Python Project – Data Analysis and Manipulation:

Background: I have downloaded a csv file from Kaggle which contains data of jobs and salaries in the data science field from 2020 to 2023. This dataset consists of 12 columns and 9355 rows (https://www.kaggle.com/datasets/hummaamqaasim/jobs-in-data).

Objective: For this project I have used libraries such as Pandas, Matplotlib and Seaborn to analyze, manipulate, clean, and visualize qualitative and quantitative data.

Steps Taken:

1. The first step in this project was to upload the csv file in Jupyter Notebook and store it in a new variable called 'jobs_in_data':

```
In [2]: import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sn

In [3]: jobs_in_data = pd.read_csv("jobs_in_data.csv")

In [4]: jobs_in_data
```

Out[4]:

	work_year	job_title	job_category	salary_currency	salary	salary_in_usd	employee_residence	experience_level	employment_type	work
0	2023	Data DevOps Engineer	Data Engineering	EUR	88000	95012	Germany	Mid-level	Full-time	
1	2023	Data Architect	Data Architecture and Modeling	USD	186000	186000	United States	Senior	Full-time	I
2	2023	Data Architect	Data Architecture and Modeling	USD	81800	81800	United States	Senior	Full-time	I
3	2023	Data Scientist	Data Science and Research	USD	212000	212000	United States	Senior	Full-time	I
4	2023	Data Scientist	Data Science and Research	USD	93300	93300	United States	Senior	Full-time	I
...	...	...	...	...	...	...	...	...	...	
9350	2021	Data Specialist	Data Management and Strategy	USD	165000	165000	United States	Senior	Full-time	
9351	2020	Data Scientist	Data Science and Research	USD	412000	412000	United States	Senior	Full-time	
9352	2021	Principal Data Scientist	Data Science and Research	USD	151000	151000	United States	Mid-level	Full-time	
9353	2020	Data Scientist	Data Science and Research	USD	105000	105000	United States	Entry-level	Full-time	
9354	2020	Business Data Analyst	Data Analysis	USD	100000	100000	United States	Entry-level	Contract	

9355 rows × 12 columns

2. By filtering data by the employee's residence, I was able to find out that there 8086 records for United States:

```
In [47]: jobs_in_data['employee_residence'] == "United States"
```

```
Out[47]: 0       False
         1        True
         2        True
         3        True
         4        True
                 ...
         9350     True
         9351     True
         9352     True
         9353     True
         9354     True
         Name: employee_residence, Length: 9355, dtype: bool
```

```
In [48]: jobs_usa = jobs_in_data.loc[jobs_in_data['employee_residence'] == "United States"]
         jobs_usa
```

Out[48]:

	work_year	job_title	job_category	salary_currency	salary	salary_in_usd	employee_residence	experience_level	employment_type	work_
1	2023	Data Architect	Data Architecture and Modeling	USD	186000	186000	United States	Senior	Full-time	In
2	2023	Data Architect	Data Architecture and Modeling	USD	81800	81800	United States	Senior	Full-time	In
3	2023	Data Scientist	Data Science and Research	USD	212000	212000	United States	Senior	Full-time	In
4	2023	Data Scientist	Data Science and Research	USD	93300	93300	United States	Senior	Full-time	In
5	2023	Data Scientist	Data Science and Research	USD	130000	130000	United States	Senior	Full-time	
...	...	...	...	...	...	...	...	...	...	
9350	2021	Data Specialist	Data Management and Strategy	USD	165000	165000	United States	Senior	Full-time	
9351	2020	Data Scientist	Data Science and Research	USD	412000	412000	United States	Senior	Full-time	
9352	2021	Principal Data Scientist	Data Science and Research	USD	151000	151000	United States	Mid-level	Full-time	
9353	2020	Data Scientist	Data Science and Research	USD	105000	105000	United States	Entry-level	Full-time	
9354	2020	Business Data Analyst	Data Analysis	USD	100000	100000	United States	Entry-level	Contract	

8086 rows × 12 columns

3. To sort residents in the United States by the company location, it could also be seen the following first 10 rows of the DataFrame:

```
In [62]: sorted_jobs_usa = jobs_usa.sort_values('company_location')
         sorted_jobs_usa.iloc[0:10]
```

Out[62]:

	salary_currency	salary	salary_in_usd	employee_residence	experience_level	employment_type	work_setting	company_location	company_size
	USD	171000	171000	United States	Senior	Full-time	Remote	Australia	L
	USD	90000	90000	United States	Entry-level	Full-time	Hybrid	Canada	L
	USD	225000	225000	United States	Senior	Full-time	Remote	Canada	L
	USD	152000	152000	United States	Senior	Full-time	Remote	France	L
	USD	50000	50000	United States	Entry-level	Full-time	Hybrid	Germany	M
	USD	160000	160000	United States	Senior	Full-time	Remote	Japan	L
	USD	221000	221000	United States	Senior	Full-time	In-person	United States	M
	USD	147000	147000	United States	Senior	Full-time	In-person	United States	M
	USD	100000	100000	United States	Senior	Full-time	In-person	United States	M
	USD	80000	80000	United States	Senior	Full-time	In-person	United States	M

4. I was able to verify there were no missing values in the DataFrame using the ISNULL function. In this case the output was False since there were no missing values (note: If the output was True, then that meant there were missing values in a specific column or row:

```
In [70]: jobs_in_data.isnull().any(axis=0)
```

```
Out[70]: work_year             False
         job_title              False
         job_category           False
         salary_currency        False
         salary                 False
         salary_in_usd          False
         employee_residence     False
         experience_level       False
         employment_type        False
         work_setting           False
         company_location       False
         company_size           False
         dtype: bool
```

```
In [69]: jobs_in_data.isnull().any(axis=1)
```

```
Out[69]: 0       False
         1       False
         2       False
         3       False
         4       False
                 ...
         9350    False
         9351    False
         9352    False
         9353    False
         9354    False
         Length: 9355, dtype: bool
```

5. To remove unnecessary columns I didn't want to see, I used the DROP function:

```
In [73]: cleaned_jobs = jobs_in_data.drop(columns = ['job_category', 'salary_currency', 'salary', 'experience_level',
                                                    'employment_type', 'work_setting'])
         cleaned_jobs
```

Out[73]:

	work_year	job_title	salary_in_usd	employee_residence	company_location	company_size
0	2023	Data DevOps Engineer	95012	Germany	Germany	L
1	2023	Data Architect	186000	United States	United States	M
2	2023	Data Architect	81800	United States	United States	M
3	2023	Data Scientist	212000	United States	United States	M
4	2023	Data Scientist	93300	United States	United States	M
...	...	...	...	...	...	...
9350	2021	Data Specialist	165000	United States	United States	L
9351	2020	Data Scientist	412000	United States	United States	L
9352	2021	Principal Data Scientist	151000	United States	United States	L
9353	2020	Data Scientist	105000	United States	United States	S
9354	2020	Business Data Analyst	100000	United States	United States	L

9355 rows × 6 columns

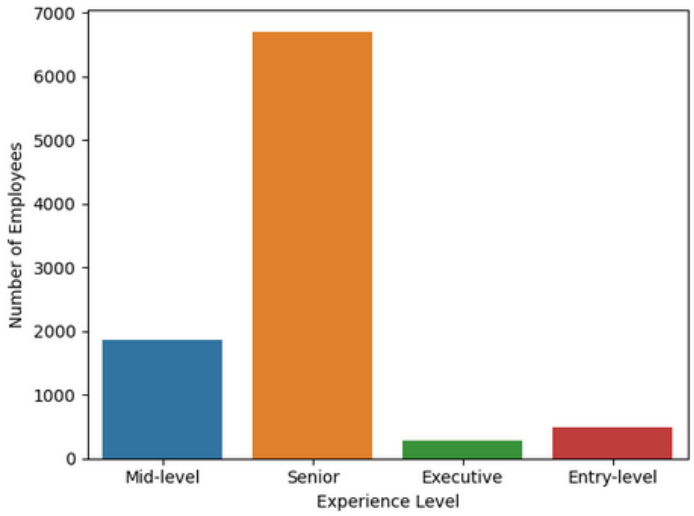
6. To check for any missing misspellings in the job title I used a method called UNIQUE, the result being:

```
In [74]: cleaned_jobs['job_title'].unique()
```

```
Out[74]: array(['Data DevOps Engineer', 'Data Architect', 'Data Scientist',
       'Machine Learning Researcher', 'Data Engineer',
       'Machine Learning Engineer', 'Data Analyst', 'Analytics Engineer',
       'Applied Scientist', 'BI Developer',
       'Business Intelligence Engineer', 'Research Scientist',
       'Research Analyst', 'Research Engineer', 'Data Science Engineer',
       'Data Quality Analyst', 'Data Product Manager',
       'Machine Learning Scientist', 'AI Engineer', 'MLOps Engineer',
       'Deep Learning Engineer', 'Data Modeler', 'Data Product Owner',
       'Data Science Consultant', 'Business Intelligence Analyst',
       'AI Developer', 'Data Manager', 'ML Engineer',
       'Data Science Director', 'Head of Data', 'BI Analyst',
       'Data Management Analyst', 'Machine Learning Modeler',
       'Data Specialist', 'BI Data Analyst', 'Data Integration Engineer',
       'Business Intelligence Manager', 'Data Integration Specialist',
       'Data Science Practitioner', 'Business Intelligence Developer',
       'AI Research Engineer', 'Data Lead', 'Data Management Specialist',
       'AI Architect', 'Data Science Manager', 'Data Strategist',
       'Business Intelligence Specialist',
       'Machine Learning Infrastructure Engineer',
       'Data Quality Engineer', 'Director of Data Science',
       'Business Data Analyst', 'Decision Scientist',
       'Financial Data Analyst', 'Data Strategy Manager',
       'Computer Vision Engineer', 'Data Visualization Specialist',
       'Insight Analyst', 'Data Visualization Engineer', 'ETL Developer',
       'Data Analytics Manager', 'Azure Data Engineer', 'Data Developer',
       'Principal Data Scientist', 'Data Science Lead',
       'Staff Data Analyst', 'Data Infrastructure Engineer',
       'Machine Learning Software Engineer',
       'Machine Learning Operations Engineer', 'AI Scientist',
       'Head of Machine Learning', 'Applied Data Scientist',
       'AI Programmer', 'Data Operations Analyst',
       'Applied Machine Learning Scientist', 'Data Analytics Lead',
       'Data Operations Engineer', 'Machine Learning Manager',
       'Lead Data Scientist', 'Principal Machine Learning Engineer',
       'Principal Data Engineer', 'Power BI Developer',
       'Head of Data Science', 'Staff Machine Learning Engineer',
       'Staff Data Scientist', 'Consultant Data Engineer',
       'Machine Learning Specialist',
       'Business Intelligence Data Analyst', 'Data Operations Manager',
       'Lead Machine Learning Engineer', 'Managing Director Data Science',
       'Data Modeller', 'Finance Data Analyst', 'Software Data Engineer',
```

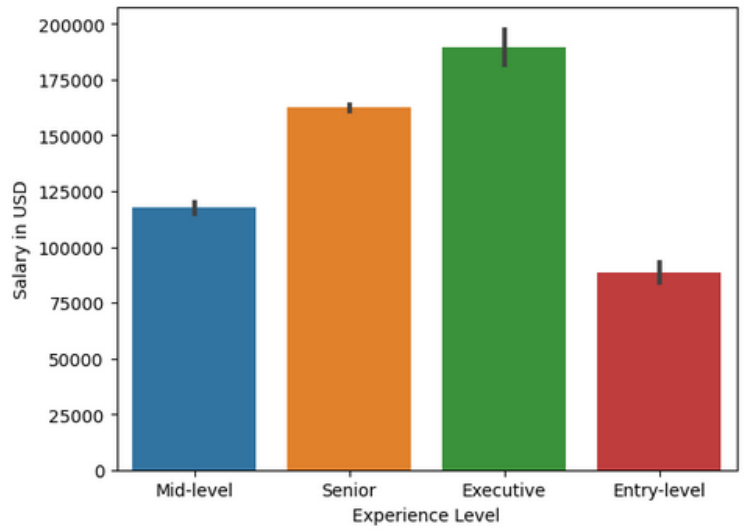
7. To visualize qualitative data such as the number of employees by experience level I used the count plot in seaborn. This way I was able to see the number of employees being Senior as the largest, and the number of employees being Executive as the smallest:

In [38]:
```
sn.countplot(x = 'experience_level', data = jobs_in_data)
plt.xlabel('Experience Level')
plt.ylabel('Number of Employees')
plt.show()
```



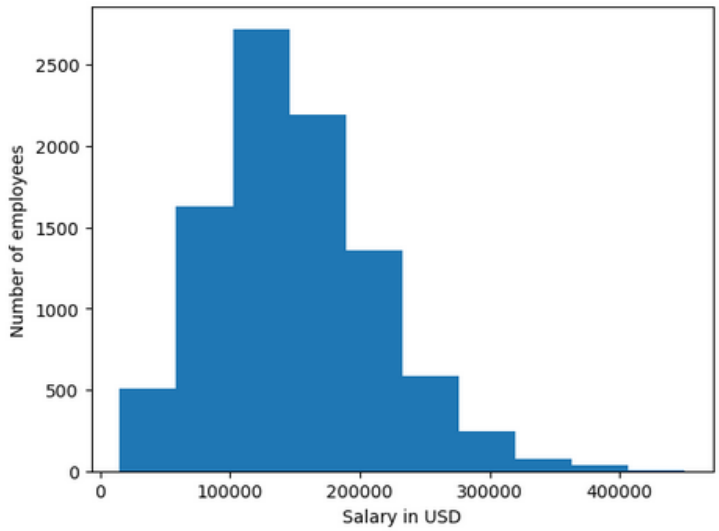
8. Likewise, by visualizing the salary of employees in USD by experience level I was able to find out the salary of executives as the highest, while the salary for entry-level as the smallest in terms of USD :

In [39]:
```
sn.barplot(x = 'experience_level', y = 'salary_in_usd', data = jobs_in_data)
plt.xlabel('Experience Level')
plt.ylabel('Salary in USD')
plt.show()
```



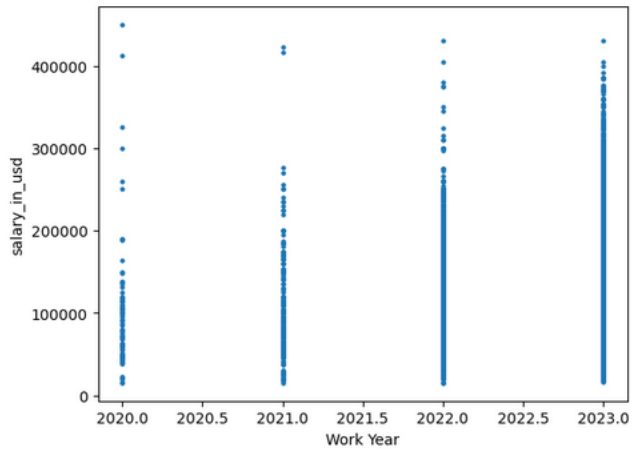
9. As for quantitative data, I used a histogram to visualize the distribution of the US salary by the number of employees. Based on the histogram it could be seen that a lot of the salaries lie between 50K and 300K:

```
In [40]: plt.hist(jobs_in_data['salary_in_usd'])
         plt.xlabel('Salary in USD')
         plt.ylabel('Number of employees')
         plt.show()
```



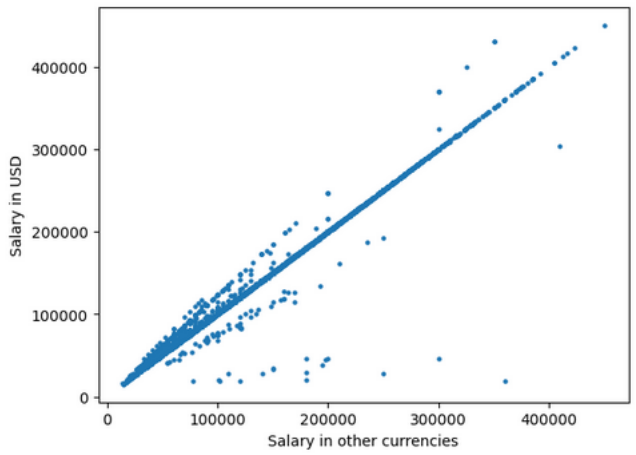
10. I have also used a scatter plot To compare the salaries in USD and the number of employees for each year. This way I was able to find out that as years passed by, there were more people whose salaries increased:

```
In [33]: plt.scatter(x = jobs_in_data['work_year'], y = jobs_in_data['salary_in_usd'], s = 5)
         plt.xlabel('Work Year')
         plt.ylabel('salary_in_usd')
         plt.show()
```



11. Finally, I have also used a scatter plot to compare the salary in USD to salaries in other currencies. There seems to be a significant correlation between the two:

```
In [34]: plt.scatter(x = jobs_in_data['salary'], y = jobs_in_data['salary_in_usd'], s = 5)
         plt.xlabel('Salary in other currencies')
         plt.ylabel('Salary in USD')
         plt.show()
```



Conclusion: By performing an exploratory data analysis (DEA) I was able to find out:

- The number of employees in the country of United States,
- If there were any missing values or misspellings in the dataset through a data cleaning process,
- The number of employees and their salary by experience level,
- A range for the number of employees who make between 50K and 300K,
- An increase of the number of employees and their salaries as years passed by,
- A correlation between salary in USD and other currencies.

References:

- https://www.kaggle.com/datasets/hummaamqaasim/jobs-in-data)