

```
In [10]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [11]: 1.#load in data
df = pd.read_excel(r"C:\Users\User\Downloads\salaries 2023.xlsx")
df
```

Out[11]:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	com
	0	2023	SE	Principal Data Scientist	80000	EUR	85847	ES	100	
	1	2023	MI	CT ML Engineer	30000	USD	30000	US	100	
	2	2023	MI	CT ML Engineer	25500	USD	25500	US	100	
	3	2023	SE	FT Data Scientist	175000	USD	175000	CA	100	
	4	2023	SE	FT Data Scientist	120000	USD	120000	CA	100	
	
	3750	2020	SE	FT Data Scientist	412000	USD	412000	US	100	
	3751	2021	MI	FT Principal Data Scientist	151000	USD	151000	US	100	
	3752	2020	EN	FT Data Scientist	105000	USD	105000	US	100	
	3753	2020	EN	CT Business Data Analyst	100000	USD	100000	US	100	
	3754	2021	SE	FT Data Science Manager	7000000	INR	94665	IN	50	

3755 rows × 11 columns

```
In [ ]: 
```

```
In [6]: #understand the data
df.shape
```

Out[6]: (3755, 11)

```
In [7]: df['work_year'].value_counts() # count of employess by work year
```

Out[7]:

2023	1785
2022	1664
2021	230
2020	76

Name: work_year, dtype: int64

```
In [38]: df['experience_level'].value_counts() #count of employess by experience level
#EN, which refers to Entry-level / Junior.

#MI, which refers to Mid-level / Intermediate.

#SE, which refers to Senior-level / Expert.

#EX, which refers to Executive-level / Director.
```

Out[38]:

SE	1554
MI	664
EN	270
EX	96

Name: experience_level, dtype: int64

```
In [11]: df['employment_type'].value_counts() # count of employess by employment type
# employment_type :
```

Out[11]:

FT	3718
PT	17
CT	10
FL	10

Name: employment_type, dtype: int64

```
In [13]: df['job_title'].value_counts() # count employess by job title
```

Out[13]:

Data Engineer	1040
Data Scientist	840
Data Analyst	612
Machine Learning Engineer	289
Analytics Engineer	103
...	...
Principal Machine Learning Engineer	1
Azure Data Engineer	1
Manager Data Management	1
Marketing Data Engineer	1
Finance Data Analyst	1

Name: job_title, Length: 93, dtype: int64

```
In [14]: df['salary'].value_counts() # count employess by their salary
```

Out[14]:

100000	112
150000	100
120000	99
160000	85
130000	85
...	...
241871	1
93919	1
385000	1
225900	1
412000	1

Name: salary, Length: 815, dtype: int64

```
In [15]: df['employee_residence'].value_counts() # count employess by their residence
```

Out[15]:

US	3004
GB	167
CA	85
ES	80
IN	71
...	...
BA	1
AM	1
CV	1
RW	1
MT	1

Name: employee_residence, Length: 78, dtype: int64

```
In [16]: df['company_location'].value_counts() # count employess by company location
```

Out[16]:

US	3040
GB	172
CA	87
ES	77
IN	58
...	...
MK	1
BS	1
IR	1
CR	1
MT	1

Name: company_location, Length: 72, dtype: int64

```
In [17]: df['company_size'].value_counts() #count employess by company size
```

Out[17]:

M	3153
L	454
S	148

Name: company_size, dtype: int64

```
In [ ]: #Data Preprocessing
```

```
In [20]: df.isnull().sum() # check null values
```

Out[20]:

work_year	0
experience_level	0
employment_type	0
job_title	0
salary	0
salary_currency	0
salary_in_usd	0
employee_residence	0
remote_ratio	0
company_location	0
company_size	0
dtype:	int64

```
In [22]: df.duplicated().sum() # check duplicate values
```

Out[22]: 0

```
In [23]: df['salary_in_usd'].groupby(df['job_title']).mean() # salary mean based on job title
```

Out[23]:

job_title	
3D Computer Vision Researcher	21352.250000
AI Developer	136666.090909
AI Programmer	55000.000000
AI Scientist	110120.875000
Analytics Engineer	150151.747253
...	...
Research Engineer	165909.393939
Research Scientist	153486.369231
Software Data Engineer	62510.000000
Staff Data Analyst	15000.000000
Staff Data Scientist	105000.000000

Name: salary_in_usd, Length: 93, dtype: float64

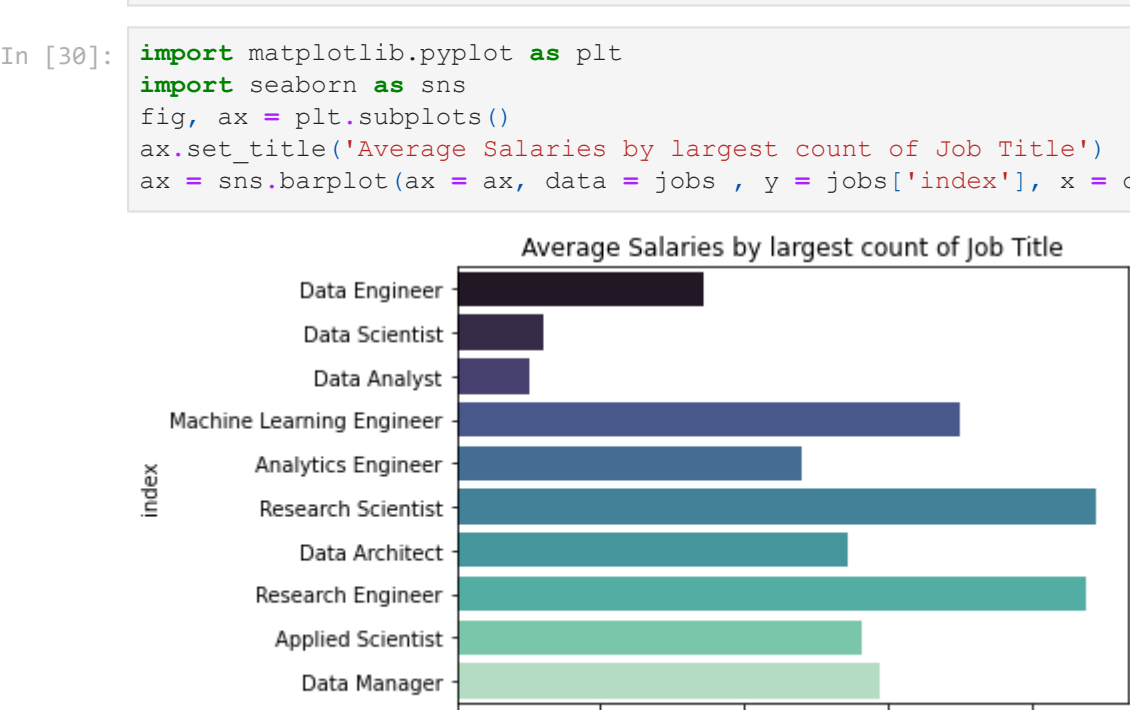
```
In [26]: jobs = df[df['work_year']==2023]['job_title'].value_counts().nlargest(10).reset_index() # 10 largest job title
jobs
```

Out[26]:

	index	job_title	
0		Data Engineer	269
1		Data Scientist	223
2		Data Analyst	195
3		Machine Learning Engineer	103
4		Analytics Engineer	41
5		Research Scientist	38
6		Data Architect	31
7		Research Engineer	28
8		Applied Scientist	20
9		Data Manager	18

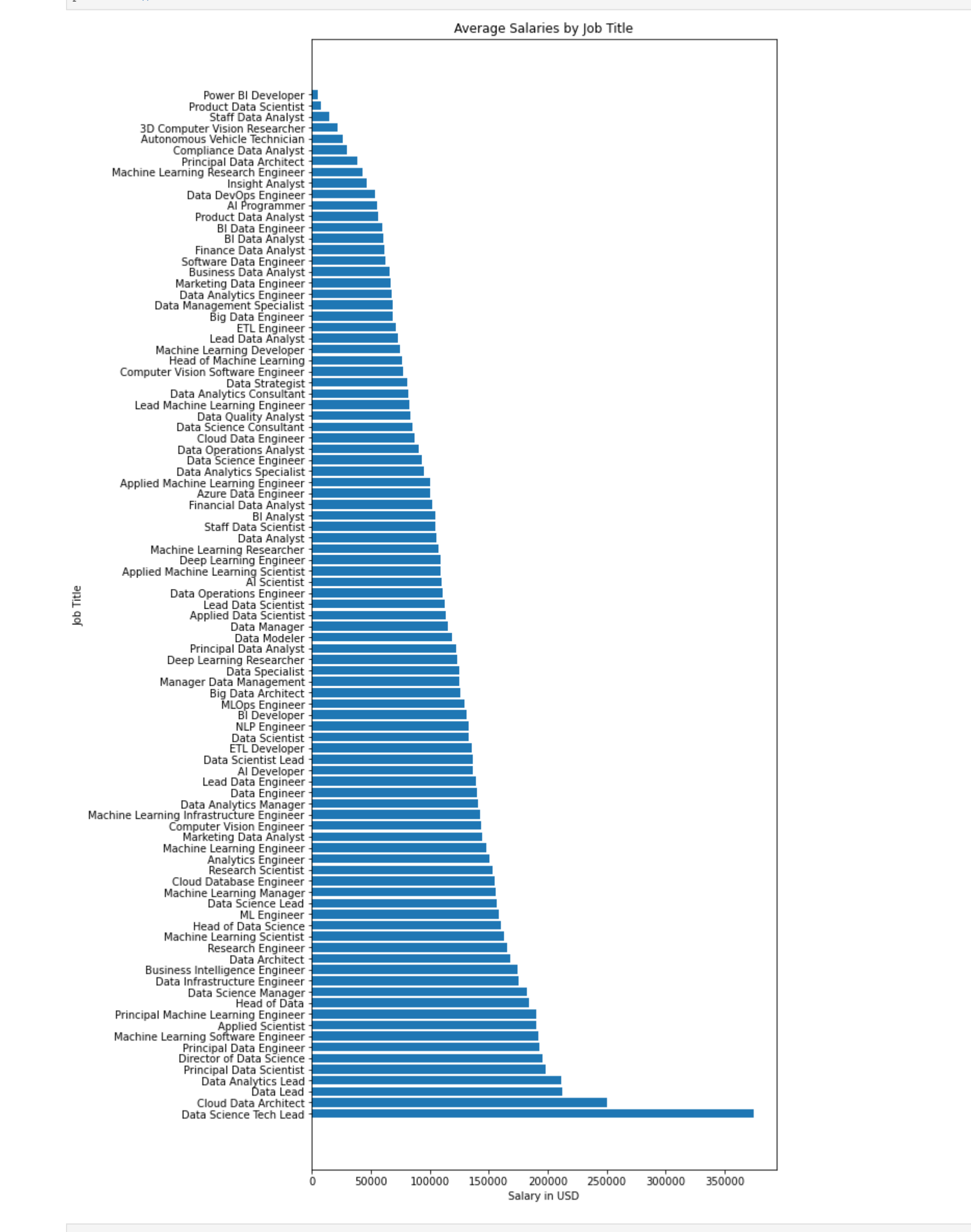
```
In [ ]: # Data engineer,Data Scientist and Data Analyst are top 3 job in industry
```

```
In [30]: import matplotlib.pyplot as plt
import seaborn as sns
fig, ax = plt.subplots()
ax.set_title('Average Salaries by largest count of Job Title')
ax = sns.barplot(ax = ax, data = jobs , y = jobs['index'], x = df['salary_in_usd'],palette='mako')
```



```
In [12]: # Calculate mean salaries for each job title
job_title_salaries = df.groupby('job_title')['salary_in_usd'].mean().sort_values(ascending=False)
```

```
In [33]: # Create horizontal bar chart
import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize=(8, 20))
ax.barh(job_title_salaries.index, job_title_salaries.values)
ax.set_title('Average Salaries by Job Title')
ax.set_xlabel('Salary in USD')
ax.set_ylabel('Job Title')
plt.show()
```



```
In [ ]: # Data science tech lead is having the highest salary followed by cloud Data Architect, whereas Power bi develop
#If compared the count of job title to its salary the top 3 most popular jobs (Data engineer,Data Scientist and
```

```
In [37]: df['salary_in_usd'].groupby(df['experience_level']).mean() #Average Salary based on work experience
```

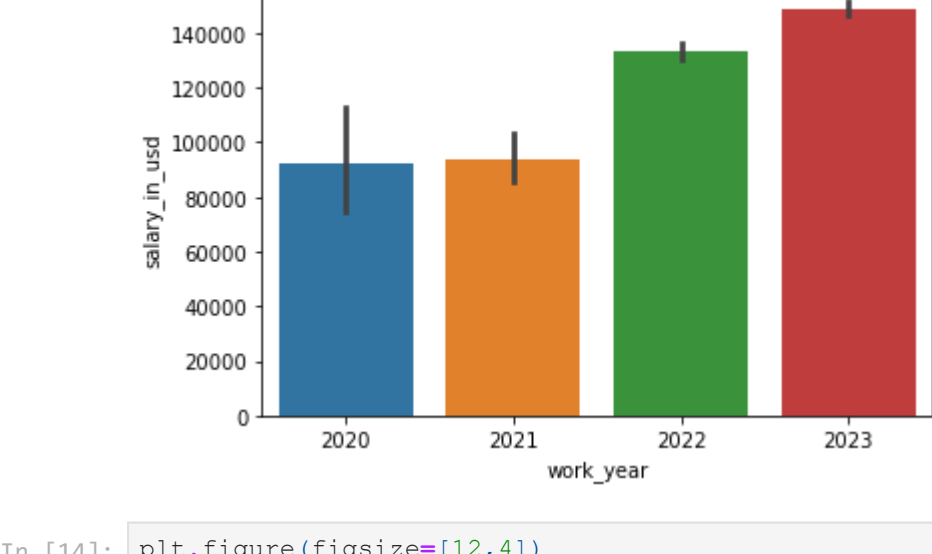
Out[37]:

experience_level	
EN	72648.685185
EX	131078.208333
MI	101828.783133
SE	153897.435650

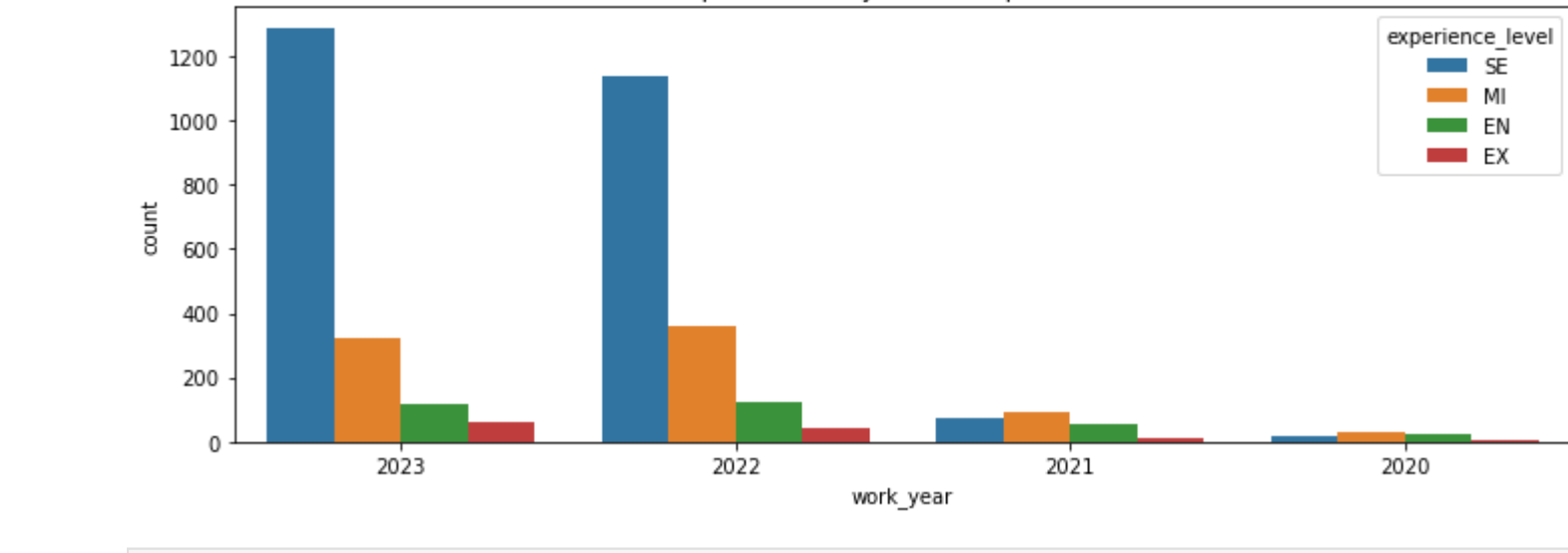
Name: salary_in_usd, dtype: float64

```
In [13]: import numpy as np
import seaborn as sns
#Average Salary based on work experience
comapanv_salary=np.array(df['salary_in_usd'].groupby(df['work_year']).mean())
sns.barplot(x='work_year', y='salary_in_usd', data=df)
```

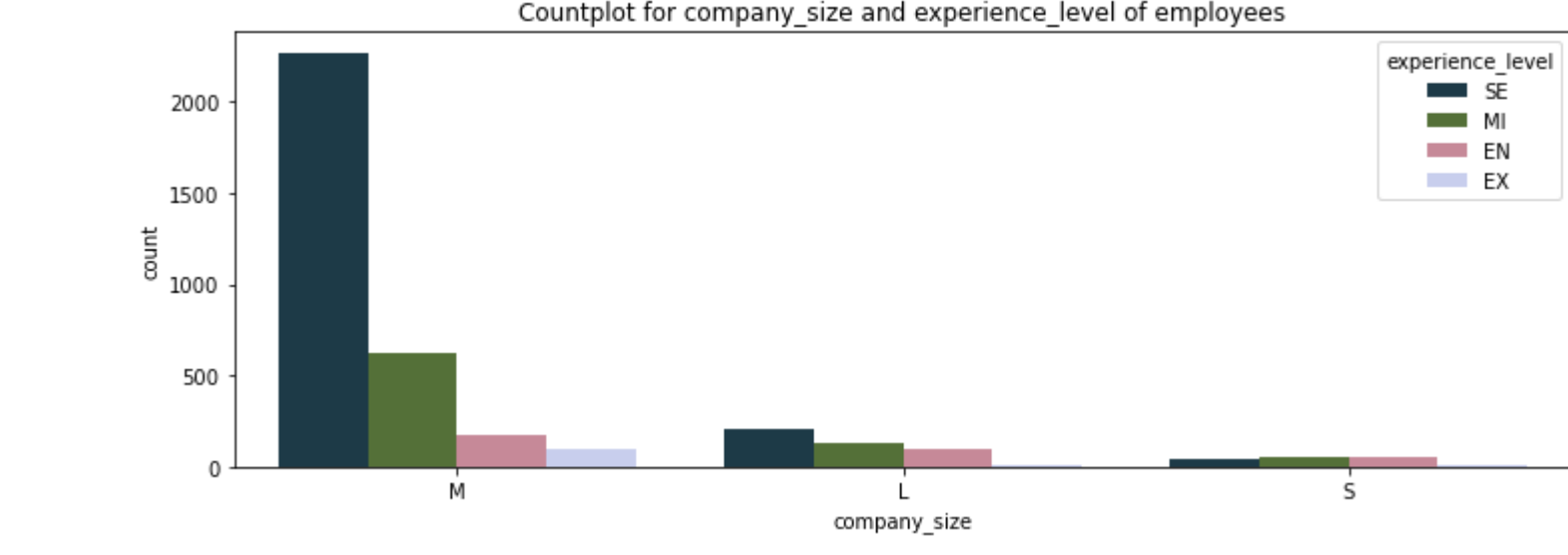
Out[13]: <AxesSubplot: xlabel='work_year', ylabel='salary_in_usd'>



```
In [14]: plt.figure(figsize=(12,4))
sns.countplot(x = df['work_year'],hue=df['experience_level'],order=df['work_year'].value_counts().index)
plt.title('Countplot for work year and experience level')
plt.show()
```



```
In [17]: plt.figure(figsize=(12,4))
sns.countplot(x = df['company_size'],hue=df['experience_level'],order=df['company_size'].value_counts().index,p
plt.title('Countplot for company_size and experience_level of employees')
plt.show()
```



```
In [56]: #Salaries based on remote ratio
compsize_based_salary=np.array(df['salary_in_usd'].groupby(df['remote_ratio']).mean())
#compsize_based_title=['Large', 'Medium', 'Small']
plt.title('Average Salaries based on Company Size')
plt.xlabel('Company size')
plt.ylabel('Salary')
sns.barplot(x='remote_ratio',y='salary_in_usd',data=df)
plt.show()
```

