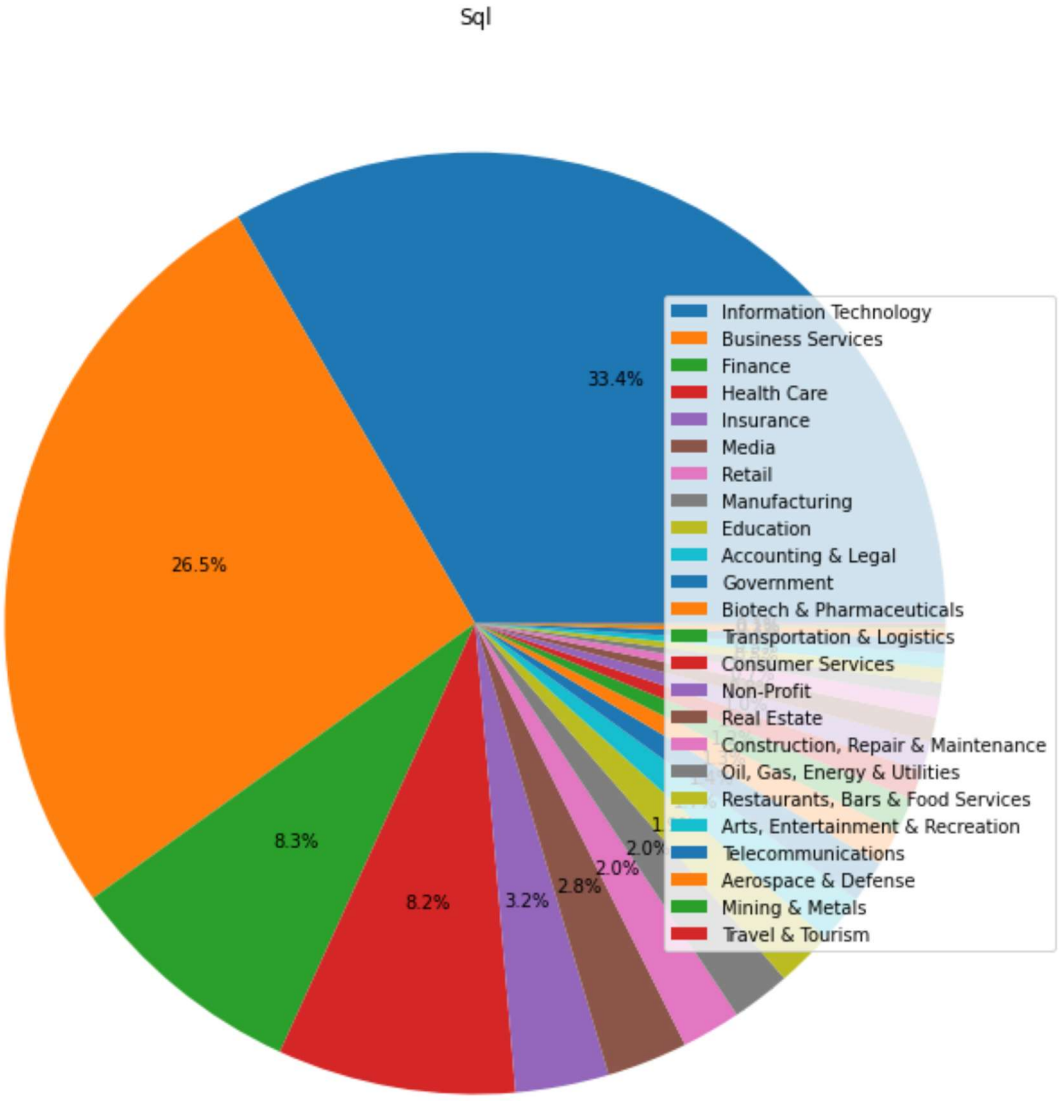
Out[87]: <matplotlib.legend.Legend at 0x1cc815e71f0>



Sector wise division for "SQL" as their skill requirement

In [88]:
```
labels = sector_with_skills['Sector']
fig, ax1 = plt.subplots(figsize = (12,12))
plt.title('Sql')
ax1.pie(sector_with_skills['sql'], autopct='%1.1f%%')
ax1.legend(labels, loc ="right")
```

Out[88]: <matplotlib.legend.Legend at 0x1cc81c709a0>



Most of the Information technology and business services companies ask for SQL and Excel as skill requirement as not everyone is a computer or coding geek, SQL and Excel are always high in demand in IT and business sector as well

18. Skills requirement depending on type of ownership and avg_salary and Rating

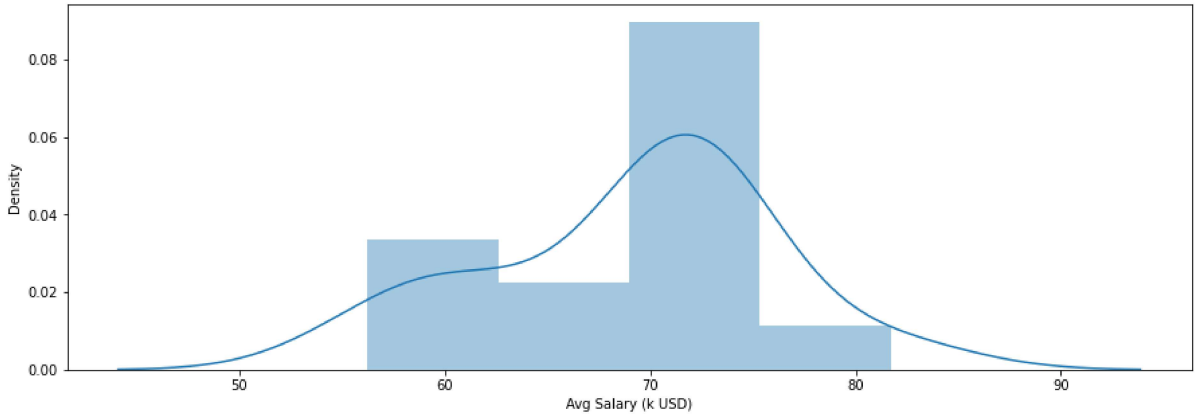
In [89]:
```
#seeing the skills required for type of ownership wise with their average sala
ry and rating
org_with_skills = df_with_skills.groupby('Type of ownership')[['python', 'sql'
,'power bi','excel']].sum()
#in top 5
org_with_skills['Avg Salary (k USD)'] = df_with_skills.groupby('Type of owners
hip')['Avg Salary (k USD)'].mean()
org_with_skills['Rating'] = df_with_skills.groupby('Type of ownership')['Ratin
g'].mean()
org_with_skills.reset_index(inplace= True)
org_with_skills.sort_values(by='Avg Salary (k USD)', ascending = False)
```

Out[89]:

	Type of ownership	python	sql	power bi	excel	Avg Salary (k USD)	Rating
6	Hospital	4	8	0	15	81.710526	3.747368
13	Unknown	6	10	1	13	73.593750	3.481818
0	College / University	10	11	2	24	73.338235	4.108824
2	Company - Public	143	270	41	287	72.701327	3.579254
3	Contract	4	6	2	3	72.500000	3.666667
1	Company - Private	369	818	101	725	72.499607	3.835430
9	Private Practice / Firm	0	6	2	8	71.611111	3.900000
8	Other Organization	1	6	0	7	70.692308	3.033333
12	Subsidiary or Business Segment	25	58	7	53	70.398876	3.585393
7	Nonprofit Organization	24	74	9	87	68.625000	3.437607
4	Franchise	2	2	1	0	63.500000	3.950000
5	Government	10	22	4	24	60.337838	3.305556
10	School / School District	2	4	0	7	59.277778	3.550000
11	Self-employed	1	2	0	2	56.250000	3.000000

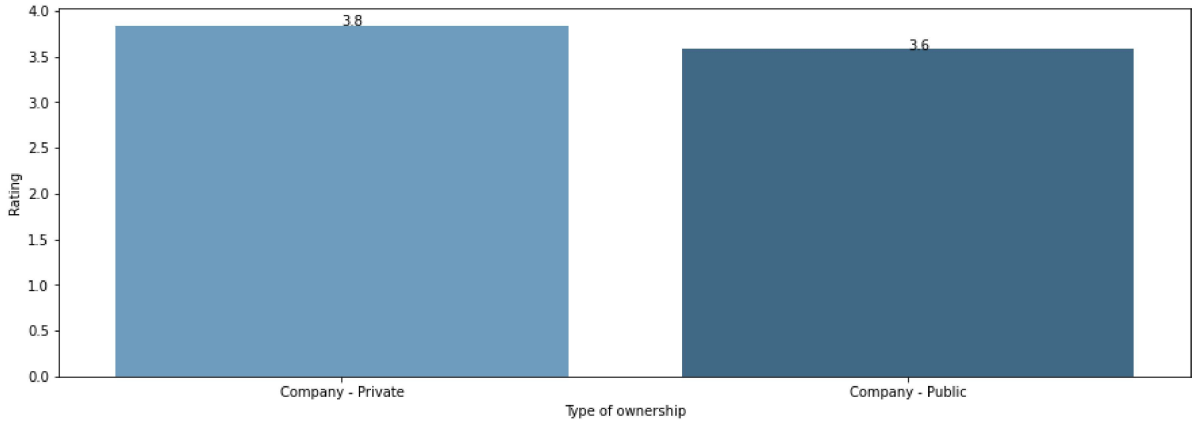
In [90]:
```
plt.figure(figsize = (15, 5))
sns.distplot(org_with_skills['Avg Salary (k USD)'], kde = True)
plt.show()
```

```
C:\Users\kusha\anaconda3\lib\site-packages\seaborn\distributions.py:2557: Fut
ureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level fu
nction with similar flexibility) or `histplot` (an axes-level function for hi
stograms).
  warnings.warn(msg, FutureWarning)
```

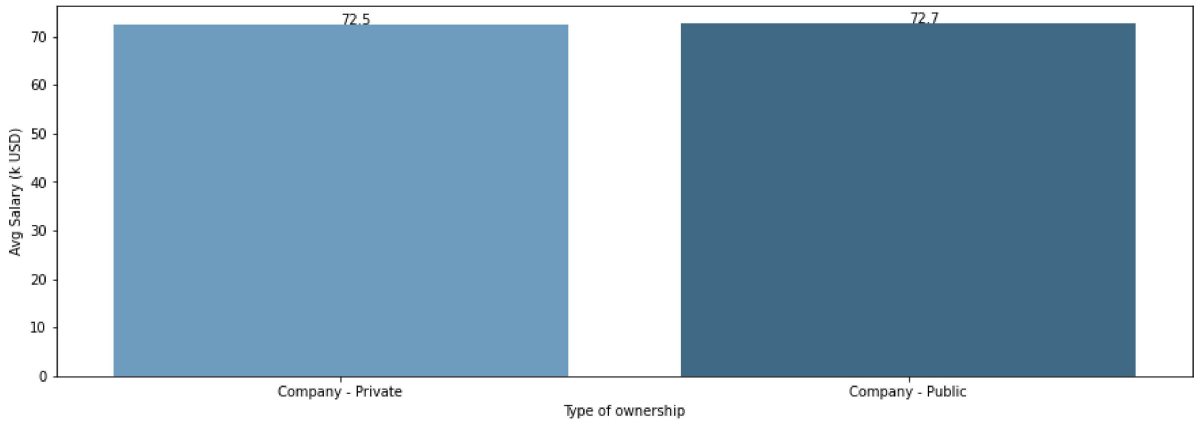


19. Checking average_salary and Rating for the Private and Public Type of Companies

In [91]:
```
#for type of organisation as public or private
fig_dims = (15, 5)
fig, ax = plt.subplots(figsize = fig_dims)
type = ["Company - Public" , "Company - Private"]
d2 = org_with_skills[org_with_skills['Type of ownership'].isin(type)]
g = sns.barplot(x = d2["Type of ownership"], y = d2.Rating, palette="Blues_d")
for index, row in d2.iterrows():
    g.text(row.name -1, row.Rating , round(row.Rating,1), color='black')
plt.show()
```



In [92]:
```
#for type of organisation as public or private
fig_dims = (15, 5)
fig, ax = plt.subplots(figsize = fig_dims)
type = ["Company - Public" , "Company - Private"]
d2 = org_with_skills[org_with_skills['Type of ownership'].isin(type)]
ax = sns.barplot(x = d2["Type of ownership"], y = d2['Avg Salary (k USD)']  ,p
alette="Blues_d")
for index, row in d2.iterrows():
    ax.text(row.name -1, row['Avg Salary (k USD)'] , round(row['Avg Salary (k
 USD)'],2), color='black')
plt.show()
```



G). Conclusion

Numerical Analysis -

1. The average salary being offered by companies that were founded in the 21st century - 74.44 K
2. Biotech & Pharmaceuticals sector offers highest average salary of 83.11 K followed by Real Estate and Arts,Entertainment and IT.
3. NeW York offers most of the analysis job in the USA with 292 companies working in this field.
4. The latest company found in the data analysis field is in 2019 in this dataset which shows that such firms and jobs role are high in demand in today's world.
5. Although Biotech & Pharmaceuticals sector provides highest salary but IT, Business Services and Finance tops the list of sector that provides highest number of jobs,etc.

Graphical Analysis -

As shown above in graphs, with the help of matplotlib and seaborn we tried to visualize various important insights from our data set such as average, minimum and maximum salary across sectors and company wise. Rating of various companies. Observing that Excel and SQL are the most popular skills required for each job profile, followed by python and power bi and many more.

Describing Data science jobs as the "finest job od XXI century" sounds so tempting and right at many levels.

Companies across any industry will benefit from the work of data analysts. From medicine, IT, tourism, retail to fast food chains, data analysis helps in almost all marketing and financial aspects. The insights given by the data analysts are very valuable to the employers who need to have a knowledge about their consumers and end users.

H). References

[1] McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. 2nd edition. O'Reilly Media.

[2] O'Neil, C., & Schutt, R. (2013). Doing Data Science: Straight Talk from the Frontline O'Reilly Media

[3] Data Analysis Job Market in the United States - https://www.kaggle.com/andrewmvd/data-analyst-jobs (https://www.kaggle.com/andrewmvd/data-analyst-jobs)