In [3]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv("ds_jobs.csv")
```

In [132…
```
print(df)
```

```
     S. no  work_year experience_level employment_type  \
0        1       2020               MI              FT
1        2       2020               SE              FT
2        3       2020               SE              FT
3        4       2020               MI              FT
4        5       2020               SE              FT
..     ...        ...              ...             ...
602    603       2022               SE              FT
603    604       2022               SE              FT
604    605       2022               SE              FT
605    606       2022               SE              FT
606    607       2022               MI              FT

                    job_title  salary salary_currency  salary_in_usd  \
0              Data Scientist   90000             EUR          79833
1    Machine Learning Scientist  260000             USD         260000
2            Big Data Engineer   85000             GBP         109024
3          Product Data Analyst   20000             USD          20000
4     Machine Learning Engineer  150000             USD         150000
..                          ...     ...             ...            ...
602              Data Engineer  154000             USD         154000
603              Data Engineer  126000             USD         126000
604               Data Analyst  129000             USD         129000
605               Data Analyst  150000             USD         150000
606               AI Scientist  200000             USD         200000

    employee_residence  remote_ratio company_location company_size
0                   DE             0               DE            L
1                   JP             0               JP            S
2                   GB            50               GB            M
3                   HN             0               HN            S
4                   US            50               US            L
..                 ...           ...              ...          ...
602                 US           100               US            M
603                 US           100               US            M
604                 US             0               US            M
605                 US           100               US            M
606                 IN           100               US            L

[607 rows x 12 columns]
```

In [37]:
```
print(df.head())
```

```
   S. no  work_year experience_level employment_type  \
0      1       2020               MI              FT
1      2       2020               SE              FT
2      3       2020               SE              FT
3      4       2020               MI              FT
4      5       2020               SE              FT

                   job_title  salary salary_currency  salary_in_usd  \
0             Data Scientist   70000             EUR          79833
1  Machine Learning Scientist  260000             USD         260000
2          Big Data Engineer   85000             GBP         109024
3        Product Data Analyst   20000             USD          20000
4   Machine Learning Engineer  150000             USD         150000

  employee_residence  remote_ratio company_location company_size
0                 DE             0               DE            L
1                 JP             0               JP            S
2                 GB            50               GB            M
3                 HN             0               HN            S
4                 US            50               US            L
```

In [38]:
```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 607 entries, 0 to 606
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   S. no               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
None
```

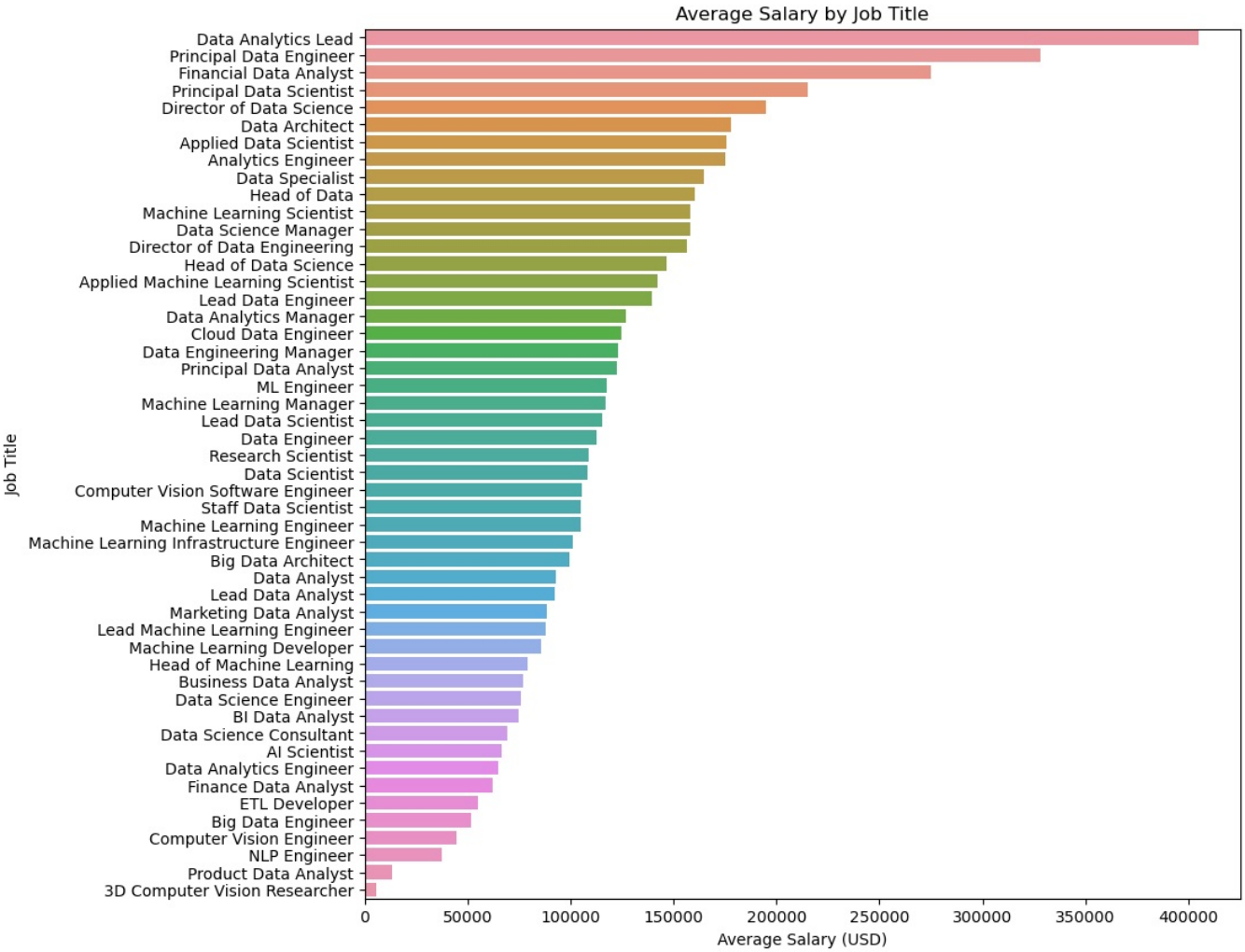
In [8]:
```
print(df.describe())
```

```
            S. no    work_year        salary  salary_in_usd  remote_ratio
count  607.000000   607.000000  6.070000e+02     607.000000    607.00000
mean   304.000000  2021.405272  3.240001e+05  112297.869852     70.92257
std    175.370085     0.692133  1.544357e+06   70957.259411     40.70913
min      1.000000  2020.000000  4.000000e+03    2859.000000      0.00000
25%    152.500000  2021.000000  7.000000e+04   62726.000000     50.00000
50%    304.000000  2022.000000  1.150000e+05  101570.000000    100.00000
75%    455.500000  2022.000000  1.650000e+05  150000.000000    100.00000
max    607.000000  2022.000000  3.040000e+07  600000.000000    100.00000
```

In [39]:
```
print(df.isnull().sum())
```

```
S. no                 0
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 [5]:
```
#analyze the average salary by job title.
avg_salary_by_job_title = df.groupby('job_title')['salary_in_usd'].mean().sort_values(ascending=False)

# Plotting the average salary by job title
plt.figure(figsize=(10, 10))
sns.barplot(x=avg_salary_by_job_title.values, y=avg_salary_by_job_title.index)
plt.xlabel('Average Salary (USD)')
plt.ylabel('Job Title')
plt.title('Average Salary by Job Title')
plt.show()
```



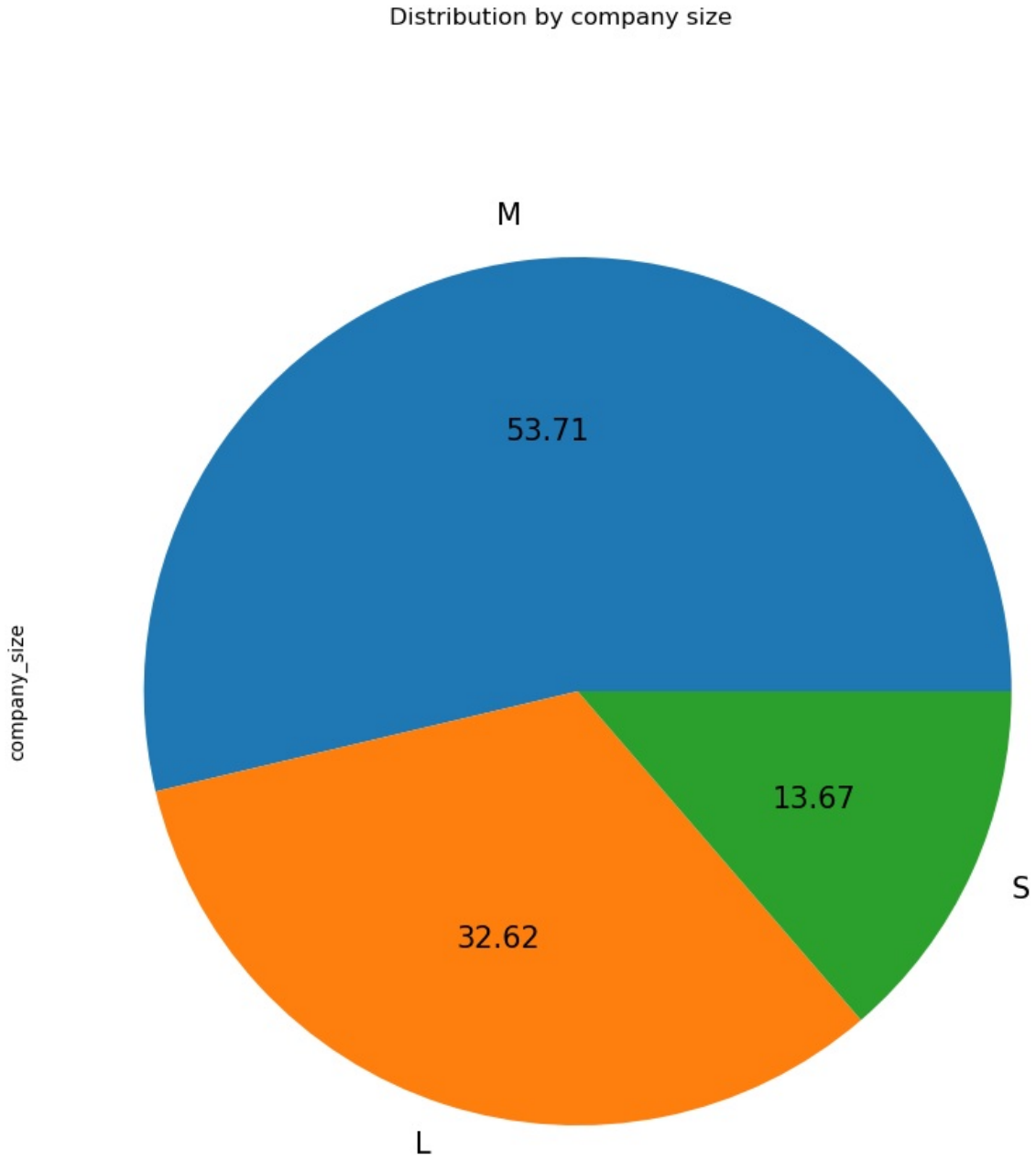
In [25]:
```
df['job_title'].value_counts()
```

Out[25]:
```
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
Principal Data Scientist                      7
AI Scientist                                  7
Data Science Consultant                       7
Director of Data Science                      7
Data Analytics Manager                        7
ML Engineer                                   6
Computer Vision Engineer                      6
BI Data Analyst                               6
Lead Data Engineer                            6
Data Engineering Manager                      5
Business Data Analyst                         5
Head of Data                                  5
Applied Data Scientist                        5
Applied Machine Learning Scientist            4
Head of Data Science                          4
Analytics Engineer                            4
Data Analytics Engineer                       4
Machine Learning Developer                    3
Machine Learning Infrastructure Engineer      3
Lead Data Scientist                           3
Computer Vision Software Engineer             3
Lead Data Analyst                             3
Data Science Engineer                         3
Principal Data Engineer                       3
Principal Data Analyst                        2
ETL Developer                                 2
Product Data Analyst                          2
Director of Data Engineering                  2
Financial Data Analyst                        2
Cloud Data Engineer                           2
Lead Machine Learning Engineer                1
NLP Engineer                                  1
Head of Machine Learning                      1
3D Computer Vision Researcher                 1
Data Specialist                               1
Staff Data Scientist                          1
Big Data Architect                            1
Finance Data Analyst                          1
Marketing Data Analyst                        1
Machine Learning Manager                      1
Data Analytics Lead                           1
Name: job_title, dtype: int64
```

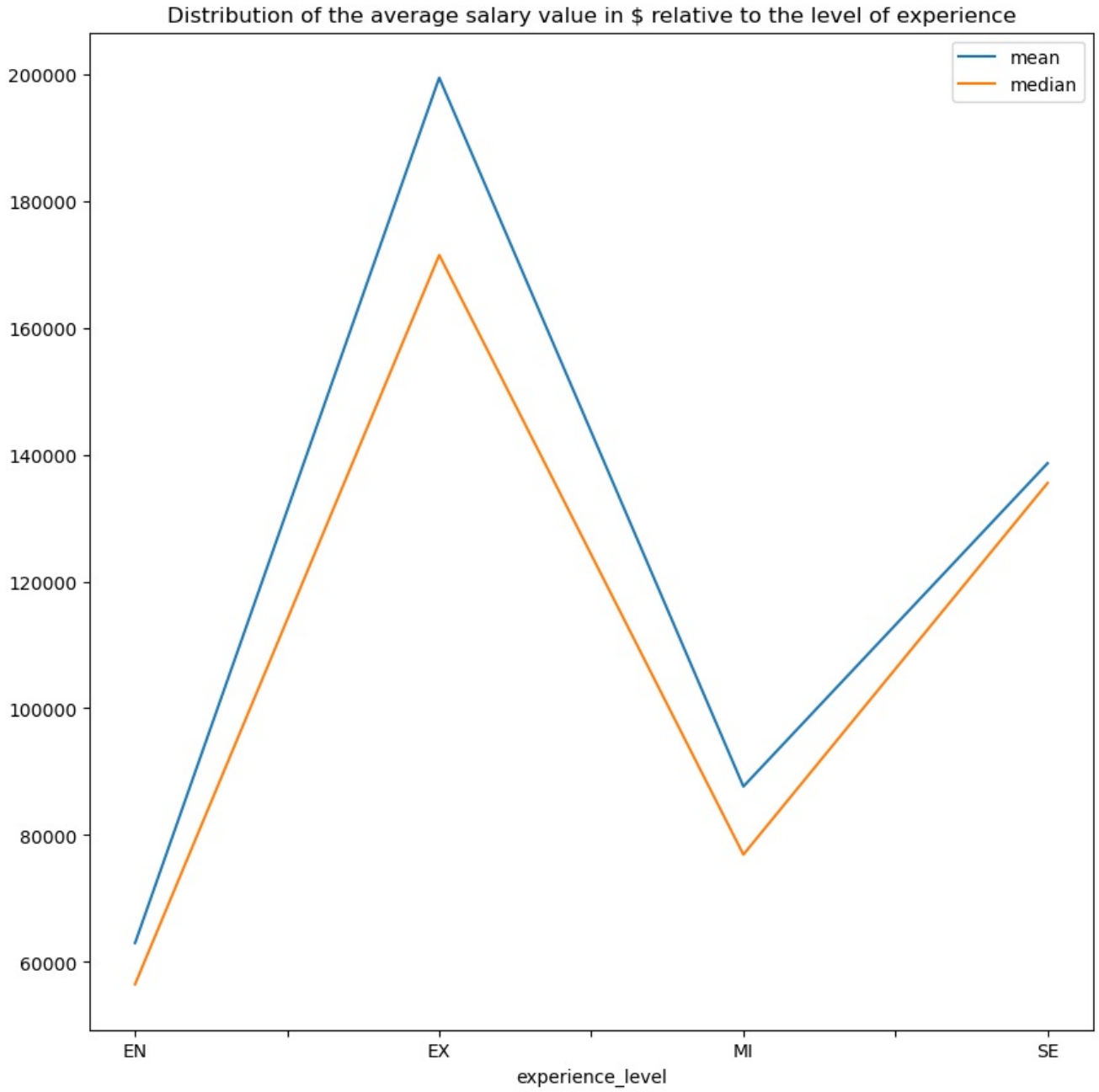
In [26]:
```
df['salary_currency'].value_counts()
```

Out[26]:
```
USD    398
EUR     95
GBP     44
INR     27
CAD     18
JPY      3
PLN      3
TRY      3
CNY      2
MXN      2
HUF      2
DKK      2
SGD      2
BRL      2
AUD      2
CLP      1
CHF      1
Name: salary_currency, dtype: int64
```

In [27]:
```
df['company_size'].value_counts().plot(kind='pie', subplots=True, autopct='%1.2f', figsize=(10,10), title='Dist
plt.show()
```



In [28]:
```
df.groupby('experience_level')['salary_in_usd'].agg(['mean','median']).plot(kind='line', figsize=(10,10), title
plt.show()
```



In [29]:
```
#for all jobs
fig, ax = plt.subplots()
ax= sns.lineplot(df, x='work_year', y='salary_in_usd', errorbar=None)
plt.xticks(ticks=[2020,2021,2022])
ax.set_title('Salary in USD by year')
ax.set_xlabel('Work Year')
ax.set_ylabel('Salary in USD')
```

Out[29]:
```
Text(0, 0.5, 'Salary in USD')
```

In [83]:
```
df.pivot_table(index='experience_level',columns='company_size',values='salary_in_usd',aggfunc={'salary_in_usd':
```

Out[83]:

company_size	L	M	S
experience_level			
EN	72813.241379	50321.800000	62185.310345
EX	221942.181818	178241.750000	201309.333333
MI	98030.372093	90091.081633	51159.379310
SE	147591.013889	137815.596774	116026.727273

In [194…
```
#highly paid jobs of 2020
salary_2020=df[df['work_year']==2020]
salary_2020.sort_values(by='salary_in_usd',ascending=False).head(3)
```

Out[194]:

	S. no	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_rati
33	34	2020	MI	FT	Research Scientist	450000.0	USD	450000	US	
63	64	2020	SE	FT	Data Scientist	412000.0	USD	412000	US	10
25	26	2020	EX	FT	Director of Data Science	325000.0	USD	325000	US	10

In [195…
```
#highly paid jobs of 2021
salary_2021=df[df['work_year']==2021]
salary_2021.sort_values(by='salary_in_usd',ascending=False).head(3)
```

Out[195]:

	S. no	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ra
252	253	2021	EX	FT	Principal Data Engineer	600000.0	USD	600000	US	1
97	98	2021	MI	FT	Financial Data Analyst	450000.0	USD	450000	US	1
157	158	2021	MI	FT	Applied Machine Learning Scientist	423000.0	USD	423000	US	

In [196…

```
#highly paid jobs of 2022
salary_2022=df[df['work_year']==2022]
salary_2022.sort_values(by='salary_in_usd',ascending=False).head(3)
```

Out[196]:

	S. no	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ra
523	524	2022	SE	FT	Data Analytics Lead	405000.0	USD	405000	US	1
519	520	2022	SE	FT	Applied Data Scientist	380000.0	USD	380000	US	1
482	483	2022	EX	FT	Data Engineer	324000.0	USD	324000	US	1

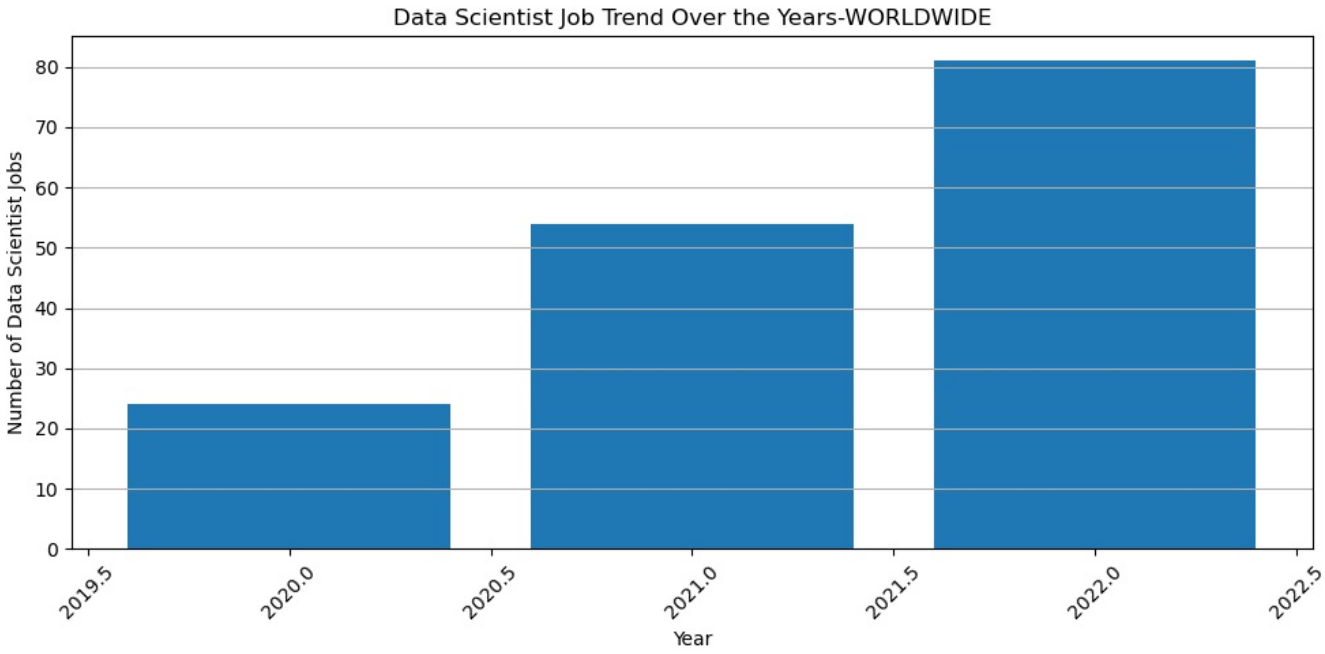
In [4]:

```
#DATA SCIENTIST JOBS IN EACH YEAR wordlwide
data_scientist_jobs = df[df['job_title'].str.contains('data scientist', case=False)]

#Extract the year from the 'work_year' column (assuming 'work_year' is in the format 'YYYY')
data_scientist_jobs['work_year'] = pd.to_datetime(data_scientist_jobs['work_year'], format='%Y').dt.year

#Group the data by year and count the number of data scientist jobs in each year
data_scientist_counts_by_year = data_scientist_jobs['work_year'].value_counts()

print("Data Scientist Jobs in each work year:")
print(data_scientist_counts_by_year)
```

```
Data Scientist Jobs in each work year:
2022    81
2021    54
2020    24
Name: work_year, dtype: int64
C:\Users\91875\AppData\Local\Temp\ipykernel_15680\1248805531.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  data_scientist_jobs['work_year'] = pd.to_datetime(data_scientist_jobs['work_year'], format='%Y').dt.year
```

In [5]:

```
plt.figure(figsize=(10, 5))
plt.bar(data_scientist_counts_by_year.index, data_scientist_counts_by_year.values)
plt.xlabel('Year')
plt.ylabel('Number of Data Scientist Jobs')
plt.title('Data Scientist Job Trend Over the Years-WORLDWIDE')
plt.xticks(rotation=45)
plt.grid(axis='y')
plt.tight_layout()
```



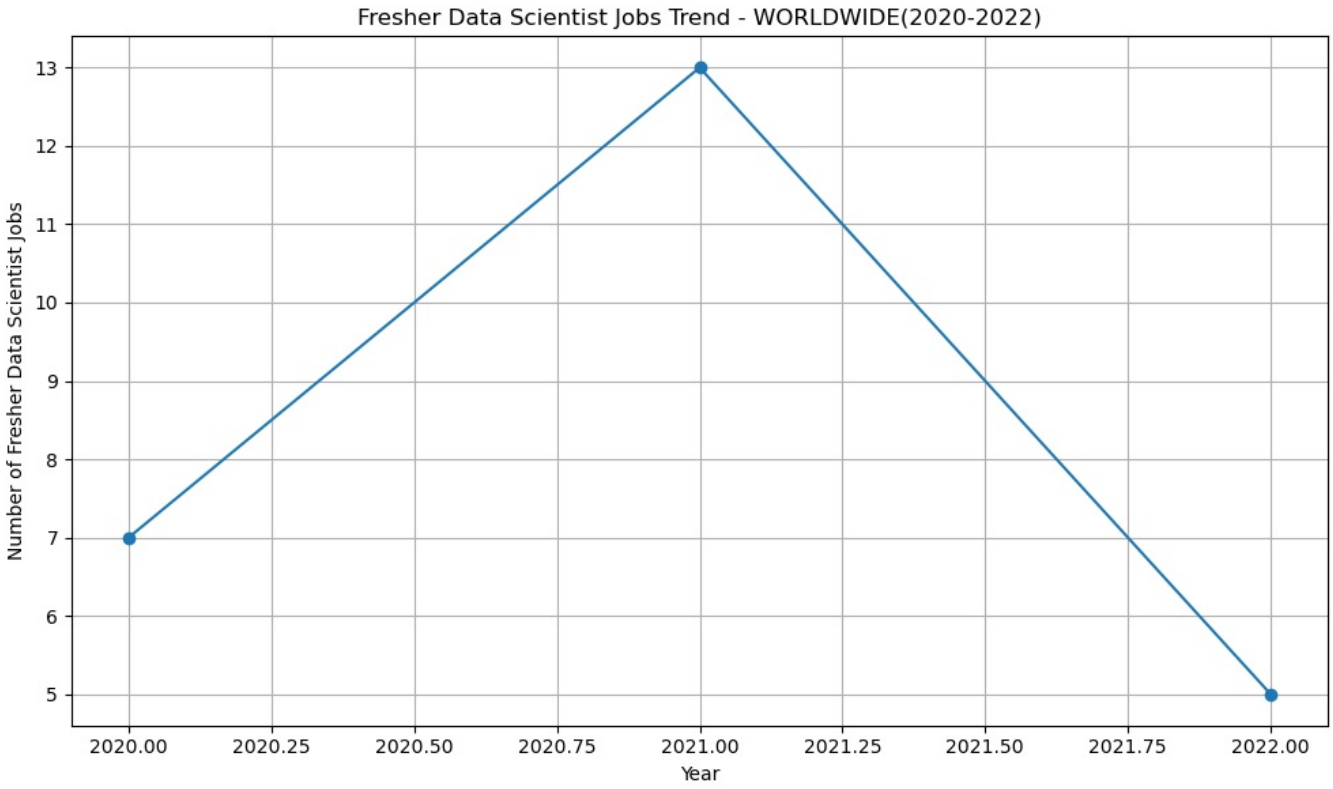
In [150…
```
#WORLDWIDE SALARY OF A DATA SCIENTIST
average_salary_data_scientist = data_scientist_jobs['salary_in_usd'].mean()
average_salary_data_scientist
```

Out[150]: 115134.60377358491

In [10]:
```
data_scientist_en_exp = data_scientist_jobs[data_scientist_jobs['experience_level'].str.contains('EN', case=Fal

data_scientist_en_exp_counts_by_year = data_scientist_en_exp['work_year'].value_counts()

print("Data Scientist Jobs Analysis:")
print("Data Scientist Jobs with 'EN' in Experience Level in each year:")
print(data_scientist_en_exp_counts_by_year)
```

```
Data Scientist Jobs Analysis:
Data Scientist Jobs with 'EN' in Experience Level in each year:
2021    13
2020     7
2022     5
Name: work_year, dtype: int64
```

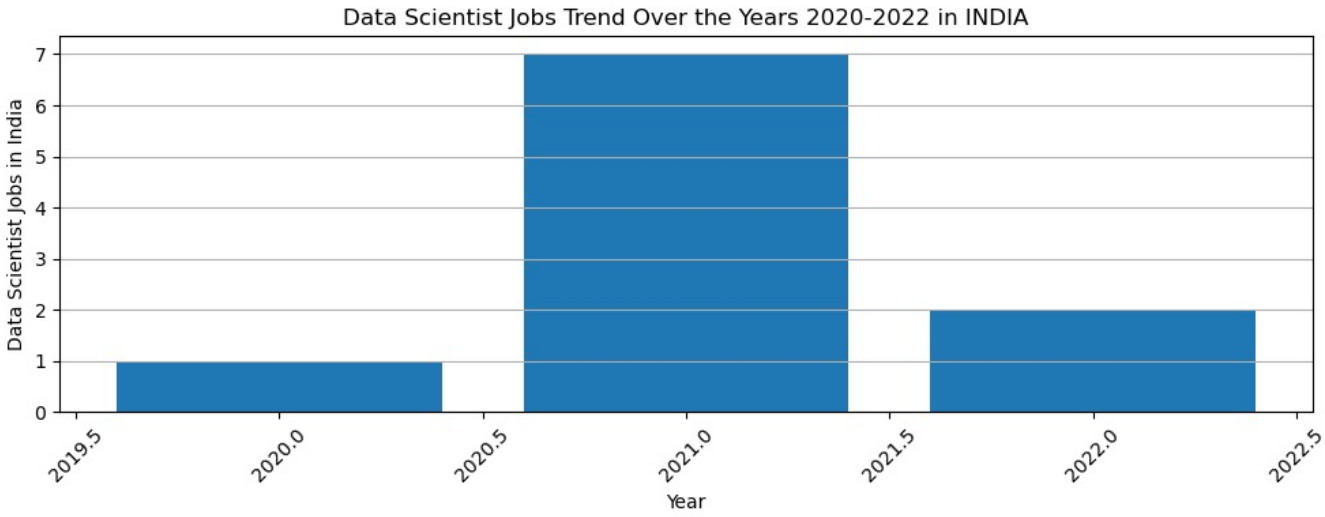
In [8]:
```
data_scientist_en_exp = data_scientist_jobs[data_scientist_jobs['experience_level'].str.contains('EN', case=Fal

data_scientist_en_exp_counts_by_year = data_scientist_en_exp['work_year'].value_counts().sort_index()


plt.figure(figsize=(10, 6))
plt.plot(data_scientist_en_exp_counts_by_year.index,data_scientist_en_exp_counts_by_year.values, marker='o')
plt.xlabel('Year')
plt.ylabel('Number of Fresher Data Scientist Jobs')
plt.title('Fresher Data Scientist Jobs Trend - WORLDWIDE(2020-2022)')
plt.grid(True)
plt.tight_layout()

plt.show()
```



In [111…

```
#JOBS AS A DATA SCIENTIST IN INDIA
#Filter the data for "Data Scientist" job titles with "IN" in location
data_scientist_en_exp_in_location = df[(df['job_title'].str.contains('Data Scientist', case=False)) &
                                       (df['employee_residence'].str.contains('IN', case=False))]

data_scientist_en_exp_in_location['work_year'] = pd.to_datetime(data_scientist_en_exp_in_location['work_year'],

pattern_job_counts_by_year = data_scientist_en_exp_in_location['work_year'].value_counts()

print("Data Scientist Jobs Analysis for'IN' Location:")
print("Data Scientist Jobs with IN' Location in each year:")
print(pattern_job_counts_by_year)
plt.figure(figsize=(10, 4))
plt.bar(pattern_job_counts_by_year.index, pattern_job_counts_by_year.values)
plt.xlabel('Year')
plt.ylabel('Data Scientist Jobs in India')
plt.title('Data Scientist Jobs Trend Over the Years 2020-2022 in INDIA')
plt.xticks(rotation=45)
plt.grid(axis='y')
plt.tight_layout()
```

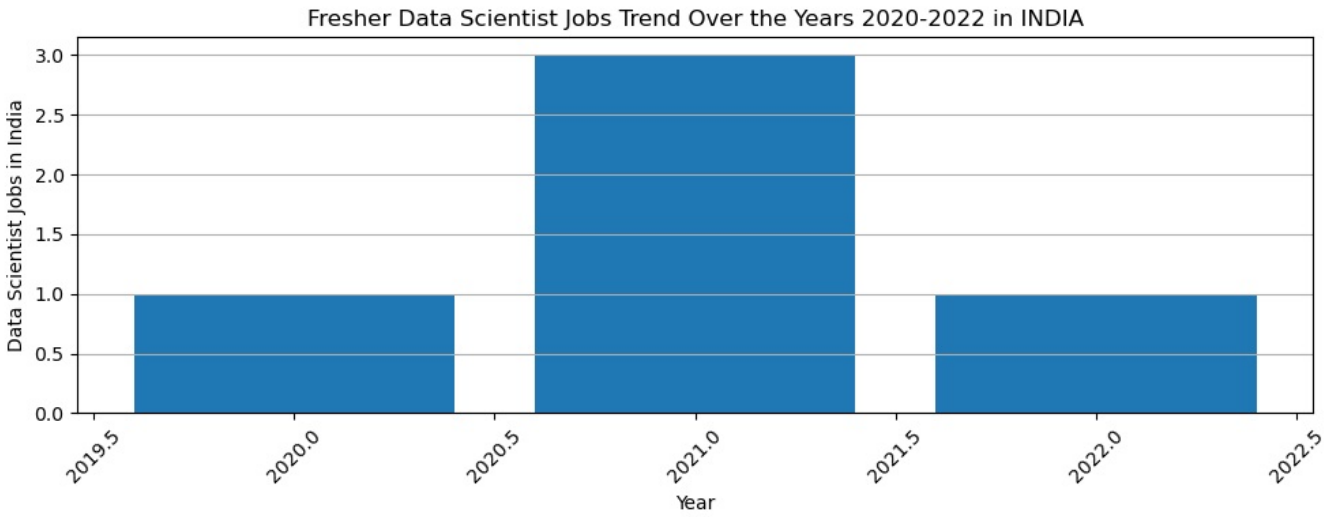
```
C:\Users\91875\AppData\Local\Temp\ipykernel_19608\3912261431.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  data_scientist_en_exp_in_location['work_year'] = pd.to_datetime(data_scientist_en_exp_in_location['work_year'
], format='%Y').dt.year
Data Scientist Jobs Analysis for'IN' Location:
Data Scientist Jobs with IN' Location in each year:
2021    7
2022    2
2020    1
Name: work_year, dtype: int64
```



In [9]:

```
#FRESHER JOBS AS A DATA SCIENTIST IN INDIA
#Filter the data for "Data Scientist" job titles with "EN" in experience level and "IN" in location
data_scientist_en_exp_in_location = df[(df['job_title'].str.contains('Data Scientist', case=False)) &
                                       (df['experience_level'].str.contains('EN', case=False)) &
                                       (df['employee_residence'].str.contains('IN', case=False))]

data_scientist_en_exp_in_location['work_year'] = pd.to_datetime(data_scientist_en_exp_in_location['work_year'],
pattern_job_counts_by_year = data_scientist_en_exp_in_location['work_year'].value_counts()
print("Data Scientist Jobs Analysis for 'EN' Experience Level and 'IN' Location:")
print("Data Scientist Jobs with 'EN' Experience Level and 'IN' Location in each year:")
print(pattern_job_counts_by_year)
plt.figure(figsize=(10, 4))
plt.bar(pattern_job_counts_by_year.index, pattern_job_counts_by_year.values)
plt.xlabel('Year')
plt.ylabel('Data Scientist Jobs in India')
plt.title('Fresher Data Scientist Jobs Trend Over the Years 2020-2022 in INDIA')
plt.xticks(rotation=45)
plt.grid(axis='y')
plt.tight_layout()
```

```
C:\Users\91875\AppData\Local\Temp\ipykernel_15680\2220564537.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  data_scientist_en_exp_in_location['work_year'] = pd.to_datetime(data_scientist_en_exp_in_location['work_year'
], format='%Y').dt.year
Data Scientist Jobs Analysis for 'EN' Experience Level and 'IN' Location:
Data Scientist Jobs with 'EN' Experience Level and 'IN' Location in each year:
2021    3
2020    1
2022    1
Name: work_year, dtype: int64
```



In [168…

```
#AVG salary as a Entry level data scientist in INDIA
EN_data_scientist_in_india = df[(df['job_title'].str.contains('Data Scientist', case=False)) &

                                (df['employee_residence'].str.contains('IN', case=False)) &
                            (df['experience_level'].str.contains('EN', case=False))]

EN_data_scientist_in_india['work_year'] = pd.to_datetime(EN_data_scientist_in_india['work_year'], format='%Y').


EN_data_scientist_in_india['salary_in_inr'] = EN_data_scientist_in_india['salary_in_usd']*82.83
average_salary_by_year = EN_data_scientist_in_india.groupby('work_year')['salary_in_inr'].mean()

print(average_salary_by_year)
```

```
work_year
2020    3353041.23
2021    1867816.50
2022    1527550.86
Name: salary_in_inr, dtype: float64
```

```
C:\Users\91875\AppData\Local\Temp\ipykernel_19608\3897211454.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  EN_data_scientist_in_india['work_year'] = pd.to_datetime(EN_data_scientist_in_india['work_year'], format='%Y'
).dt.year
C:\Users\91875\AppData\Local\Temp\ipykernel_19608\3897211454.py:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  EN_data_scientist_in_india['salary_in_inr'] = EN_data_scientist_in_india['salary_in_usd']*82.83
```

In [ ]: