In [ ]:
```
 pip install matplotlib
```

In [ ]:
```
pip install seaborn
```

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
df=pd.read_csv("salaries_dataset.csv")
df
```

Out[2]:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	emplo
0	2024	SE	FT	AI Engineer	90000	USD	90000	
1	2024	SE	FT	Machine Learning Engineer	180500	USD	180500	
2	2024	SE	FT	Machine Learning Engineer	96200	USD	96200	
3	2024	SE	FT	Machine Learning Engineer	235000	USD	235000	
4	2024	SE	FT	Machine Learning Engineer	175000	USD	175000	
...	...	...	...	...	...	...	...	
13967	2020	SE	FT	Data Scientist	412000	USD	412000	
13968	2021	MI	FT	Principal Data Scientist	151000	USD	151000	
13969	2020	EN	FT	Data Scientist	105000	USD	105000	
13970	2020	EN	CT	Business Data Analyst	100000	USD	100000	
13971	2021	SE	FT	Data Science Manager	7000000	INR	94665	

13972 rows × 11 columns

In [3]:
```
df.head()
```

Out[3]:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_r
0	2024	SE	FT	AI Engineer	90000	USD	90000	
1	2024	SE	FT	Machine Learning Engineer	180500	USD	180500	
2	2024	SE	FT	Machine Learning Engineer	96200	USD	96200	
3	2024	SE	FT	Machine Learning Engineer	235000	USD	235000	
4	2024	SE	FT	Machine Learning Engineer	175000	USD	175000	

In [4]:
```
df.tail()
```

Out[4]:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	emplo
13967	2020	SE	FT	Data Scientist	412000	USD	412000	
13968	2021	MI	FT	Principal Data Scientist	151000	USD	151000	
13969	2020	EN	FT	Data Scientist	105000	USD	105000	
13970	2020	EN	CT	Business Data Analyst	100000	USD	100000	
13971	2021	SE	FT	Data Science Manager	7000000	INR	94665	

In [6]:
```
df.sample(5)
```

Out[6]:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employ
6061	2023	SE	FT	Data Engineer	190000	USD	190000	
7787	2023	SE	FT	Machine Learning Engineer	176000	USD	176000	
8221	2023	SE	FT	Analytics Engineer	170000	USD	170000	
2222	2024	SE	FT	Machine Learning Scientist	242000	CAD	186153	
10315	2023	SE	FT	Data Scientist	262000	USD	262000	

In [5]:
```
df.shape
```

Out[5]:
```
(13972, 11)
```

In [7]:
```
df.columns
```

Out[7]:
```
Index(['work_year', 'experience_level', 'employment_type', 'job_title',
       'salary', 'salary_currency', 'salary_in_usd', 'employee_residence',
       'remote_ratio', 'company_location', 'company_size'],
      dtype='object')
```

In [8]:
```
df.dtypes
```

Out[8]:
```
work_year              int64
experience_level      object
employment_type       object
job_title             object
salary                 int64
salary_currency       object
salary_in_usd          int64
employee_residence    object
remote_ratio           int64
company_location      object
company_size          object
dtype: object
```

In [9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13972 entries, 0 to 13971
Data columns (total 11 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   work_year           13972 non-null  int64 
 1   experience_level    13972 non-null  object
 2   employment_type     13972 non-null  object
 3   job_title           13972 non-null  object
 4   salary              13972 non-null  int64 
 5   salary_currency     13972 non-null  object
 6   salary_in_usd       13972 non-null  int64 
 7   employee_residence  13972 non-null  object
 8   remote_ratio        13972 non-null  int64 
 9   company_location    13972 non-null  object
 10  company_size        13972 non-null  object
dtypes: int64(4), object(7)
memory usage: 1.2+ MB
```

In [10]:
```
df.isnull()
```

Out[10]:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employe
0	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	
13967	False	False	False	False	False	False	False	
13968	False	False	False	False	False	False	False	
13969	False	False	False	False	False	False	False	
13970	False	False	False	False	False	False	False	
13971	False	False	False	False	False	False	False	

13972 rows × 11 columns

In [11]:
```
df.isnull().sum()
```

Out[11]:
```
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 [12]:
```
df.describe()
```

Out[12]:

	work_year	salary	salary_in_usd	remote_ratio
count	13972.000000	1.397200e+04	13972.000000	13972.000000
mean	2023.085600	1.660011e+05	150028.812124	33.334526
std	0.687612	3.661545e+05	68634.306349	46.682931
min	2020.000000	1.400000e+04	15000.000000	0.000000
25%	2023.000000	1.040000e+05	103000.000000	0.000000
50%	2023.000000	1.422000e+05	141600.000000	0.000000
75%	2024.000000	1.880000e+05	185900.000000	100.000000
max	2024.000000	3.040000e+07	800000.000000	100.000000

In [45]:
```
df.describe(exclude="number")
```

Out[45]:

	experience_level	employment_type	job_title	salary_currency	employee_residence	company_location	c
count	8594	8594	8594	8594	8594	8594	
unique	4	4	146	23	88	77	
top	SE	FT	Data Engineer	USD	US	US	
freq	5168	8534	1635	7667	7026	7075	

In [13]:
```
df.duplicated().sum()
```

Out[13]:
```
5378
```

In [135…
```
df.drop_duplicates(inplace=True)
```

In [47]:
```
df.isna().sum()
```

Out[47]:
```
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 [16]:
```
print(df.work_year.min()," ", df.work_year.max())
```

```
2020   2024
```

In [17]:
```
df.employment_type.unique()
```

Out[17]:
```
array(['FT', 'CT', 'PT', 'FL'], dtype=object)
```

In [18]:
```
df.experience_level.unique()
```

Out[18]:
```
array(['SE', 'MI', 'EN', 'EX'], dtype=object)
```

In [19]:
```
df.job_title.unique()
```

Out[19]:
```
array(['AI Engineer', 'Machine Learning Engineer',
       'Business Intelligence Developer', 'Data Engineer',
       'Data Scientist', 'Cloud Database Engineer', 'Research Engineer',
       'Data Analyst', 'Machine Learning Scientist', 'Applied Scientist',
       'Data Science Manager', 'Research Scientist', 'Prompt Engineer',
       'Data Science', 'Data Science Consultant',
       'Data Management Analyst', 'Research Analyst',
       'Data Operations Analyst', 'Data Management Consultant',
       'Business Intelligence Analyst', 'Analytics Engineer',
       'Data Quality Analyst', 'Data Architect', 'Data Manager',
       'ML Engineer', 'Robotics Software Engineer',
       'Machine Learning Researcher', 'AI Architect',
       'Data DevOps Engineer', 'Business Intelligence',
       'AI Software Engineer', 'Data Integration Engineer',
       'Data Operations Specialist', 'BI Analyst', 'Data Product Manager',
       'Business Intelligence Engineer', 'Data Specialist',
       'AI Research Scientist', 'Data Science Director',
       'Data Strategist', 'Big Data Developer', 'BI Developer',
       'Quantitative Research Analyst', 'Lead Machine Learning Engineer',
       'Machine Learning Research Engineer',
       'Data Infrastructure Engineer', 'Data Analytics Lead',
       'Business Intelligence Manager', 'Data Analytics Manager',
       'Data Developer', 'Data Analytics Consultant',
       'AI Research Engineer', 'Data Analytics Specialist',
       'ETL Developer', 'Data Science Engineer', 'Big Data Engineer',
       'Data Modeler', 'Robotics Engineer', 'Business Intelligence Lead',
       'AI Programmer', 'ETL Engineer', 'Head of Data',
       'AI Product Manager', 'Data Management Specialist',
       'Data Operations Associate', 'AI Developer',
       'Admin & Data Analyst', 'AI Scientist',
       'Data Integration Specialist', 'Computer Vision Engineer',
       'Head of Machine Learning', 'Data Analyst Lead',
       'Machine Learning Operations Engineer', 'Data Lead',
       'Data Science Practitioner', 'MLOps Engineer',
       'Data Integration Developer', 'ML Ops Engineer',
       'Data Pipeline Engineer', 'Lead Data Analyst', 'Data Science Lead',
       'Director of Data Science', 'Managing Director Data Science',
       'Data Visualization Specialist', 'Data Quality Manager',
       'Data Product Owner', 'Machine Learning Infrastructure Engineer',
       'Business Data Analyst', 'NLP Engineer',
       'Marketing Data Scientist', 'Insight Analyst',
       'Deep Learning Engineer', 'Machine Learning Modeler',
       'BI Data Analyst', 'Business Intelligence Specialist',
       'Data Quality Engineer', 'Decision Scientist',
       'Financial Data Analyst', 'Data Strategy Manager',
       'Data Visualization Engineer', 'Azure Data Engineer',
       'Principal Data Scientist', 'Staff Data Analyst',
       'Machine Learning Software Engineer', 'Applied Data Scientist',
       'Applied Machine Learning Scientist', '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',
       'Data Modeller', 'Finance Data Analyst', 'Software Data Engineer',
       'Compliance Data Analyst', 'Cloud Data Engineer',
       'Analytics Engineering Manager', 'AWS Data Architect',
       'Product Data Analyst', 'Machine Learning Developer',
       'Data Visualization Analyst', 'Autonomous Vehicle Technician',
       'Sales Data Analyst', 'Applied Machine Learning Engineer',
       'BI Data Engineer', 'Deep Learning Researcher',
       'Big Data Architect', 'Computer Vision Software Engineer',
       'Marketing Data Engineer', 'Manager Data Management',
```

```
       'Data Science Tech Lead', 'Data Scientist Lead',
       'Marketing Data Analyst', 'Principal Data Architect',
       'Data Analytics Engineer', 'Cloud Data Architect',
       'Lead Data Engineer', 'Principal Data Analyst'], dtype=object)
```

In [20]:
```
df.salary_in_usd.min()
```

Out[20]:
```
15000
```

In [21]:
```
df.salary_in_usd.max()
```

Out[21]:
```
800000
```

In [76]:
```
df.salary_in_usd.mean()
```

Out[76]:
```
146745.2243425646
```

In [75]:
```
df.work_year.value_counts()
```

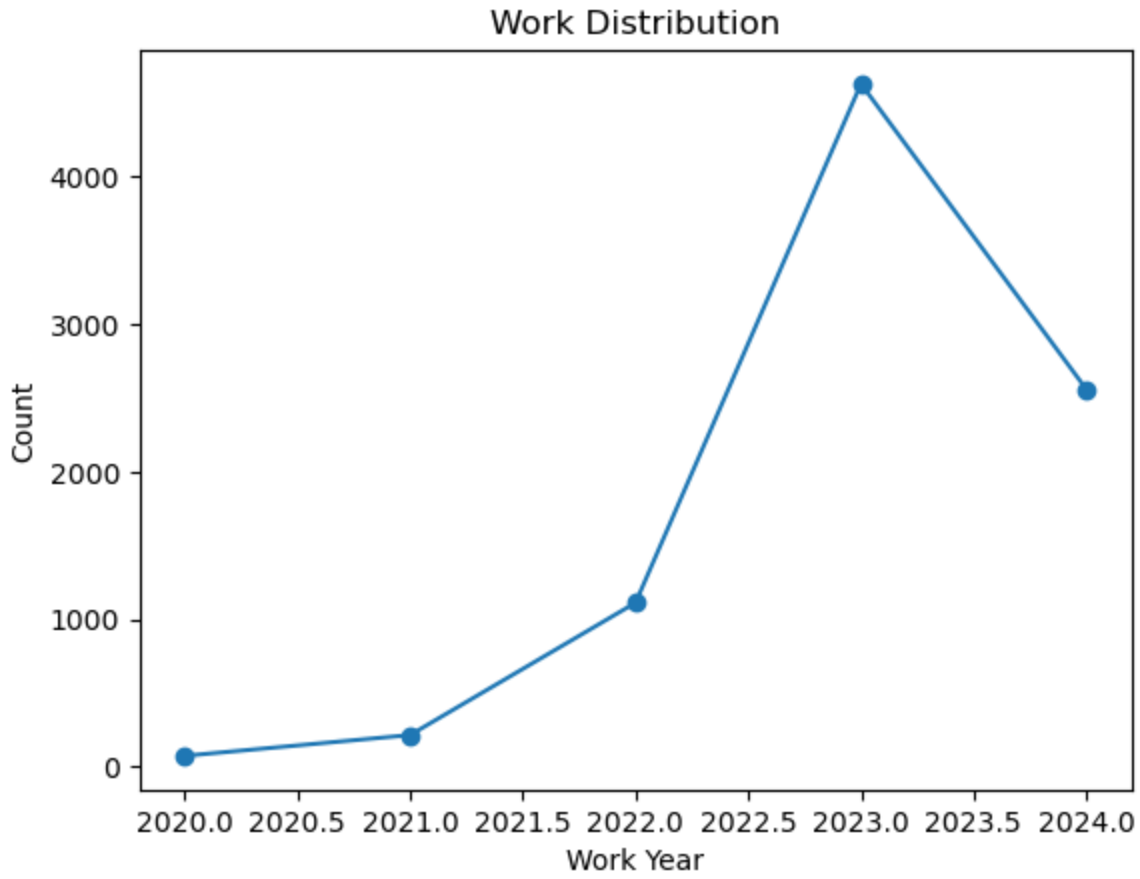
Out[75]:
```
work_year
2023    4631
2024    2559
2022    1113
2021     216
2020      75
Name: count, dtype: int64
```

In [77]:
```
g=df.work_year.value_counts().sort_index()
g
```

Out[77]:
```
work_year
2020      75
2021     216
2022    1113
2023    4631
2024    2559
Name: count, dtype: int64
```

In [119…
```
plt.plot(g.index,g.values,'o-')
plt.xlabel("Work Year")
plt.ylabel("Count")
plt.title("Work Distribution")
plt.plot()
```

Out[119]:
```
[]
```



In [31]:
```
df.job_title.value_counts()
```

Out[31]:
```
job_title
Data Engineer                       1635
Data Scientist                      1611
Data Analyst                        1185
Machine Learning Engineer            827
Analytics Engineer                   326
                                    ...
Analytics Engineering Manager          1
AWS Data Architect                     1
Sales Data Analyst                     1
Big Data Developer                     1
BI Data Engineer                       1
Name: count, Length: 146, dtype: int64
```

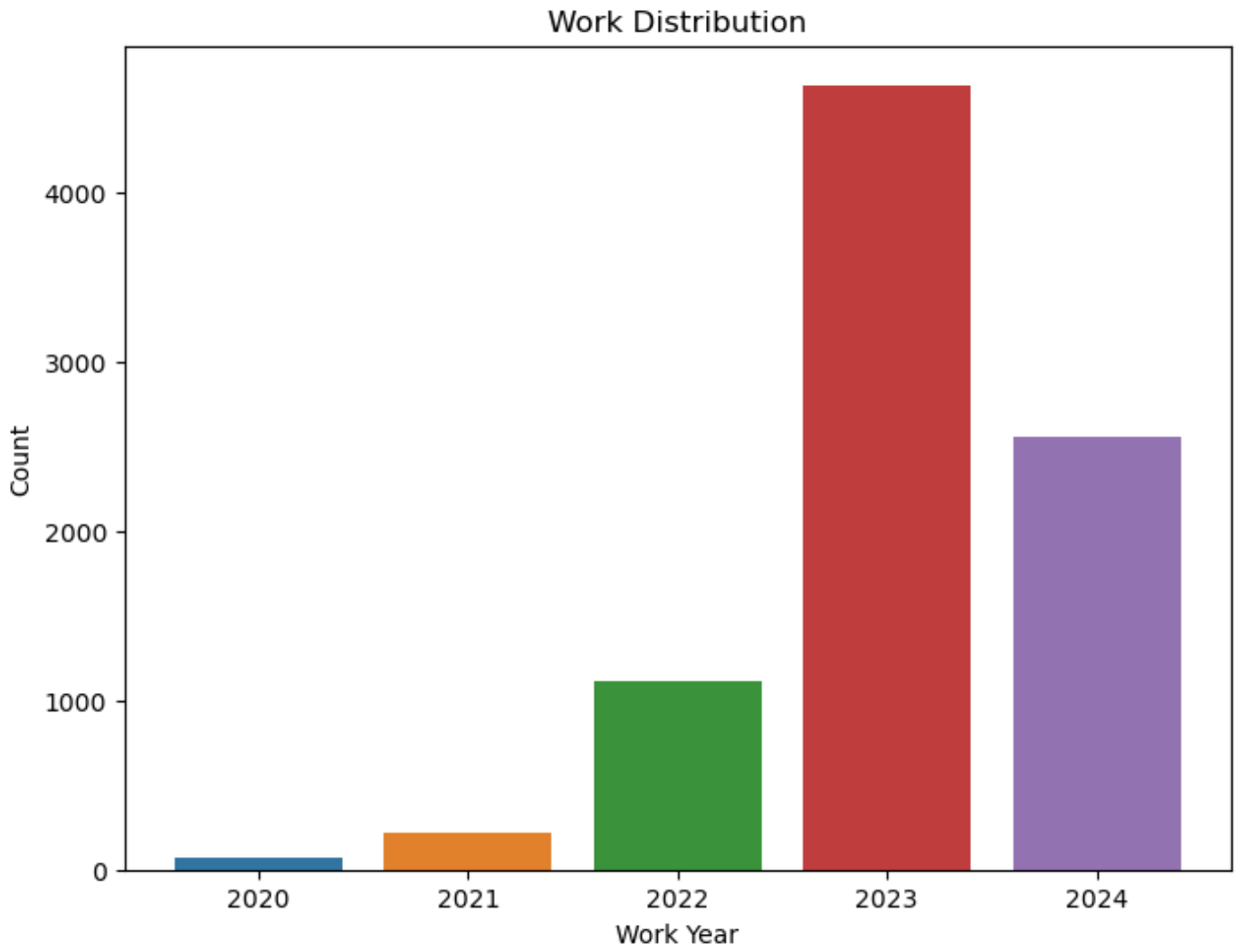
In [32]:
```
g= df.job_title.value_counts().head(5)
g
```

Out[32]:
```
job_title
Data Engineer                1635
Data Scientist               1611
Data Analyst                 1185
Machine Learning Engineer     827
Analytics Engineer            326
Name: count, dtype: int64
```

In [61]:
```
df['experience_level'].replace({'EN':'Entry-Level','MI':'Mid-Level','EX':'Executive Leve
df['employment_type'].replace({'PT':'Part-Time','FT':'Full-Time','CT':'Contract','FL':'F
```

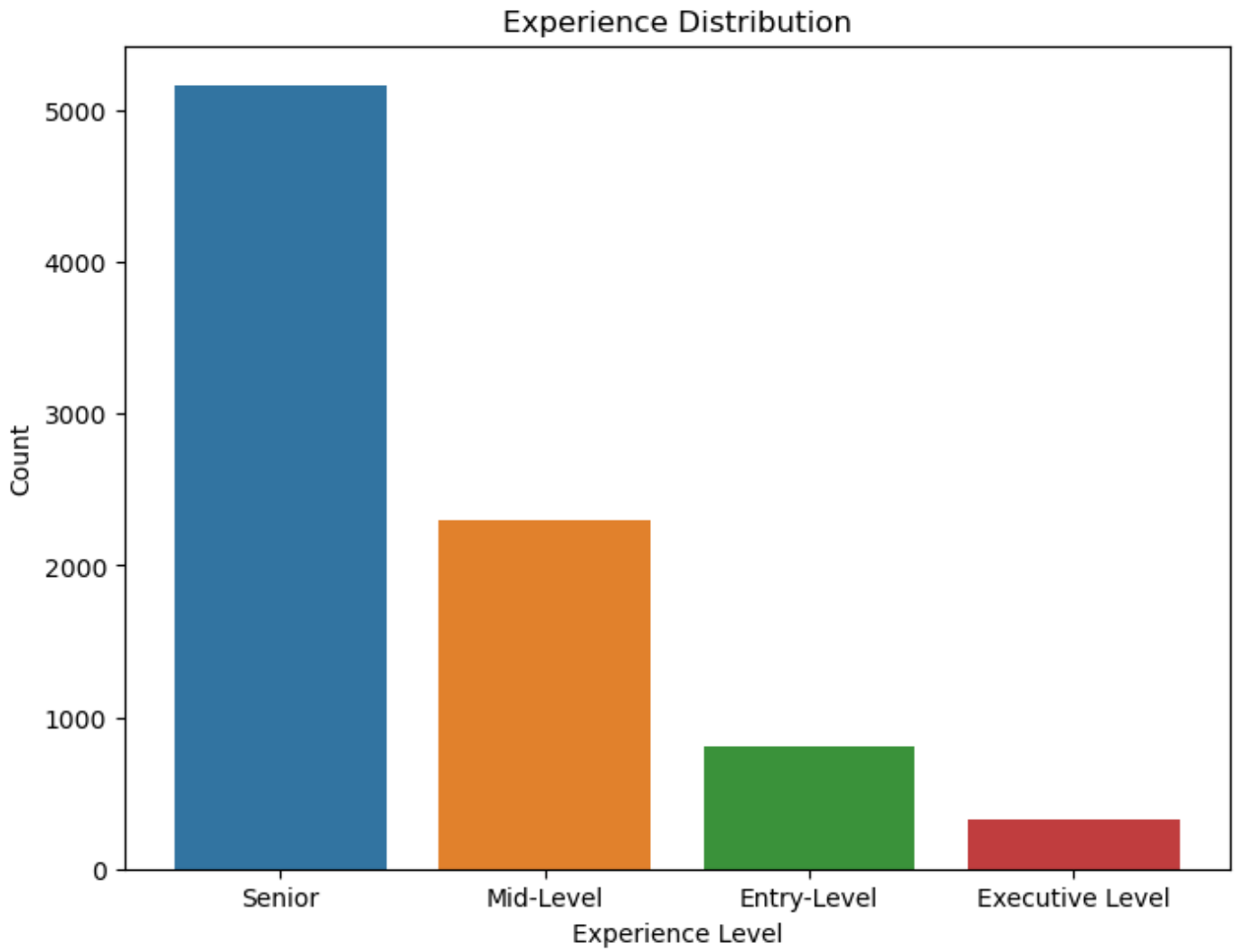
In [62]:
```
plt.figure(figsize = (8,6))
sns.countplot(data = df,x = 'work_year')
plt.xlabel("Work Year")
plt.ylabel("Count")
plt.title("Work Distribution")
```

Out[62]: []



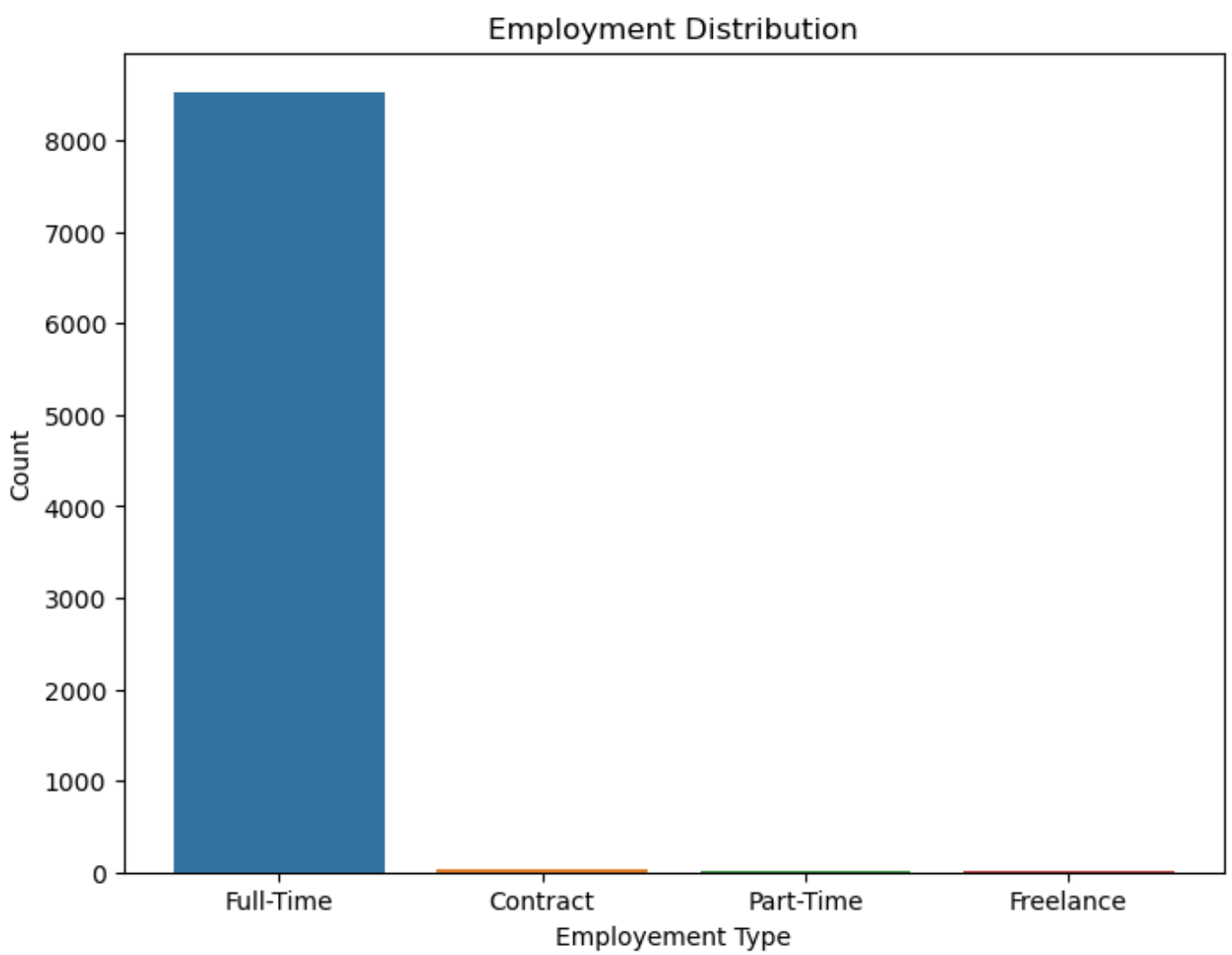
In [63]:
```
plt.figure(figsize = (8,6))
sns.countplot(data = df,x = 'experience_level')
plt.xlabel("Experience Level")
plt.ylabel("Count")
plt.title("Experience Distribution")
plt.plot()
```

Out[63]: []



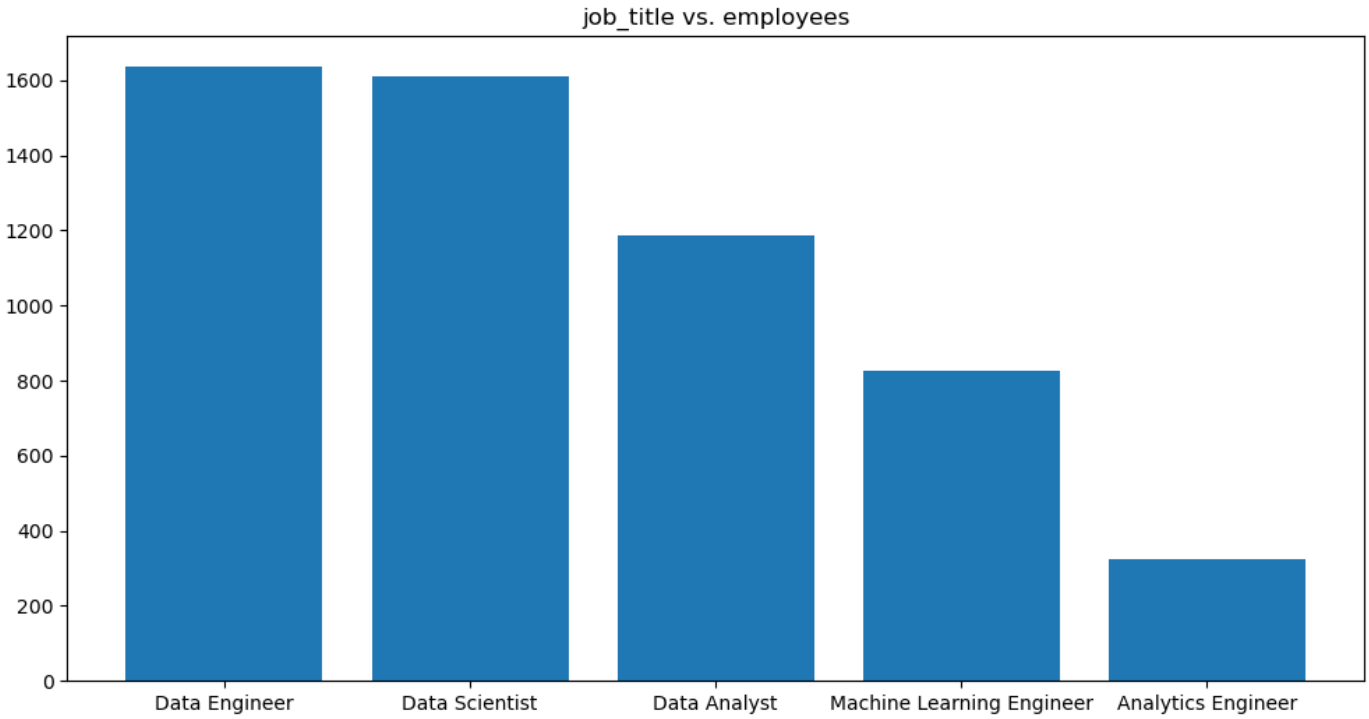
In [64]:
```
plt.figure(figsize = (8,6))
sns.countplot(data = df,x = 'employment_type')
plt.xlabel("Employement Type")
plt.ylabel("Count")
plt.title("Employment Distribution")
plt.plot()
```

Out[64]:
```
[]
```



In [71]:
```
plt.figure(figsize=(12,6), dpi=100)
plt.bar(g.index,g.values)
plt.title('job_title vs. employees')
```

Out[71]:
```
Text(0.5, 1.0, 'job_title vs. employees')
```



In [34]:
```
df.company_location.value_counts()
```

Out[34]:
```
company_location
US    7075
GB     481
CA     321
DE      95
ES      70
      ...
GI       1
EC       1
AD       1
MU       1
MD       1
Name: count, Length: 77, dtype: int64
```

In [35]:
```
df.company_size.value_counts()
```

Out[35]:
```
company_size
M    7805
L     612
S     177
Name: count, dtype: int64
```

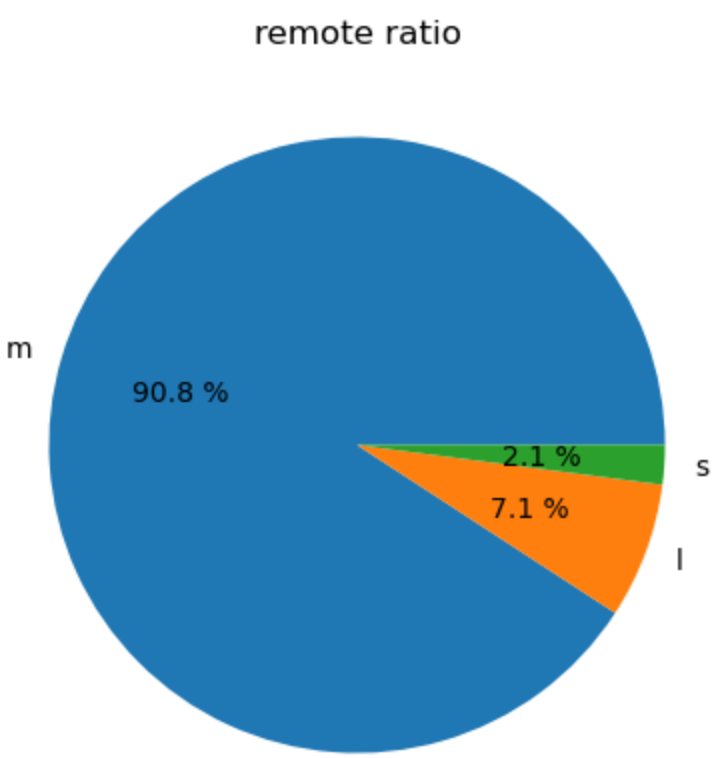
In [36]:
```
df.groupby("company_size")["company_location"].value_counts()
```

Out[36]:
```
company_size  company_location
L             US                  360
              IN                   37
              DE                   28
              CA                   27
              GB                   26
                                 ...
S             AT                    1
              AU                    1
              BE                    1
              EE                    1
              FI                    1
Name: count, Length: 150, dtype: int64
```

In [37]:
```
m=df.loc[df["company_size"]=="M"].count()[0]
l=df.loc[df["company_size"]=="L"].count()[0]
s=df.loc[df["company_size"]=="S"].count()[0]
```

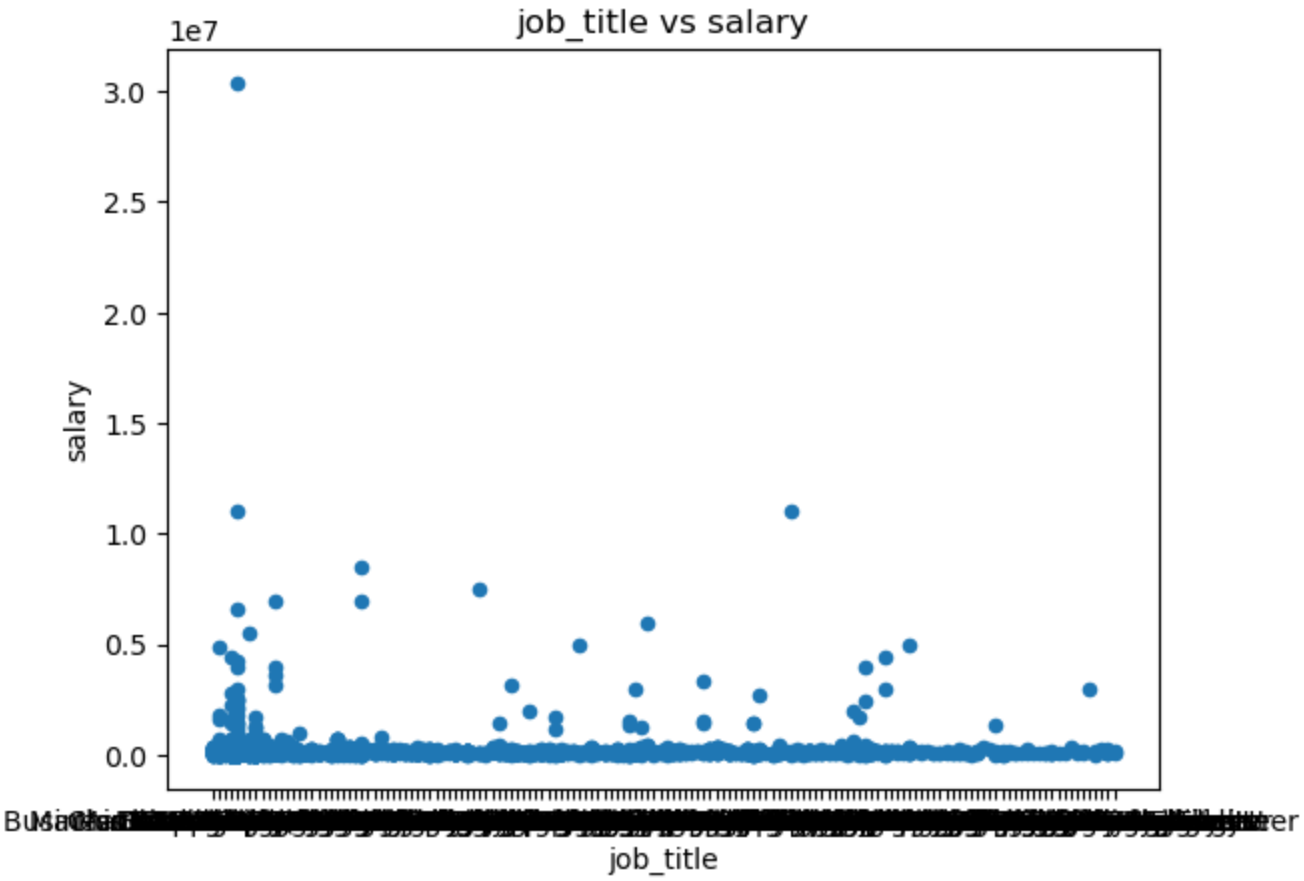
In [95]:
```
plt.figure(figsize=(8,5))
labels = ['m', 'l','s']
plt.pie([m, l,s], labels = labels, autopct='%1.1f %%')
plt.title("remote ratio")
```

Out[95]:
```
Text(0.5, 1.0, 'remote ratio')
```



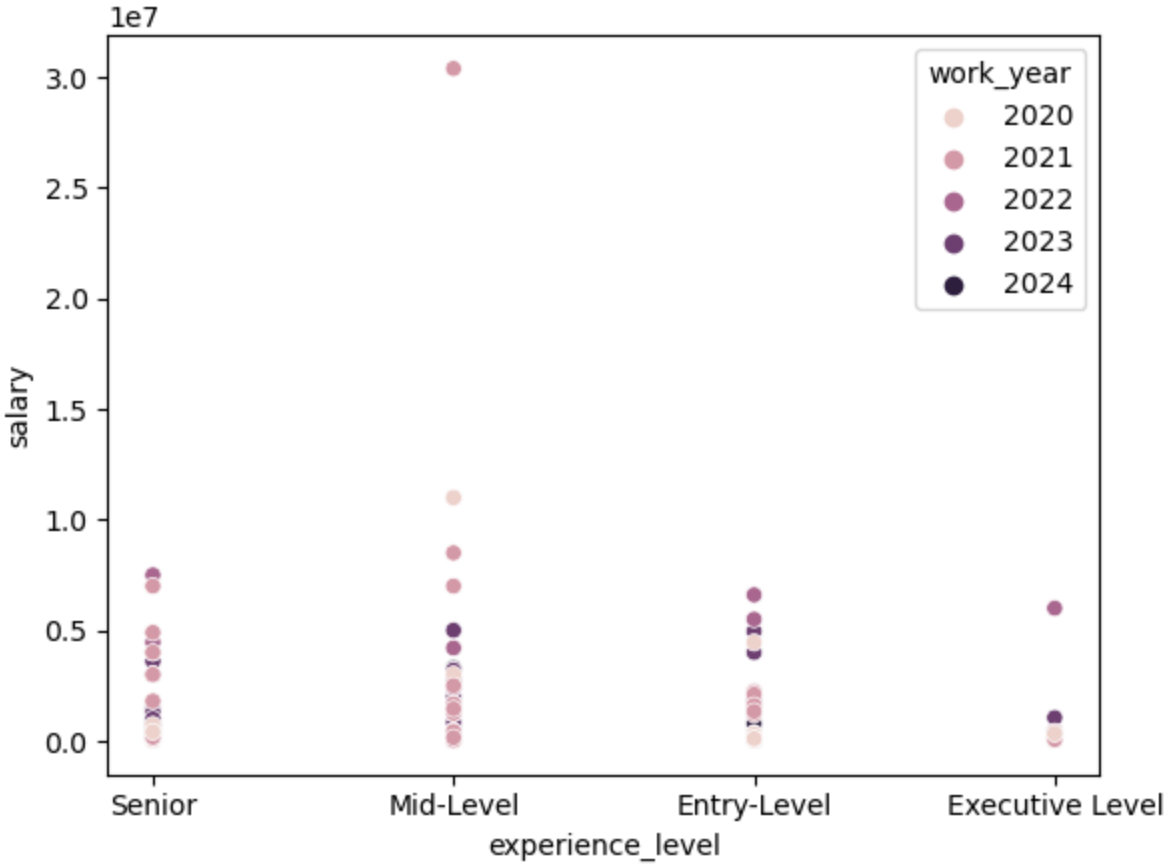
In [87]:
```
df.plot(kind='scatter',
        x='job_title',y='salary')
plt.title('job_title vs salary')
```

Out[87]:
```
Text(0.5, 1.0, 'job_title vs salary')
```

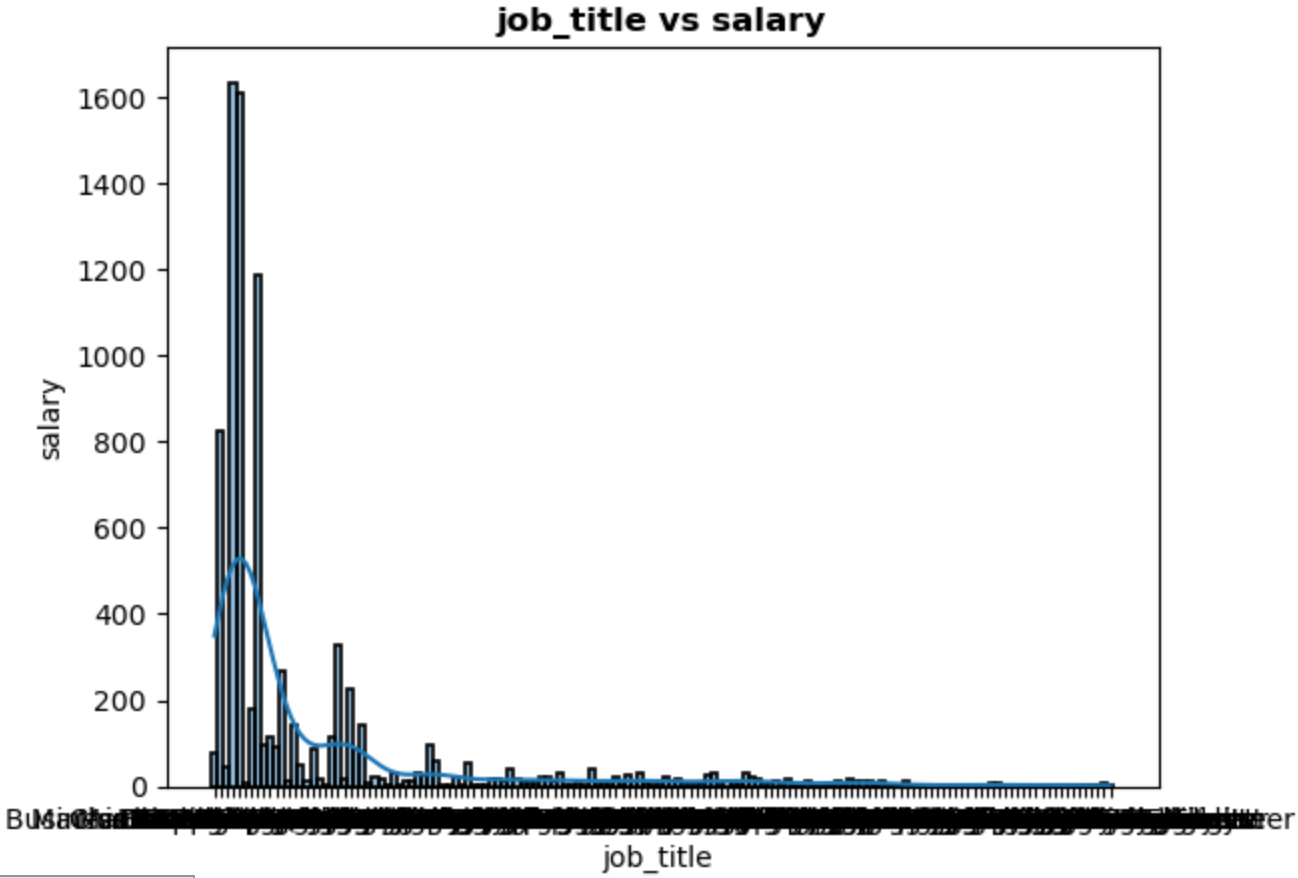


In [80]:
```
sns.scatterplot(x='experience_level', y='salary', hue='work_year', data=df)
```

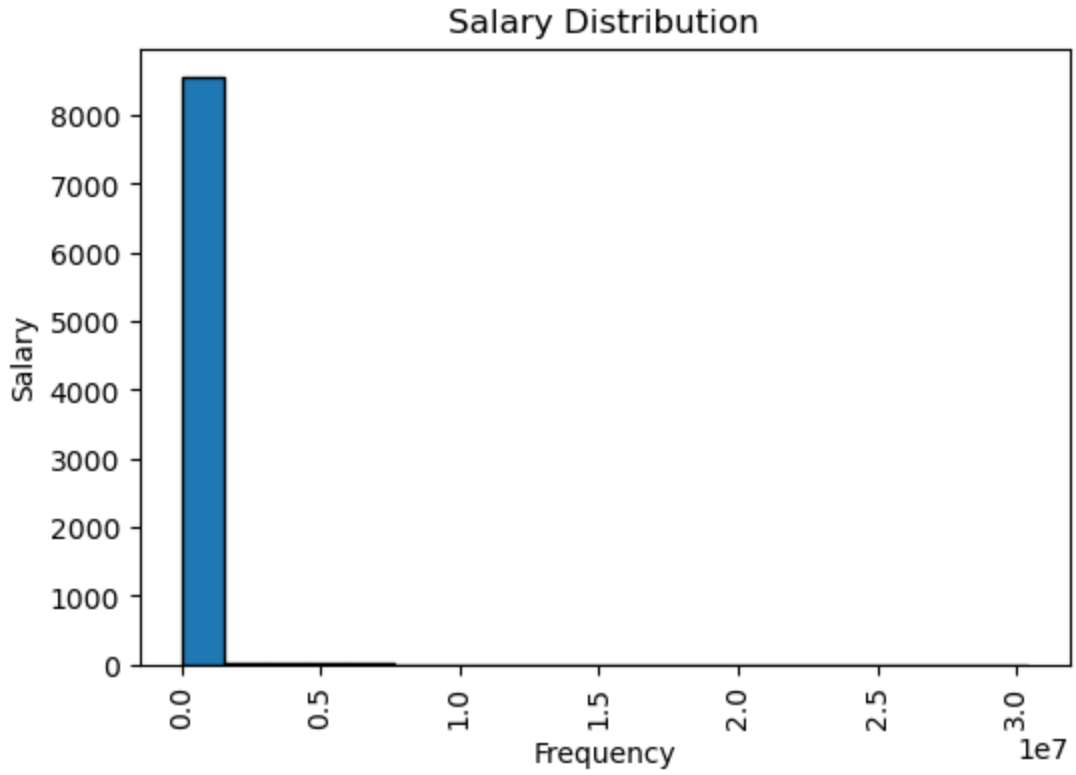
Out[80]: <Axes: xlabel='experience_level', ylabel='salary'>



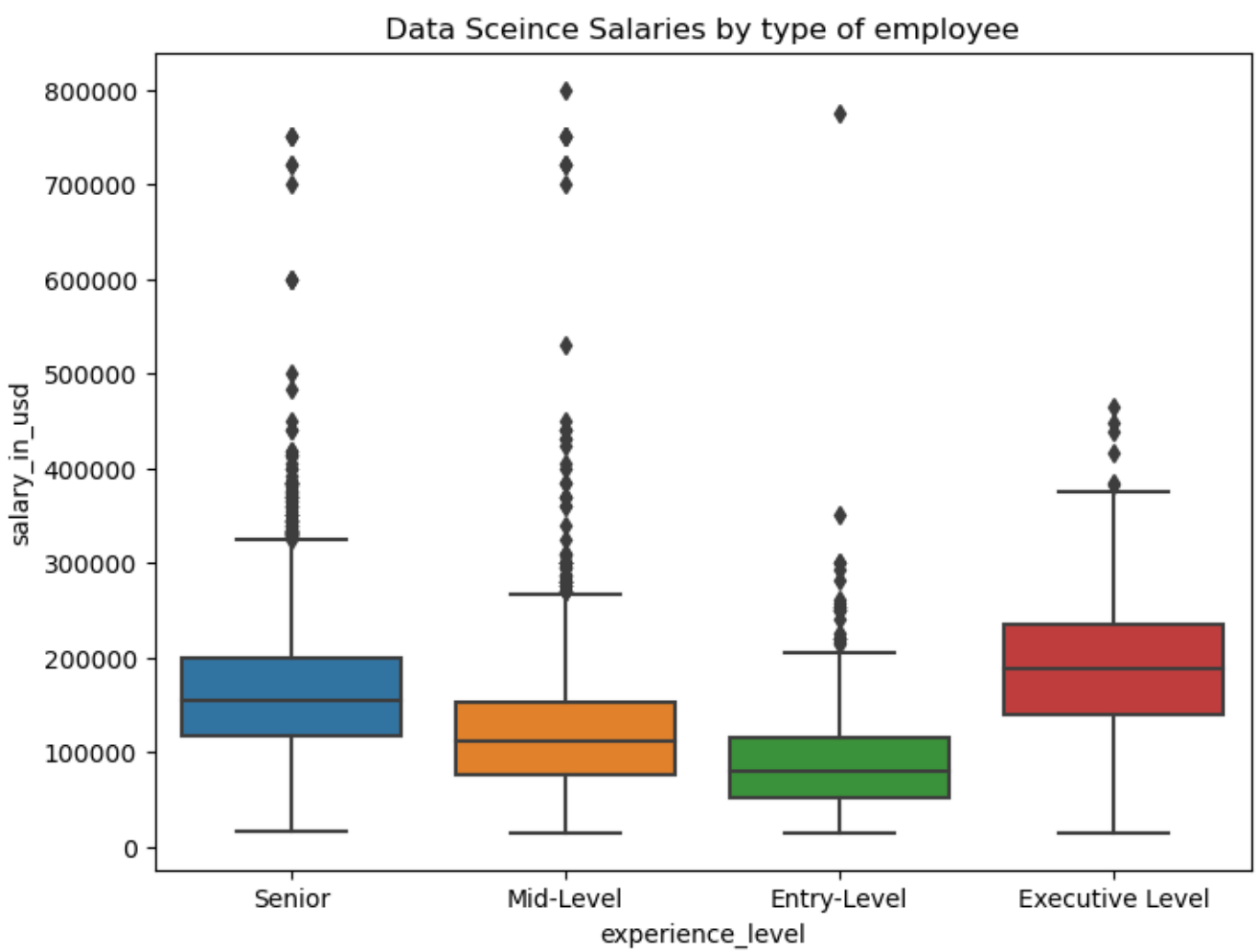
In [94]:
```
sns.histplot(data=df, x="job_title", kde=True, edgecolor="black", linewidth=1.2)
plt.title("job_title vs salary", weight="bold")
plt.xlabel("job_title ")
plt.ylabel("salary")
plt.show()
```



In [66]:
```
plt.figure(figsize=(6, 4))
plt.hist(df["salary"], bins=20, edgecolor="black")
plt.xlabel("Frequency")
plt.ylabel("Salary")
plt.xticks(rotation=90)
plt.title("Salary Distribution")
plt.show()
```

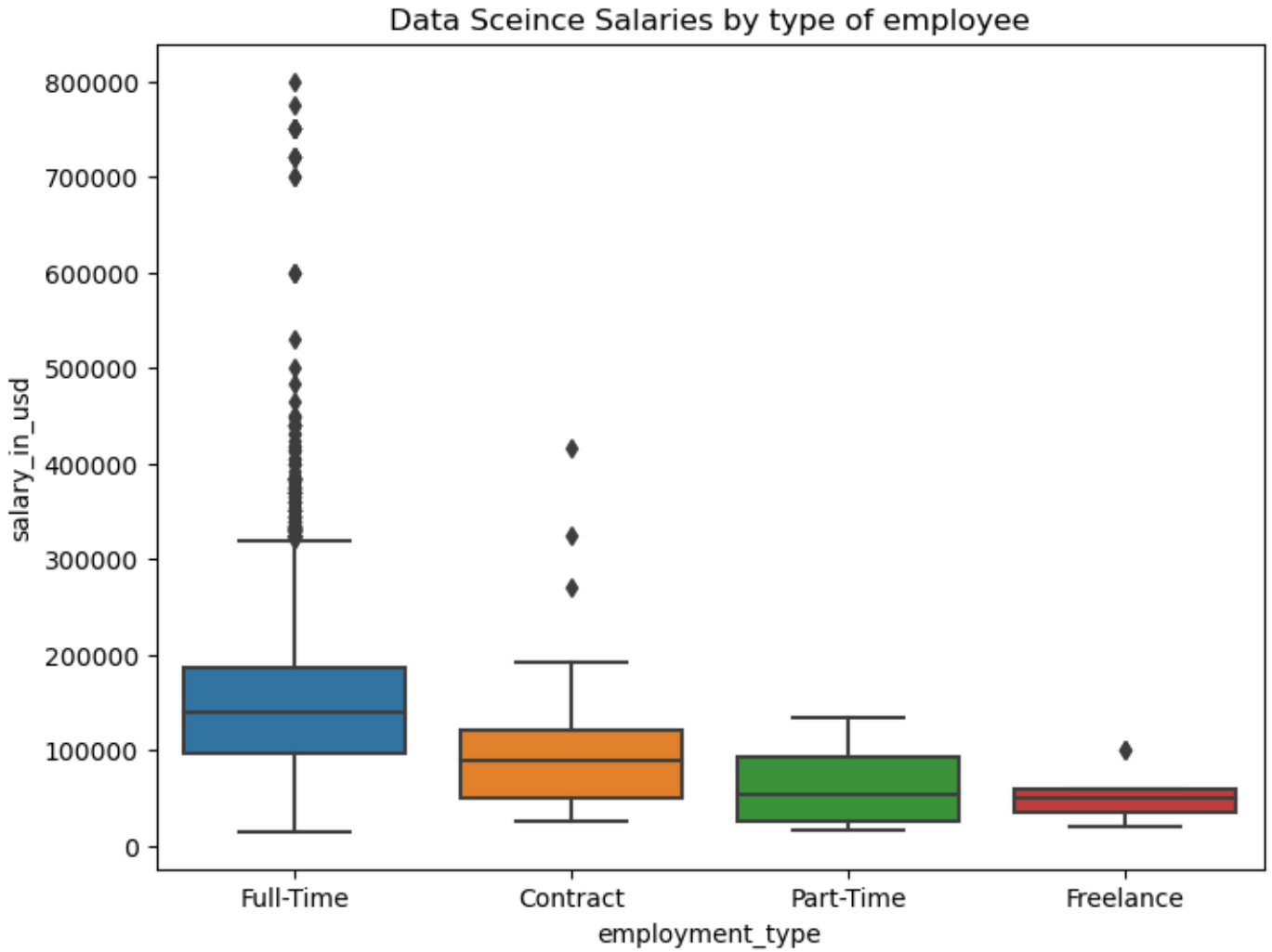


In [89]:
```
plt.figure(figsize=(8, 6))
sns.boxplot(df,x='experience_level',y='salary_in_usd')
plt.title("Data Sceince Salaries by type of employee")
plt.show()
```



In [134…

```
plt.figure(figsize=(8, 6))
sns.boxplot(df,x='employment_type',y='salary_in_usd')
plt.title("Data Sceince Salaries by type of employee")
plt.show()
```



In [96]:
```
df.employment_type.value_counts()
```

Out[96]:
```
employment_type
Full-Time    8534
Contract       26
Part-Time      22
Freelance      12
Name: count, dtype: int64
```

In [97]:
```
df.salary_in_usd.value_counts()
```

Out[97]:
```
salary_in_usd
150000    130
100000    126
120000    110
130000     94
160000     92
         ...
135800      1
240810      1
100758      1
154101      1
94665       1
Name: count, Length: 2596, dtype: int64
```

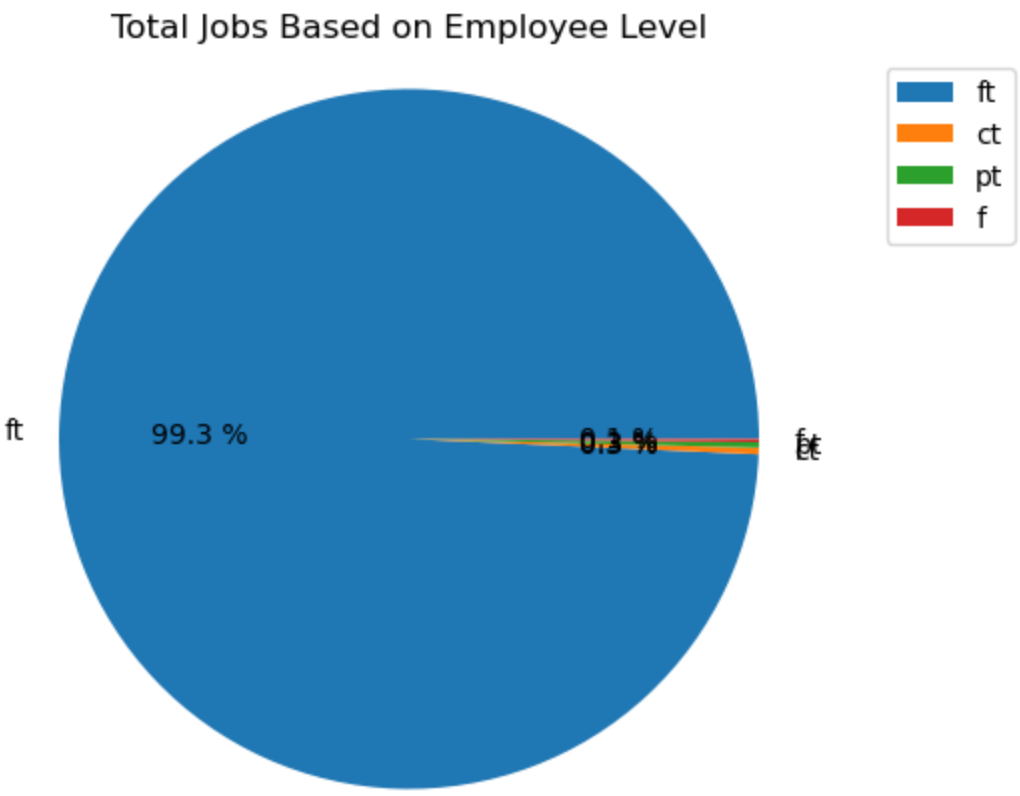
In [98]:
```
df.groupby("employment_type")["salary_in_usd"].value_counts()
```

Out[98]:
```
employment_type  salary_in_usd
Contract         50000           2
                 60000           2
                 105000          2
                 25500           1
                 191027          1
                                 ..
Part-Time        19073           1
                 21669           1
                 25216           1
                 28609           1
                 34320           1
Name: count, Length: 2624, dtype: int64
```

In [104…
```
ft=df.loc[df["employment_type"]=="Full-Time"].count()[0]
ct=df.loc[df["employment_type"]=="Contract"].count()[0]
pt=df.loc[df["employment_type"]=="Part-Time"].count()[0]
f=df.loc[df["employment_type"]=="Freelance"].count()[0]
```

In [132…
```
plt.figure(figsize=(8,5))
labels = ['ft','ct','pt','f']
plt.pie([ft,ct,pt,f], labels = labels, autopct='%1.1f %%',shadow=False)
plt.legend()
plt.axis('equal')
plt.show
plt.title("Total Jobs Based on Employee Level")
```

Out[132]:
```
Text(0.5, 1.0, 'Total Jobs Based on Employee Level')
```



In [ ]: