Brian D'Alelio DSE5002 Project 1

The CEO wants to understand how much to pay a data scientist, and potentially a team of data scientists in the future, both in the United States and Internationally. She understands that salaries vary widely across the world which creates uncertainty around this decision. The job landscape is seeing salaries increasing due to the great recession making the market highly competitive. My task is to analyze data science salaries and recommend a salary range. I'll present my recommendations visually in a Powerpoint presentation.

What should the company offer as a competitive salary for a data scientist? How do data science salaries differ around thw world, and what pay range would attract strong talent? Given global salary trends and rising competition, what is an appropriate compensation package for a full-time data scientist? Should the analysis account for experience level?

In [11]:

```
import pandas as pd
df = pd.read_csv("r project data-1-1.csv")
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 607 entries, 0 to 606
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   Unnamed: 0          607 non-null    int64 
 1   work_year           607 non-null    int64 
 2   experience_level    607 non-null    object
 3   employment_type     607 non-null    object
 4   job_title           607 non-null    object
 5   salary              607 non-null    int64 
 6   salary_currency     607 non-null    object
 7   salary_in_usd       607 non-null    int64 
 8   employee_residence  607 non-null    object
 9   remote_ratio        607 non-null    int64 
 10  company_location    607 non-null    object
 11  company_size        607 non-null    object
dtypes: int64(5), object(7)
memory usage: 57.0+ KB
```

In [12]:

```
df['experience_level']=pd.Categorical(df.experience_level)
df['employment_type']=pd.Categorical(df.employment_type)
df['job_title']=pd.Categorical(df.job_title)
df['salary_currency']=pd.Categorical(df.salary_currency)
df['employee_residence']=pd.Categorical(df.employee_residence)
df['company_location']=pd.Categorical(df.company_location)
df['company_size']=pd.Categorical(df.company_size)
df['remote_ratio']=pd.Categorical(df.remote_ratio)

df.dtypes
```

Out[12]:

```
Unnamed: 0                int64
work_year                 int64
experience_level       category
employment_type        category
job_title              category
salary                    int64
salary_currency        category
salary_in_usd             int64
employee_residence     category
remote_ratio           category
company_location       category
company_size           category
dtype: object
```

In [13]:

```
df.describe(include='all')
```

Out[13]:

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salar
count	607.000000	607.000000	607	607	607	6.070000e+0
unique	NaN	NaN	4	4	50	Na
top	NaN	NaN	SE	FT	Data Scientist	Na
freq	NaN	NaN	280	588	143	Na
mean	303.000000	2021.405272	NaN	NaN	NaN	3.240001e+0
std	175.370085	0.692133	NaN	NaN	NaN	1.544357e+0
min	0.000000	2020.000000	NaN	NaN	NaN	4.000000e+0
25%	151.500000	2021.000000	NaN	NaN	NaN	7.000000e+0
50%	303.000000	2022.000000	NaN	NaN	NaN	1.150000e+0
75%	454.500000	2022.000000	NaN	NaN	NaN	1.650000e+0
max	606.000000	2022.000000	NaN	NaN	NaN	3.040000e+0

In [14]:

```
df['experience_level'].value_counts()
```

Out[14]:

```
experience_level
SE    280
MI    213
EN     88
EX     26
Name: count, dtype: int64
```

In [15]:

```
df['employment_type'].value_counts()
```

Out[15]:

```
employment_type
FT    588
PT     10
CT      5
FL      4
Name: count, dtype: int64
```

In [16]:

```
df['job_title'].value_counts().head(20)
```

Out[16]:

```
job_title
Data Scientist                143
Data Engineer                 132
Data Analyst                   97
Machine Learning Engineer      41
Research Scientist             16
Data Science Manager           12
Data Architect                 11
Big Data Engineer               8
Machine Learning Scientist      8
Data Analytics Manager          7
Data Science Consultant         7
Director of Data Science        7
Principal Data Scientist        7
AI Scientist                    7
Computer Vision Engineer        6
BI Data Analyst                 6
ML Engineer                     6
Lead Data Engineer              6
Business Data Analyst           5
Applied Data Scientist          5
Name: count, dtype: int64
```

In [17]:

```
df['employee_residence'].value_counts().head(20)
```

Out[17]:

```
employee_residence
US    332
GB     44
IN     30
CA     29
DE     25
FR     18
ES     15
GR     13
JP      7
PK      6
BR      6
PT      6
NL      5
IT      4
PL      4
RU      4
TR      3
AU      3
VN      3
AT      3
Name: count, dtype: int64
```

In [18]:

```
df['company_location'].value_counts().head(20)
```

Out[18]:

```
company_location
US    355
GB     47
CA     30
DE     28
IN     24
FR     15
ES     14
GR     11
JP      6
AT      4
PL      4
NL      4
PT      4
AE      3
BR      3
DK      3
AU      3
LU      3
MX      3
TR      3
Name: count, dtype: int64
```

In [19]:

```
df['company_size'].value_counts()
```

Out[19]:

```
company_size
M    326
L    198
S     83
Name: count, dtype: int64
```

In [20]:
```
df['remote_ratio'].value_counts()
```

Out[20]:
```
remote_ratio
100    381
0      127
50      99
Name: count, dtype: int64
```

In [21]:
```
df['salary_in_usd'].describe()
```

Out[21]:
```
count       607.000000
mean     112297.869852
std       70957.259411
min        2859.000000
25%       62726.000000
50%      101570.000000
75%      150000.000000
max      600000.000000
Name: salary_in_usd, dtype: float64
```

In [22]:
```
df['salary'].describe()
```

Out[22]:
```
count    6.070000e+02
mean     3.240001e+05
std      1.544357e+06
min      4.000000e+03
25%      7.000000e+04
50%      1.150000e+05
75%      1.650000e+05
max      3.040000e+07
Name: salary, dtype: float64
```

In [23]:
```
df['salary_currency'].value_counts()
```

Out[23]:
```
salary_currency
USD    398
EUR     95
GBP     44
INR     27
CAD     18
JPY      3
PLN      3
TRY      3
AUD      2
BRL      2
HUF      2
CNY      2
DKK      2
MXN      2
SGD      2
CHF      1
CLP      1
Name: count, dtype: int64
```

In [24]:
```
df['work_year'].value_counts()
```

Out[24]:

```
work_year
2022    318
2021    217
2020     72
Name: count, dtype: int64
```

In [ ]:

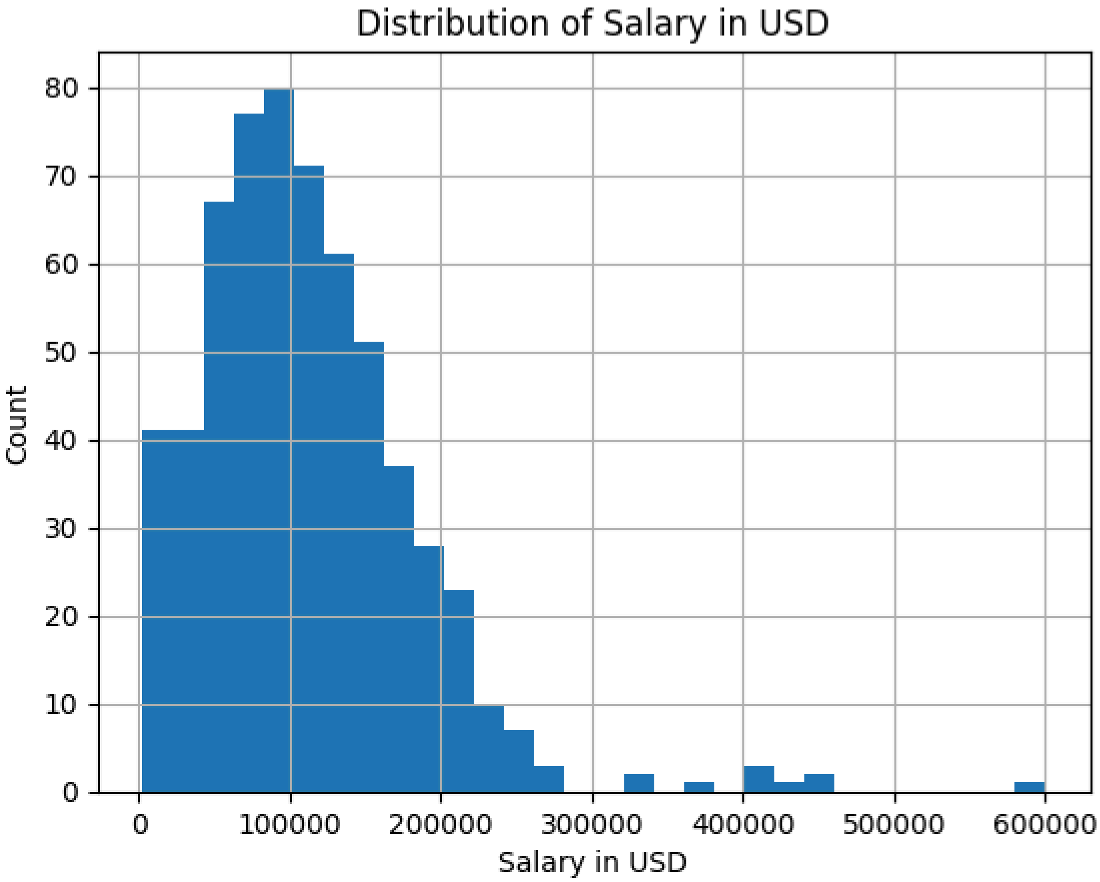
```
work_year
    Values range from 2020-2022
    Most records are recent as Median is 2022
experience_level
    4 unique levels - EN, MI, SE, EX
employment_type
    4 categories. Full time (FT) most common by a wide margin
job_title
    50 unique titles. Most common is "Data Scientist".
    Data Engineer, Data Analyst, Machine Learning Engineer also significantly repre
salary
    Salary_in_usd will be cleaner to work with as this is showing salaries in many
salary_currency
    most salaries are denominated in USD. The Euro and Great British Pound also hav
salary_in_usd
    Large SD and min max disparity due to differences in geography, experience leve
    Median is lower than the mean. This suggests a small number of high earners dra
employee_residence
    There are many countries represented in the data, but the United States is the
remote_ratio
    Fully remote roles are the most common, aligning with a global shift in this di
company_location
     company location looks similar to employee residence. This makes sense as empl
     employer of the same country. The differences we do see here indicate cross-bo
     at least somewhat common.
company_size
    Three categories - Large, Medium, and Small. Medium and large companies make up
    of the data set, with Medium leading. Small companies appear less often, but st
    meaningful share.
```

In [25]:

```python
import matplotlib.pyplot as plt
```

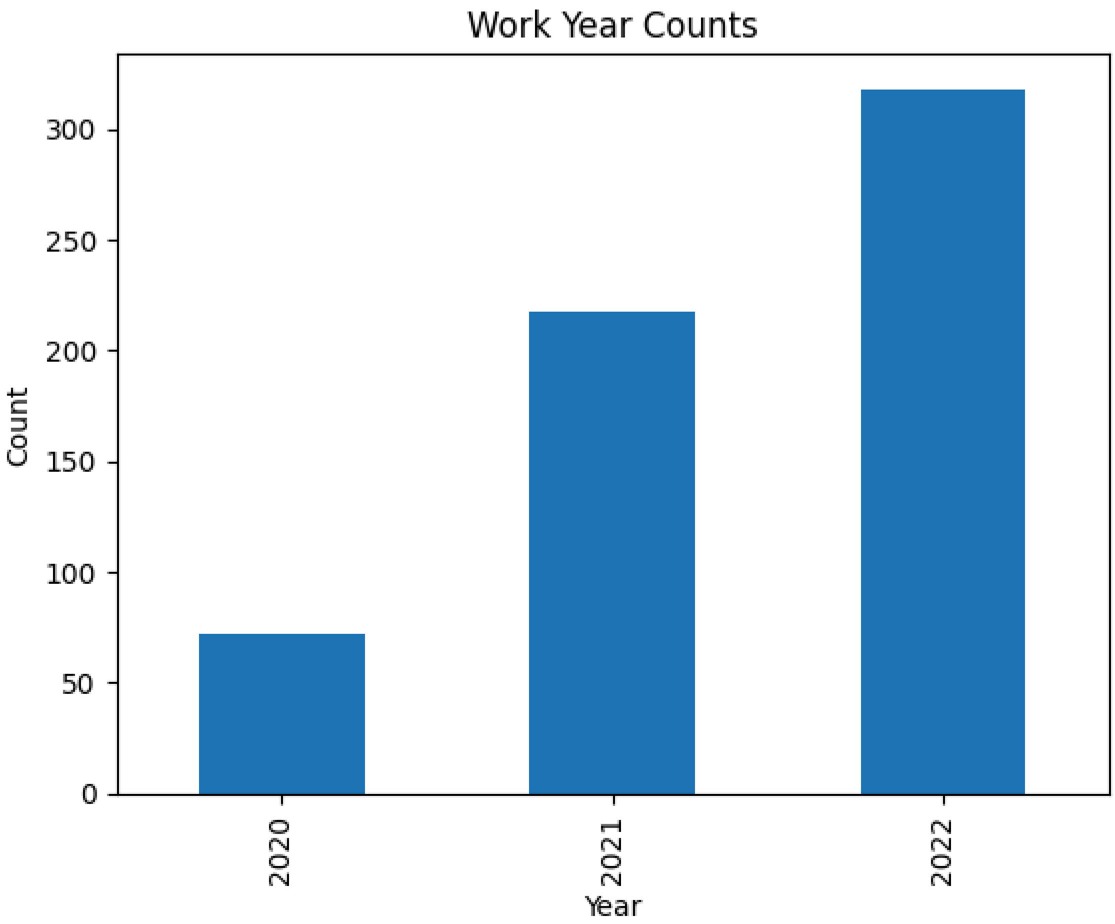
In [26]:

```python
df['salary_in_usd'].hist(bins=30)
plt.title("Distribution of Salary in USD")
plt.xlabel("Salary in USD")
plt.ylabel("Count")
plt.show()
```

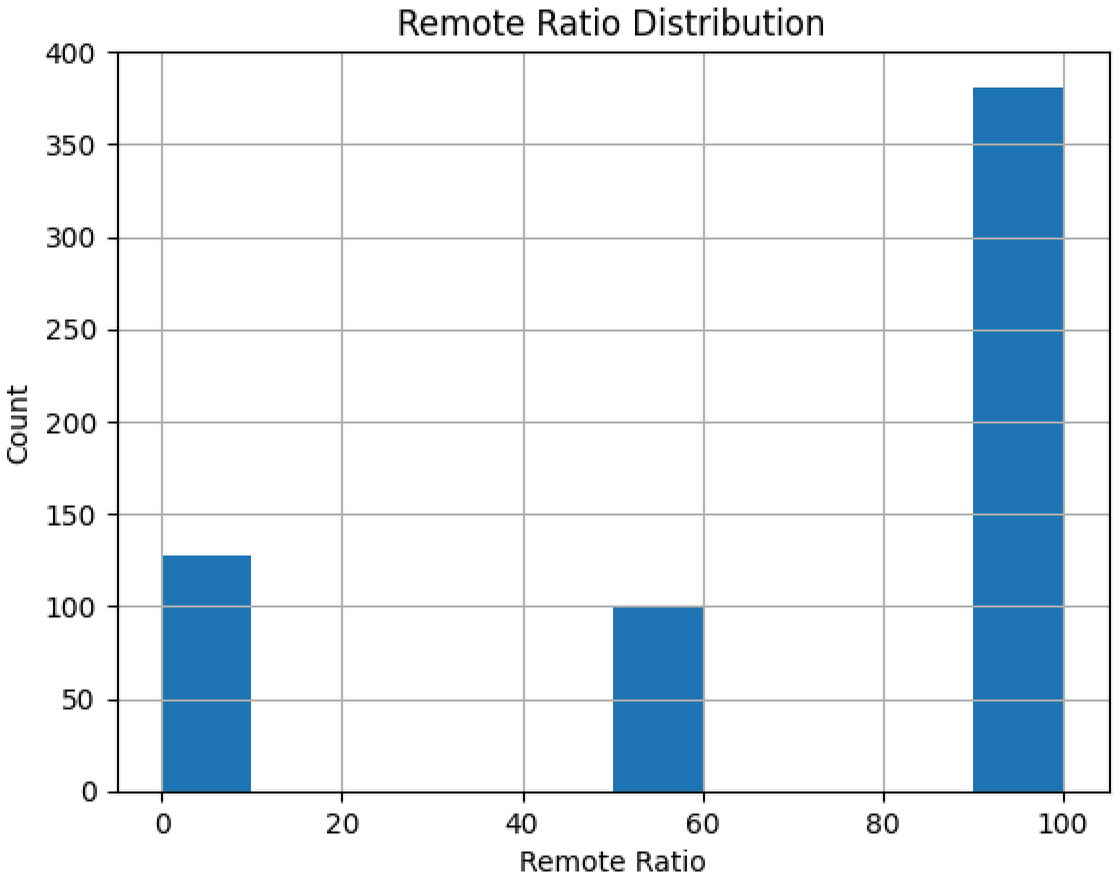


In [27]:

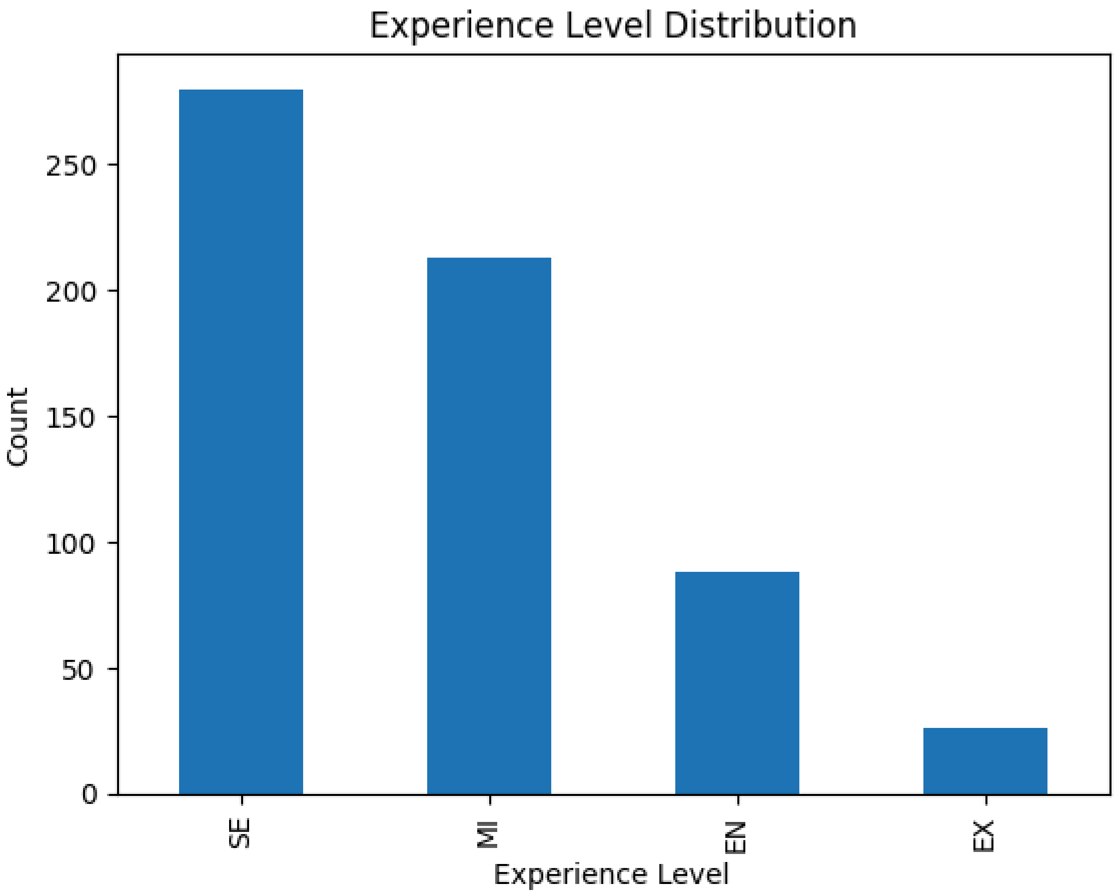
```
df['work_year'].value_counts().sort_index().plot(kind='bar')
plt.title("Work Year Counts")
plt.xlabel("Year")
plt.ylabel("Count")
plt.show()
```



In [28]:
```
df['remote_ratio'].hist()
plt.title("Remote Ratio Distribution")
plt.xlabel("Remote Ratio")
plt.ylabel("Count")
plt.show()
```

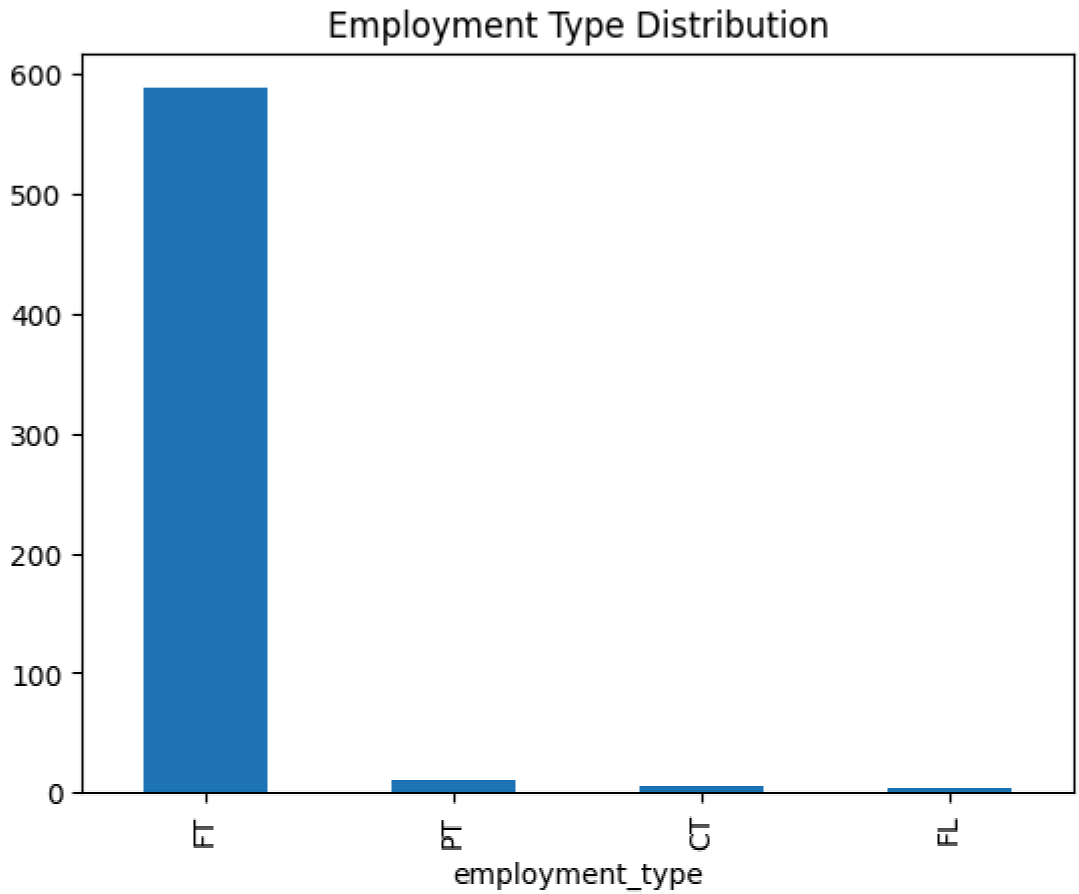


In [29]:
```
df['experience_level'].value_counts().plot(kind='bar')
plt.title("Experience Level Distribution")
plt.xlabel("Experience Level")
plt.ylabel("Count")
plt.show()
```



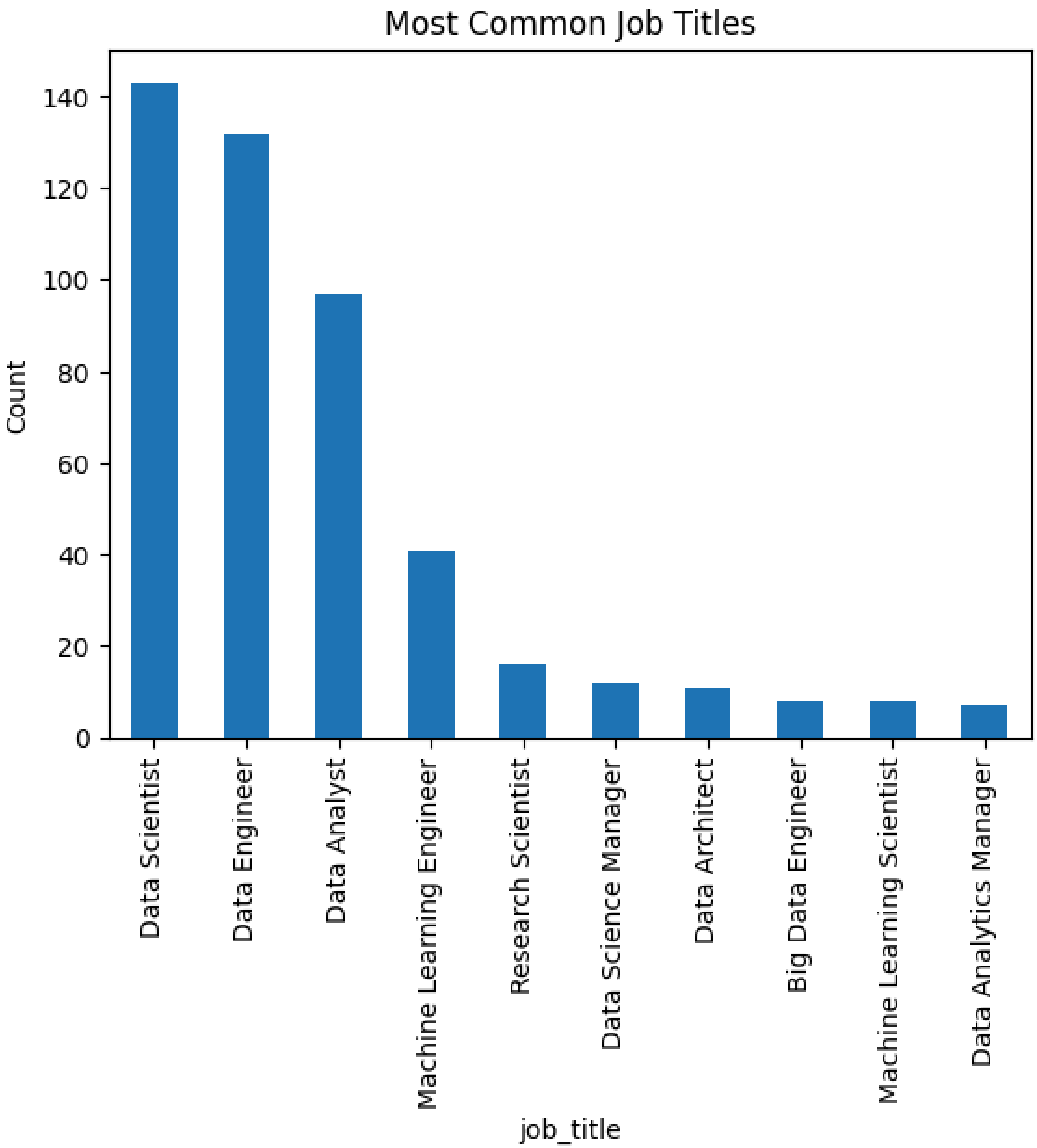
In [30]:

```
df['employment_type'].value_counts().plot(kind='bar')
plt.title("Employment Type Distribution")
plt.show()
```

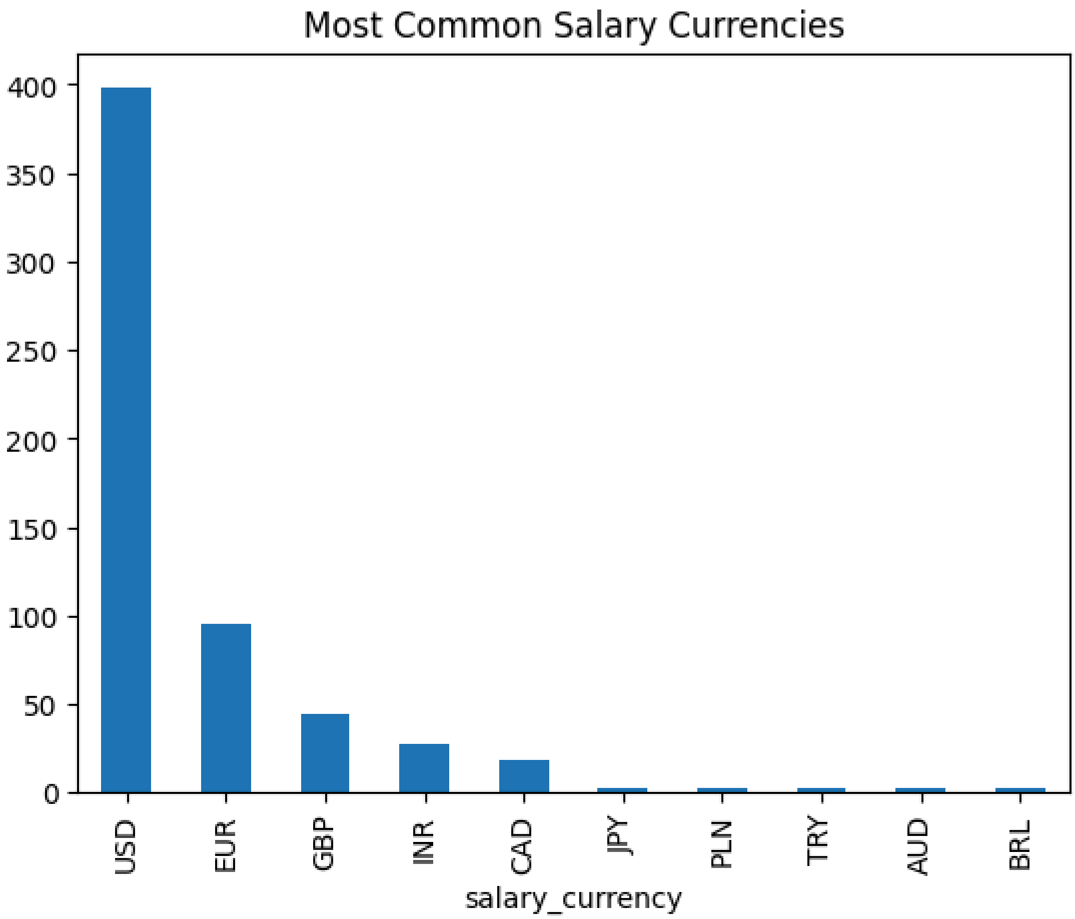


In [31]:

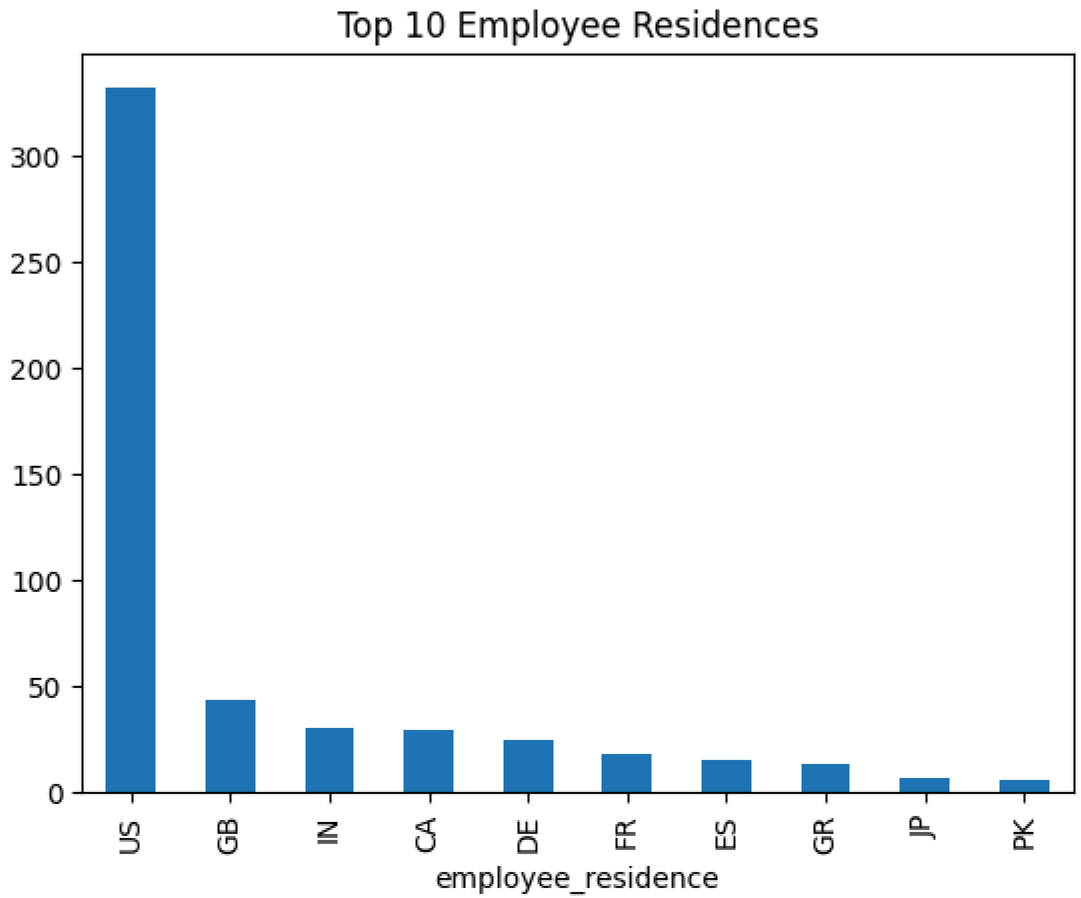
```
df['job_title'].value_counts().head(10).plot(kind='bar')
plt.title("Most Common Job Titles")
plt.ylabel("Count")
plt.show()
```



In [32]:
```
df['salary_currency'].value_counts().head(10).plot(kind='bar')
plt.title("Most Common Salary Currencies")
plt.show()
```

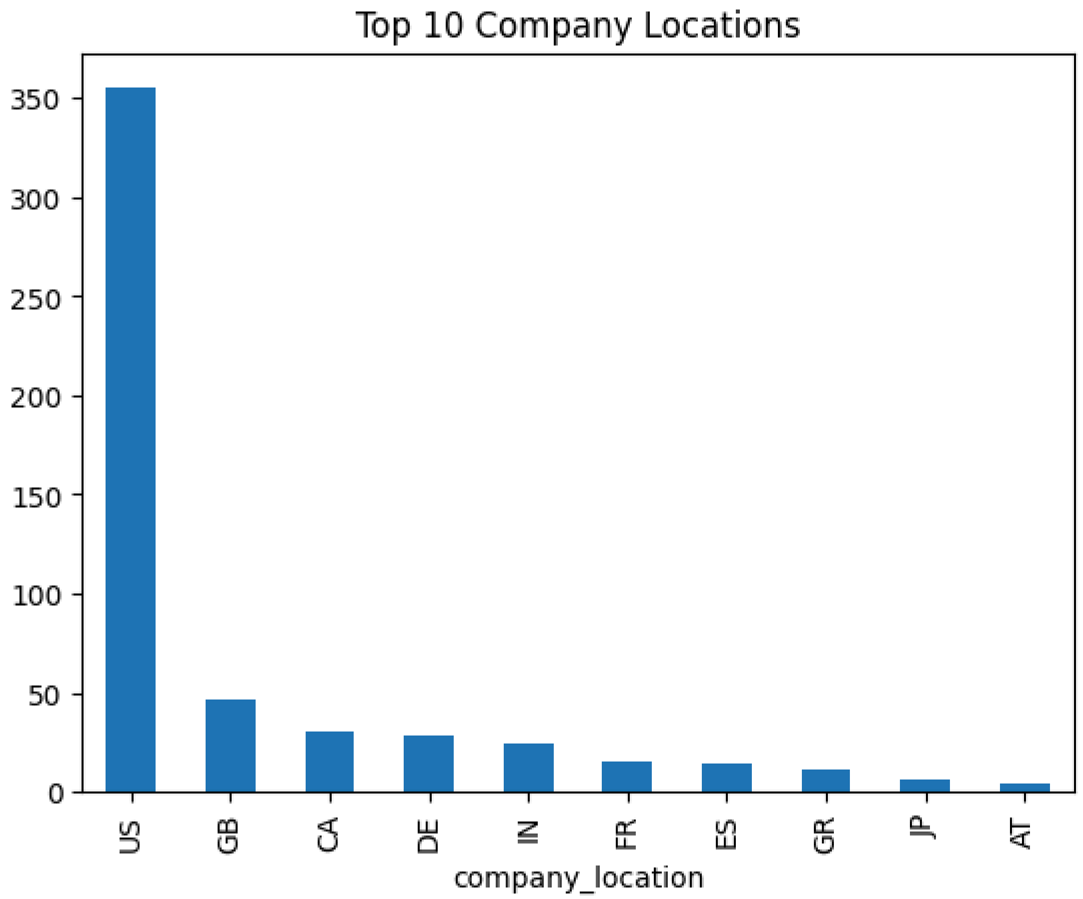


In [33]:
```
df['employee_residence'].value_counts().head(10).plot(kind='bar')
plt.title("Top 10 Employee Residences")
plt.show()
```



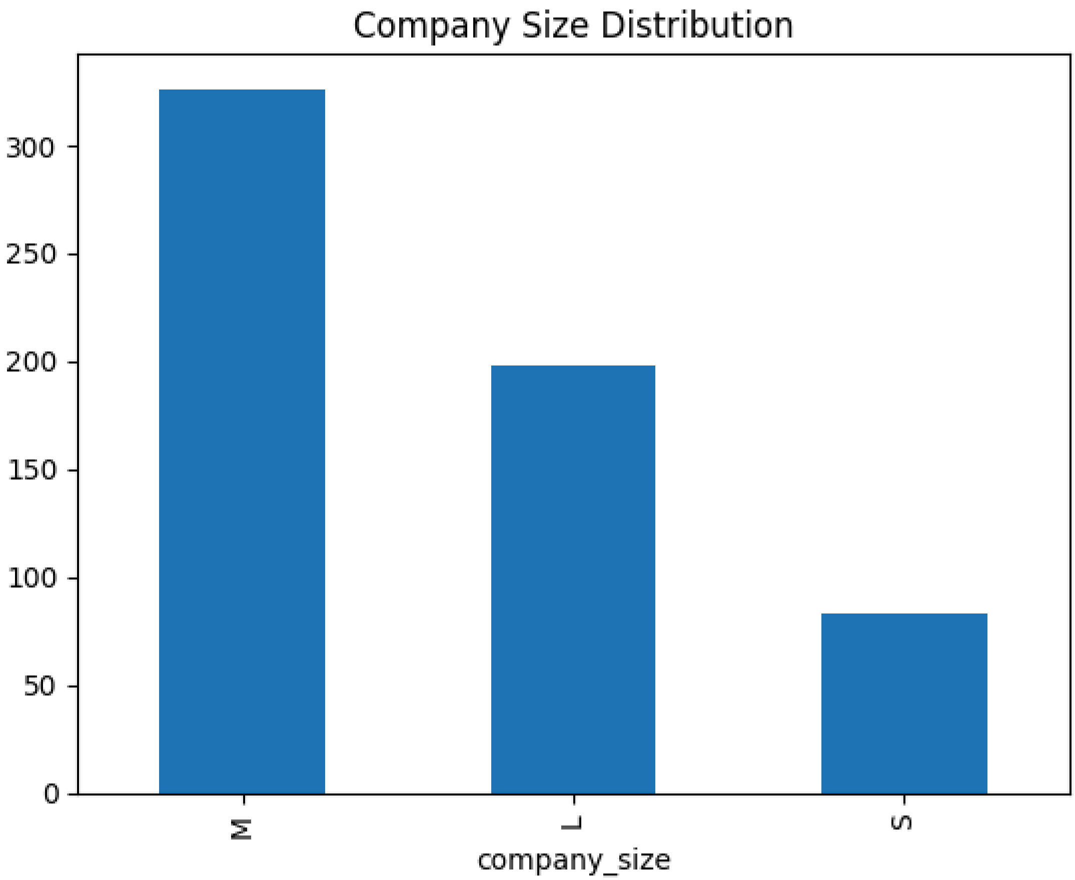
In [34]:

```
df['company_location'].value_counts().head(10).plot(kind='bar')
plt.title("Top 10 Company Locations")
plt.show()
```



In [35]:

```
df['company_size'].value_counts().plot(kind='bar')
plt.title("Company Size Distribution")
plt.show()
```



In [36]:

```
exp_map = {
    "EN": "Entry-level",
    "MI": "Mid-level",
    "SE": "Senior-level",
    "EX": "Executive"
}

exp_level = (
    df.groupby("experience_level")["salary_in_usd"]
      .mean()
      .round(0)
      .reset_index()
)

exp_level["experience_description"] = exp_level["experience_level"].map(exp_map)

exp_level = exp_level.sort_values("salary_in_usd", ascending=False)
exp_level
```

```
C:\Users\Brian\AppData\Local\Temp\ipykernel_22240\891921729.py:9: FutureWarning: The
default of observed=False is deprecated and will be changed to True in a future vers
ion of pandas. Pass observed=False to retain current behavior or observed=True to ad
opt the future default and silence this warning.
  df.groupby("experience_level")["salary_in_usd"]
```

Out[36]:

	experience_level	salary_in_usd	experience_description
1	EX	199392.0	Executive
3	SE	138617.0	Senior-level
2	MI	87996.0	Mid-level
0	EN	61643.0	Entry-level

In [37]:

```python
import pycountry

def country_name(code):
    try:
        return pycountry.countries.get(alpha_2=code).name
    except:
        mapping = {"GR": "Greece", "PT": "Portugal"}
        return mapping.get(code, None)

emp_residence = (
    df.groupby("employee_residence")["salary_in_usd"]
      .agg(mean_salary_usd="mean", median_salary_usd="median")
      .round(0)
      .reset_index()
)

emp_residence["country_name"] = emp_residence["employee_residence"].apply(country_n
emp_residence.sort_values("mean_salary_usd", ascending=False, inplace=True)
emp_residence.head(10)
```

```
C:\Users\Brian\AppData\Local\Temp\ipykernel_22240\626899275.py:11: FutureWarning: Th
e default of observed=False is deprecated and will be changed to True in a future ve
rsion of pandas. Pass observed=False to retain current behavior or observed=True to
adopt the future default and silence this warning.
  df.groupby("employee_residence")["salary_in_usd"]
```

Out[37]:

	employee_residence	mean_salary_usd	median_salary_usd	country_name
38	MY	200000.0	200000.0	Malaysia
45	PR	160000.0	160000.0	Puerto Rico
55	US	149194.0	138475.0	United States
41	NZ	125000.0	125000.0	New Zealand
9	CH	122346.0	122346.0	Switzerland
3	AU	108043.0	87425.0	Australia
49	RU	105750.0	72500.0	Russian Federation
50	SG	104176.0	104176.0	Singapore
32	JP	103538.0	74000.0	Japan
31	JE	100000.0	100000.0	Jersey

In [38]:

```
job_title = (
    df.groupby("job_title")
      .agg(
          count=("job_title", "size"),
          mean_salary_usd=("salary_in_usd", "mean"),
          median_salary_usd=("salary_in_usd", "median")
      )
      .sort_values("count", ascending=False)
      .head(15)
      .round(0)
      .reset_index()
)

job_title
```

```
C:\Users\Brian\AppData\Local\Temp\ipykernel_22240\1116427211.py:2: FutureWarning: Th
e default of observed=False is deprecated and will be changed to True in a future ve
rsion of pandas. Pass observed=False to retain current behavior or observed=True to
adopt the future default and silence this warning.
  df.groupby("job_title")
```

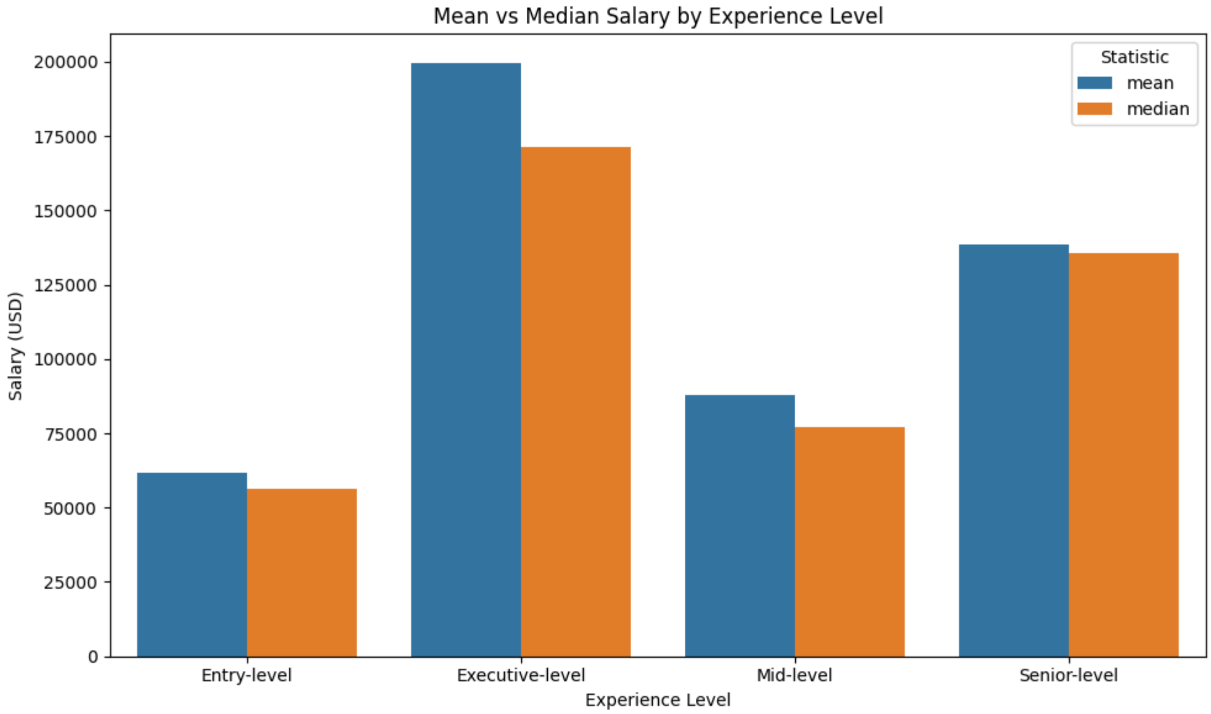
Out[38]:

	job_title	count	mean_salary_usd	median_salary_usd
0	Data Scientist	143	108188.0	103691.0
1	Data Engineer	132	112725.0	105500.0
2	Data Analyst	97	92893.0	90320.0
3	Machine Learning Engineer	41	104880.0	87932.0
4	Research Scientist	16	109020.0	76264.0
5	Data Science Manager	12	158328.0	155750.0
6	Data Architect	11	177874.0	180000.0
7	Big Data Engineer	8	51974.0	41306.0
8	Machine Learning Scientist	8	158412.0	156500.0
9	Data Analytics Manager	7	127134.0	120000.0
10	Data Science Consultant	7	69421.0	76833.0
11	Director of Data Science	7	195074.0	168000.0
12	Principal Data Scientist	7	215242.0	173762.0
13	AI Scientist	7	66136.0	45896.0
14	Computer Vision Engineer	6	44419.0	26304.0

In [39]:

```
import seaborn as sns

exp_level = df.groupby("experience_level")["salary_in_usd"].agg(["mean", "median"])

exp_level["experience_level"] = exp_level["experience_level"].map({
    "EN": "Entry-level",
    "MI": "Mid-level",
    "SE": "Senior-level",
    "EX": "Executive-level"
})

exp_level_melted = exp_level.melt(id_vars="experience_level", value_vars=["mean", "
                                  var_name="Statistic", value_name="Salary")

plt.figure(figsize=(10,6))
sns.barplot(data=exp_level_melted, x="experience_level", y="Salary", hue="Statistic
plt.xlabel("Experience Level")
plt.ylabel("Salary (USD)")
plt.title("Mean vs Median Salary by Experience Level")
plt.tight_layout()
plt.show()
```

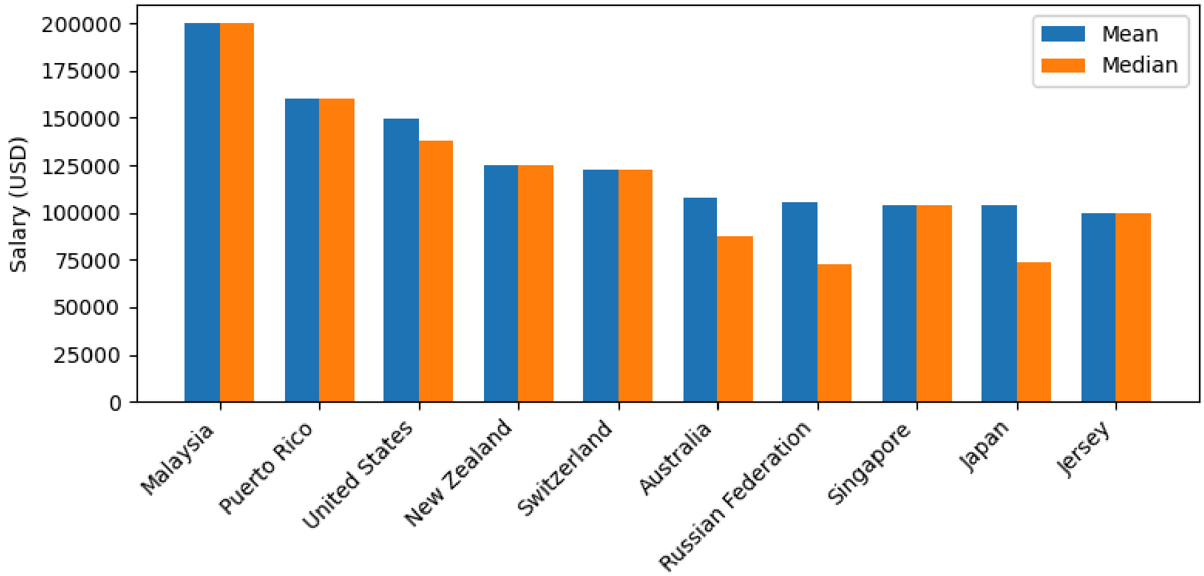
```
C:\Users\Brian\AppData\Local\Temp\ipykernel_22240\2736388065.py:3: FutureWarning: Th
e default of observed=False is deprecated and will be changed to True in a future ve
rsion of pandas. Pass observed=False to retain current behavior or observed=True to
adopt the future default and silence this warning.
  exp_level = df.groupby("experience_level")["salary_in_usd"].agg(["mean", "media
n"]).reset_index()
```



In [44]:

```
top10 = (
    emp_residence
    .dropna(subset=["country_name"])
    .sort_values("mean_salary_usd", ascending=False)
    .head(10)
)

plt.figure(figsize=(8,4))
x = range(len(top10))
width = 0.35
```

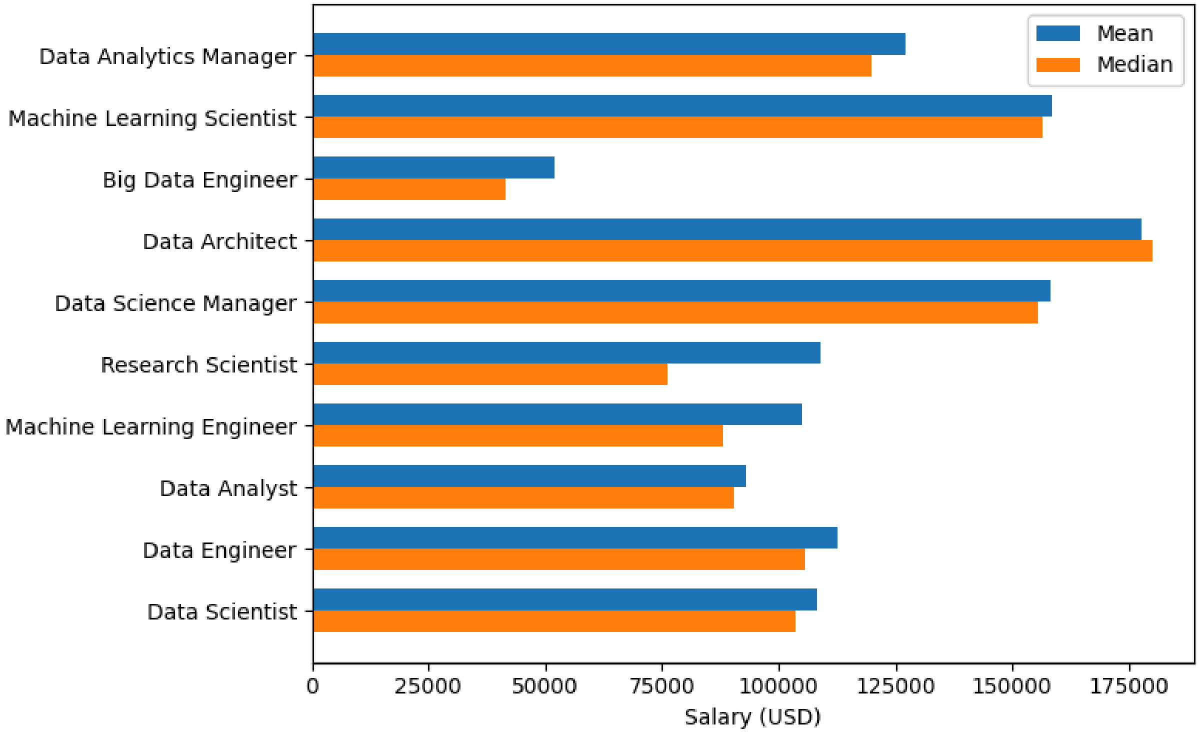
```
plt.bar([i - width/2 for i in x], top10["mean_salary_usd"], width, label="Mean")
plt.bar([i + width/2 for i in x], top10["median_salary_usd"], width, label="Median"

plt.xticks(list(x), top10["country_name"], rotation=45, ha="right")
plt.ylabel("Salary (USD)")
plt.legend()
plt.tight_layout()
plt.show()
```



In [46]:

```
job_title = (
    df.groupby("job_title")["salary_in_usd"]
      .agg(mean_salary_usd="mean", median_salary_usd="median", count="size")
      .reset_index()
)

top_jobs = job_title.sort_values("count", ascending=False).head(10)

plt.figure(figsize=(8,5))

y = range(len(top_jobs))
width = 0.35

plt.barh([i + width/2 for i in y], top_jobs["mean_salary_usd"], height=width, label
plt.barh([i - width/2 for i in y], top_jobs["median_salary_usd"], height=width, lab

plt.yticks(y, top_jobs["job_title"])
plt.xlabel("Salary (USD)")
plt.legend()
plt.tight_layout()
plt.show()
```

```
C:\Users\Brian\AppData\Local\Temp\ipykernel_22240\410625151.py:2: FutureWarning: The
default of observed=False is deprecated and will be changed to True in a future vers
ion of pandas. Pass observed=False to retain current behavior or observed=True to ad
opt the future default and silence this warning.
  df.groupby("job_title")["salary_in_usd"]
```



Summary:

This analysis explores global data science salaries to identify a competitive pay range for hiring a full-time data scientist. Salaries increase significantly by experience level, vary by employee location (with U.S. workers earning among the highest), and differ across specific job titles. By combining these factors, we can identify salary expectations that will position the company to attract strong candidates in a competitive, rapidly growing market.