Data Science Job Salaries Dataset

September 12, 2023

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
df = pd.read_csv('ds_salaries - ds_salaries.csv')
df.tail(10)
```

[2]:

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
597	597	2022	SE	FT	Data Analyst	170000	USD	170000	US	100	US	M
598	598	2022	MI	FT	Data Scientist	160000	USD	160000	US	100	US	M
599	599	2022	MI	FT	Data Scientist	130000	USD	130000	US	100	US	M
600	600	2022	EN	FT	Data Analyst	67000	USD	67000	CA	0	CA	M
601	601	2022	EN	FT	Data Analyst	52000	USD	52000	CA	0	CA	M
602	602	2022	SE	FT	Data Engineer	154000	USD	154000	US	100		
603	603	2022	SE	FT	Data Engineer	126000	USD	126000	US	100		
604	604	2022	SE	FT	Data Analyst	129000	USD	129000	US	0		
605	605	2022	SE	FT	Data Analyst	150000	USD	150000	US	100		
606	606	2022	MI	FT	AI Scientist	200000	USD	200000	IN	100		

```
602             US          M
603             US          M
604             US          M
605             US          M
606             US          L
```

0.0.1 Feature Description

Column : Description

Work_year: The year the salary was paid.

Experience_level : The experience level in the job during the year with the following-possible values: EN Entry-level / Junior MI Mid-level / Intermediate SE Senior-level / Expert EX Executive-level / Director

Employment_type : The type of employement for the role: PT Part-time, FT Full-time, CT Contract FL Freelance

Job_title : The role worked in during the year.

Salary : The total gross salary amount paid.

Salary_currency : The currency of the salary paid as an ISO 4217 currency code.

Salaryinusd : The salary in USD (FX rate divided by avg. USD rate for the respective year via fxdata.foorilla.com).

Employee_residence : Employee's primary country of residence in during the work year as an ISO 3166 country code.

Remote_ratio :The overall amount of work done remotely, possible values are as follows: 0 No remote work (less than 20%) 50 Partially remote 100 Fully remote (more than 80%)

Company_location :The country of the employer's main office or contracting branch as an ISO 3166 country code.

Company_size :The average number of people that worked for the company during the year: S less than 50 employees (small) M 50 to 250 employees (medium) L more than 250 employees (large)

[3]:
```
df.shape
```

[3]: (607, 12)

[4]:
```
# Calculate descriptive statistics of the salary column
statistics = df['salary_in_usd'].describe()
print(statistics)
```

```
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
```

[5]:
```
df.describe()
```

[5]:

	Unnamed: 0	work_year	salary	salary_in_usd	remote_ratio
count	607.000000	607.000000	6.070000e+02	607.000000	607.00000
mean	303.000000	2021.405272	3.240001e+05	112297.869852	70.92257
std	175.370085	0.692133	1.544357e+06	70957.259411	40.70913
min	0.000000	2020.000000	4.000000e+03	2859.000000	0.00000
25%	151.500000	2021.000000	7.000000e+04	62726.000000	50.00000
50%	303.000000	2022.000000	1.150000e+05	101570.000000	100.00000
75%	454.500000	2022.000000	1.650000e+05	150000.000000	100.00000
max	606.000000	2022.000000	3.040000e+07	600000.000000	100.00000

[6]:
```
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
```

[7]:
```
print(df.isnull().sum())
```

```
Unnamed: 0            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
```

[8]:
```
column_name = 'Unnamed: 0'
df = df.drop(column_name, axis=1)
```

[9]:
```
df.head()
```

[9]:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	2020	MI	FT	Data Scientist	70000	EUR	79833	DE	0	DE	L
1	2020	SE	FT	Machine Learning Scientist	260000	USD	260000	JP	0	JP	S
2	2020	SE	FT	Big Data Engineer	85000	GBP	109024	GB	50	GB	M
3	2020	MI	FT	Product Data Analyst	20000	USD	20000	HN	0	HN	S
4	2020	SE	FT	Machine Learning Engineer	150000	USD	150000	US	50	US	L

[10]:
```
column_name = 'employee_residence'
num_features = df[column_name].unique()

print("Number of unique features in column {}: {}".format(column_name,␣
↪num_features))
```

```
Number of unique features in column employee_residence: ['DE' 'JP' 'GB' 'HN'
'US' 'HU' 'NZ' 'FR' 'IN' 'PK' 'PL' 'PT' 'CN' 'GR'
 'AE' 'NL' 'MX' 'CA' 'AT' 'NG' 'PH' 'ES' 'DK' 'RU' 'IT' 'HR' 'BG' 'SG'
 'BR' 'IQ' 'VN' 'BE' 'UA' 'MT' 'CL' 'RO' 'IR' 'CO' 'MD' 'KE' 'SI' 'HK'
 'TR' 'RS' 'PR' 'LU' 'JE' 'CZ' 'AR' 'DZ' 'TN' 'MY' 'EE' 'AU' 'BO' 'IE'
 'CH']
```

[11]:
```
pip install pycountry
```

```
Requirement already satisfied: pycountry in c:\users\mainoot\anaconda3\lib\site-
packages (22.3.5)Note: you may need to restart the kernel to use updated
packages.

Requirement already satisfied: setuptools in
c:\users\mainoot\anaconda3\lib\site-packages (from pycountry) (63.4.1)
```

[12]:
```
import pycountry
```

[13]:
```
abbreviations = ['DE', 'JP', 'GB', 'HN', 'US', 'HU', 'NZ', 'FR', 'IN', 'PK',␣
 ↪'PL', 'PT', 'CN', 'GR', 'AE', 'NL', 'MX', 'CA', 'AT', 'NG', 'PH', 'ES',␣
 ↪'DK', 'RU', 'IT', 'HR', 'BG', 'SG', 'BR', 'IQ', 'VN', 'BE', 'UA', 'MT',␣
 ↪'CL', 'RO', 'IR', 'CO', 'MD', 'KE', 'SI', 'HK', 'TR', 'RS', 'PR', 'LU',␣
 ↪'JE', 'CZ', 'AR', 'DZ', 'TN', 'MY', 'EE', 'AU', 'BO', 'IE', 'CH']

# Iterate over each abbreviation
for abbreviation in abbreviations:
    try:
        # Get the country object from the abbreviation
        country_obj = pycountry.countries.get(alpha_2=abbreviation)

        # Check if the country object is found
        if country_obj is not None:
            # Access the country name
            country_name = country_obj.name
            print("Abbreviation:", abbreviation, "Country Name:", country_name)
        else:
            print("Country not found for the abbreviation:", abbreviation)
    except LookupError:
        print("Error: Lookup failed for abbreviation:", abbreviation)
```

```
Abbreviation: DE Country Name: Germany
Abbreviation: JP Country Name: Japan
Abbreviation: GB Country Name: United Kingdom
Abbreviation: HN Country Name: Honduras
Abbreviation: US Country Name: United States
Abbreviation: HU Country Name: Hungary
Abbreviation: NZ Country Name: New Zealand
Abbreviation: FR Country Name: France
Abbreviation: IN Country Name: India
```

Abbreviation: PK Country Name: Pakistan
Abbreviation: PL Country Name: Poland
Abbreviation: PT Country Name: Portugal
Abbreviation: CN Country Name: China
Abbreviation: GR Country Name: Greece
Abbreviation: AE Country Name: United Arab Emirates
Abbreviation: NL Country Name: Netherlands
Abbreviation: MX Country Name: Mexico
Abbreviation: CA Country Name: Canada
Abbreviation: AT Country Name: Austria
Abbreviation: NG Country Name: Nigeria
Abbreviation: PH Country Name: Philippines
Abbreviation: ES Country Name: Spain
Abbreviation: DK Country Name: Denmark
Abbreviation: RU Country Name: Russian Federation
Abbreviation: IT Country Name: Italy
Abbreviation: HR Country Name: Croatia
Abbreviation: BG Country Name: Bulgaria
Abbreviation: SG Country Name: Singapore
Abbreviation: BR Country Name: Brazil
Abbreviation: IQ Country Name: Iraq
Abbreviation: VN Country Name: Viet Nam
Abbreviation: BE Country Name: Belgium
Abbreviation: UA Country Name: Ukraine
Abbreviation: MT Country Name: Malta
Abbreviation: CL Country Name: Chile
Abbreviation: RO Country Name: Romania
Abbreviation: IR Country Name: Iran, Islamic Republic of
Abbreviation: CO Country Name: Colombia
Abbreviation: MD Country Name: Moldova, Republic of
Abbreviation: KE Country Name: Kenya
Abbreviation: SI Country Name: Slovenia
Abbreviation: HK Country Name: Hong Kong
Abbreviation: TR Country Name: Turkey
Abbreviation: RS Country Name: Serbia
Abbreviation: PR Country Name: Puerto Rico
Abbreviation: LU Country Name: Luxembourg
Abbreviation: JE Country Name: Jersey
Abbreviation: CZ Country Name: Czechia
Abbreviation: AR Country Name: Argentina
Abbreviation: DZ Country Name: Algeria
Abbreviation: TN Country Name: Tunisia
Abbreviation: MY Country Name: Malaysia
Abbreviation: EE Country Name: Estonia
Abbreviation: AU Country Name: Australia
Abbreviation: BO Country Name: Bolivia, Plurinational State of
Abbreviation: IE Country Name: Ireland
Abbreviation: CH Country Name: Switzerland

[14]:
```
# Specify the column name containing the abbreviations
column_name =  'employee_residence'


# Replace the abbreviations with ISO country names
def replace_abbreviation(x):
    try:
        country = pycountry.countries.get(alpha_2=x)
        if country is not None:
            return country.name
        else:
            return x
    except Exception:
        return x

df[column_name] = df[column_name].apply(lambda x: replace_abbreviation(x))

# Print the updated DataFrame
print(df)
```

```
     work_year experience_level employment_type                   job_title  \
0         2020               MI              FT              Data Scientist
1         2020               SE              FT  Machine Learning Scientist
2         2020               SE              FT           Big Data Engineer
3         2020               MI              FT        Product Data Analyst
4         2020               SE              FT   Machine Learning Engineer
..         …                …               …                          …
602       2022               SE              FT               Data Engineer
603       2022               SE              FT               Data Engineer
604       2022               SE              FT                Data Analyst
605       2022               SE              FT                Data Analyst
606       2022               MI              FT                AI Scientist

     salary salary_currency  salary_in_usd employee_residence  remote_ratio  \
0     70000             EUR          79833            Germany             0
1    260000             USD         260000              Japan             0
2     85000             GBP         109024     United Kingdom            50
3     20000             USD          20000           Honduras             0
4    150000             USD         150000      United States            50
..      …               …             …                 …           …
602  154000             USD         154000      United States           100
603  126000             USD         126000      United States           100
604  129000             USD         129000      United States             0
605  150000             USD         150000      United States           100
606  200000             USD         200000              India           100

    company_location company_size
0                 DE            L
```

```
1                JP            S
2                GB            M
3                HN            S
4                US            L
..              …           …
602              US            M
603              US            M
604              US            M
605              US            M
606              US            L

[607 rows x 11 columns]
```

[15]:
```python
# Specify the column name containing the abbreviations
column_names =   'company_location'


# Replace the abbreviations with ISO country names
def replace_abbreviation(x):
    try:
        country = pycountry.countries.get(alpha_2=x)
        if country is not None:
            return country.name
        else:
            return x
    except Exception:
        return x

df[column_names] = df[column_names].apply(lambda x: replace_abbreviation(x))

# Define the abbreviation-to-word mappings
experience_level_mapping = {
    'MI': 'Mid-level',
    'SE': 'Senior-level',
    'EN': 'Entry-level',
    'EX': 'Executive-level'
}

company_size_mapping = {
    'S': 'Small',
    'M': 'Medium',
    'L': 'Large'
}

employment_type_mapping = {
    'FT': 'Full-Time',
    'PT': 'Part-Time',
```

```
    'CT': 'Contract',
    'FL': 'Freelance'
}


# Replace the abbreviations with full words
df['experience_level'] = df['experience_level'].
 ↪replace(experience_level_mapping)
df['company_size'] = df['company_size'].replace(company_size_mapping)
df['employment_type'] = df['employment_type'].replace(employment_type_mapping)
# Print the updated DataFrame
print(df)

# Print the updated DataFrame
print(df)

print(df)
```

```
     work_year experience_level employment_type                 job_title  \
0         2020        Mid-level       Full-Time              Data Scientist
1         2020     Senior-level       Full-Time  Machine Learning Scientist
2         2020     Senior-level       Full-Time           Big Data Engineer
3         2020        Mid-level       Full-Time        Product Data Analyst
4         2020     Senior-level       Full-Time   Machine Learning Engineer
..         …              …              …                        …
602       2022     Senior-level       Full-Time               Data Engineer
603       2022     Senior-level       Full-Time               Data Engineer
604       2022     Senior-level       Full-Time                Data Analyst
605       2022     Senior-level       Full-Time                Data Analyst
606       2022        Mid-level       Full-Time                AI Scientist

     salary salary_currency  salary_in_usd employee_residence  remote_ratio  \
0     70000             EUR          79833            Germany             0
1    260000             USD         260000              Japan             0
2     85000             GBP         109024     United Kingdom            50
3     20000             USD          20000           Honduras             0
4    150000             USD         150000      United States            50
..      …              …              …                 …            …
602  154000             USD         154000      United States           100
603  126000             USD         126000      United States           100
604  129000             USD         129000      United States             0
605  150000             USD         150000      United States           100
606  200000             USD         200000              India           100

    company_location company_size
0            Germany        Large
1              Japan        Small
2     United Kingdom       Medium
```

```
3           Honduras        Small
4      United States        Large
..               …          …
602    United States       Medium
603    United States       Medium
604    United States       Medium
605    United States       Medium
606    United States        Large

[607 rows x 11 columns]
     work_year experience_level employment_type                   job_title  \
0         2020        Mid-level       Full-Time              Data Scientist
1         2020     Senior-level       Full-Time  Machine Learning Scientist
2         2020     Senior-level       Full-Time           Big Data Engineer
3         2020        Mid-level       Full-Time        Product Data Analyst
4         2020     Senior-level       Full-Time   Machine Learning Engineer
..         …              …             …                         …
602       2022     Senior-level       Full-Time               Data Engineer
603       2022     Senior-level       Full-Time               Data Engineer
604       2022     Senior-level       Full-Time                Data Analyst
605       2022     Senior-level       Full-Time                Data Analyst
606       2022        Mid-level       Full-Time                AI Scientist

     salary salary_currency  salary_in_usd employee_residence  remote_ratio  \
0     70000             EUR          79833            Germany             0
1    260000             USD         260000              Japan             0
2     85000             GBP         109024     United Kingdom            50
3     20000             USD          20000           Honduras             0
4    150000             USD         150000      United States            50
..      …              …             …                …           …
602  154000             USD         154000      United States           100
603  126000             USD         126000      United States           100
604  129000             USD         129000      United States             0
605  150000             USD         150000      United States           100
606  200000             USD         200000              India           100

    company_location company_size
0            Germany        Large
1              Japan        Small
2     United Kingdom       Medium
3           Honduras        Small
4      United States        Large
..               …          …
602    United States       Medium
603    United States       Medium
604    United States       Medium
605    United States       Medium
606    United States        Large
```

```
[607 rows x 11 columns]
     work_year experience_level employment_type                   job_title  \
0         2020        Mid-level       Full-Time              Data Scientist
1         2020     Senior-level       Full-Time  Machine Learning Scientist
2         2020     Senior-level       Full-Time           Big Data Engineer
3         2020        Mid-level       Full-Time        Product Data Analyst
4         2020     Senior-level       Full-Time   Machine Learning Engineer
..         …              …             …                        …
602       2022     Senior-level       Full-Time               Data Engineer
603       2022     Senior-level       Full-Time               Data Engineer
604       2022     Senior-level       Full-Time                Data Analyst
605       2022     Senior-level       Full-Time                Data Analyst
606       2022        Mid-level       Full-Time                AI Scientist

     salary salary_currency  salary_in_usd employee_residence  remote_ratio  \
0     70000             EUR          79833            Germany             0
1    260000             USD         260000              Japan             0
2     85000             GBP         109024     United Kingdom            50
3     20000             USD          20000           Honduras             0
4    150000             USD         150000      United States            50
..      …              …             …                 …           …
602  154000             USD         154000      United States           100
603  126000             USD         126000      United States           100
604  129000             USD         129000      United States             0
605  150000             USD         150000      United States           100
606  200000             USD         200000              India           100

    company_location company_size
0            Germany        Large
1              Japan        Small
2     United Kingdom       Medium
3           Honduras        Small
4      United States        Large
..               …          …
602    United States       Medium
603    United States       Medium
604    United States       Medium
605    United States       Medium
606    United States        Large

[607 rows x 11 columns]
```

[16]:
```
df.head()
```

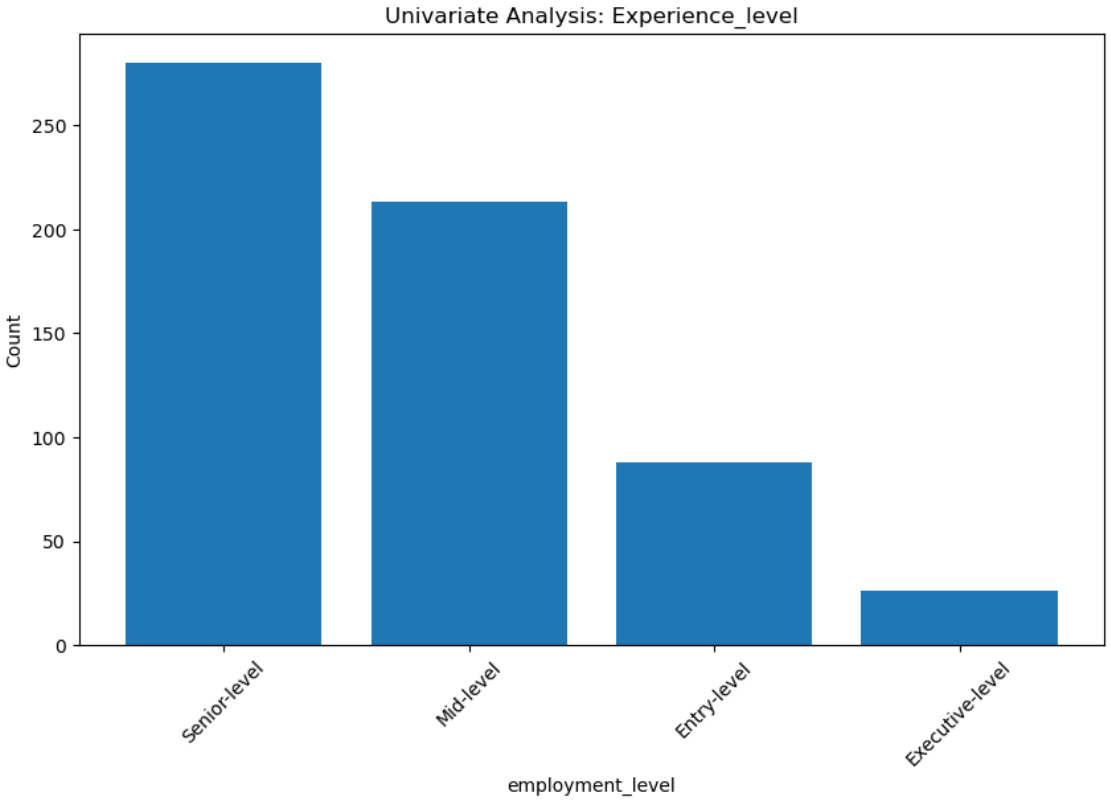
[16]:
```
   work_year experience_level employment_type                 job_title  \
0       2020        Mid-level       Full-Time            Data Scientist
```

```
1       2020     Senior-level       Full-Time  Machine Learning Scientist
2       2020     Senior-level       Full-Time           Big Data Engineer
3       2020        Mid-level       Full-Time        Product Data Analyst
4       2020     Senior-level       Full-Time   Machine Learning Engineer

   salary salary_currency  salary_in_usd employee_residence  remote_ratio  \
0   70000             EUR          79833             Germany             0
1  260000             USD         260000               Japan             0
2   85000             GBP         109024      United Kingdom            50
3   20000             USD          20000            Honduras             0
4  150000             USD         150000       United States            50

  company_location company_size
0          Germany        Large
1            Japan        Small
2   United Kingdom       Medium
3         Honduras        Small
4    United States        Large
```

0.0.2 Exploratery data analysis

Univariate Analysis

[17]:
```
# Perform univariate analysis on 'Employment Level'
employment_level_counts = df['experience_level'].value_counts()
plt.figure(figsize=(10, 6))
plt.bar(employment_level_counts.index, employment_level_counts.values)
plt.title('Univariate Analysis: Experience_level')
plt.xlabel('employment_level')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```

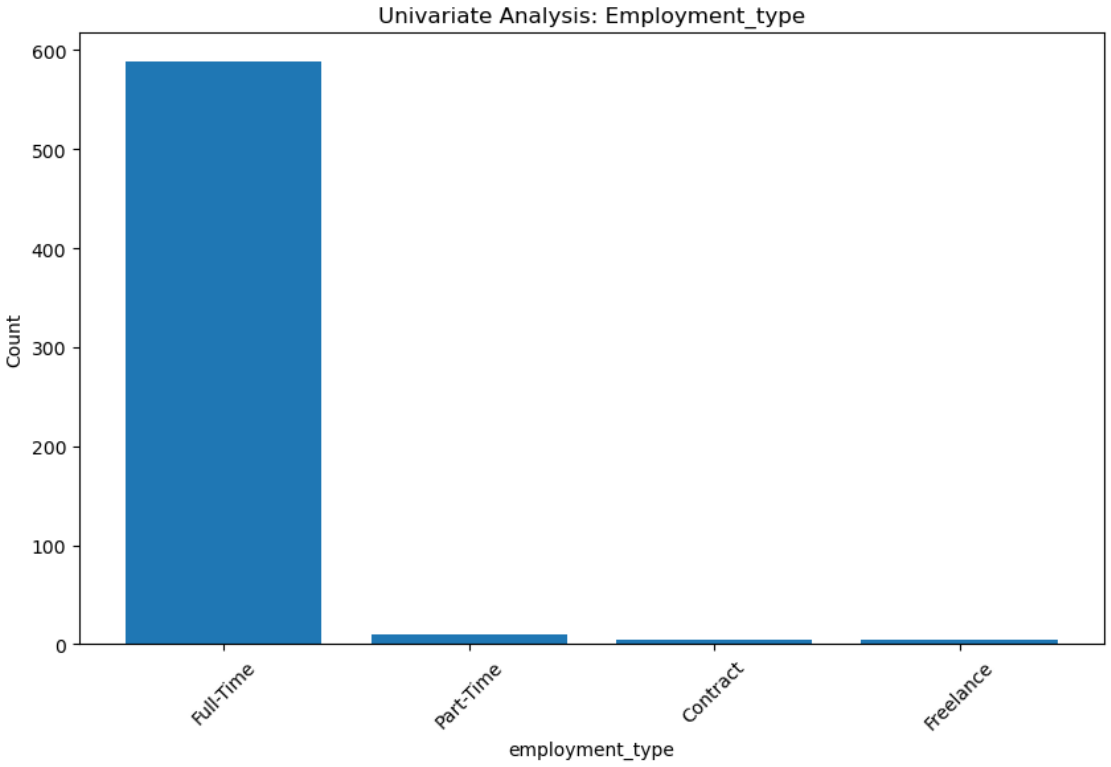


OBSERVATION: From the visuals above for the experience level the Senior level are more prevalent followed by the mid level and the executive level are the lowest.

[ ]:

[ ]:

[18]:
```
# Perform univariate analysis on 'Employment Type'
employment_type_counts = df['employment_type'].value_counts()
plt.figure(figsize=(10, 6))
plt.bar(employment_type_counts.index, employment_type_counts.values)
plt.title('Univariate Analysis: Employment_type')
plt.xlabel('employment_type')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```

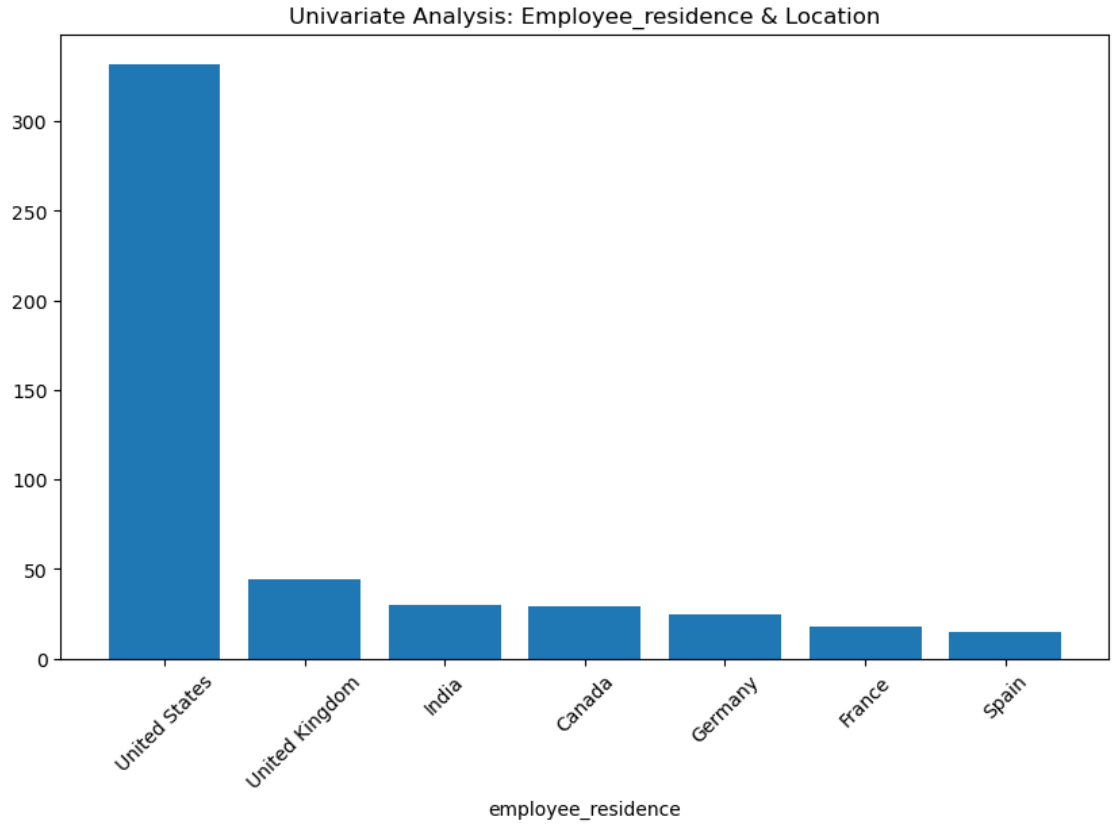


OBSERVATION : The distribution of employment types indicates that the majority of individuals in the dataset are engaged in full-time employment, constituting the highest count. Part-time employment follows as the second most common type, while contract and freelance employment exhibit subsequent counts. This distribution highlights a prevalence of full-time positions and a descending trend in engagement with part-time, contract, and freelance roles

[ ]:

[ ]:

[19]:
```
# Perform univariate analysis on 'Employee_residence'
employee_residenc_counts = df['employee_residence'].value_counts()

employee_residence_counts = employee_residenc_counts.nlargest(7)
plt.figure(figsize=(10, 6))
plt.bar(employee_residence_counts.index, employee_residence_counts.values)
plt.title('Univariate Analysis: Employee_residence & Location')
plt.xlabel('employee_residence')
plt.xticks(rotation=45)
plt.show()
```



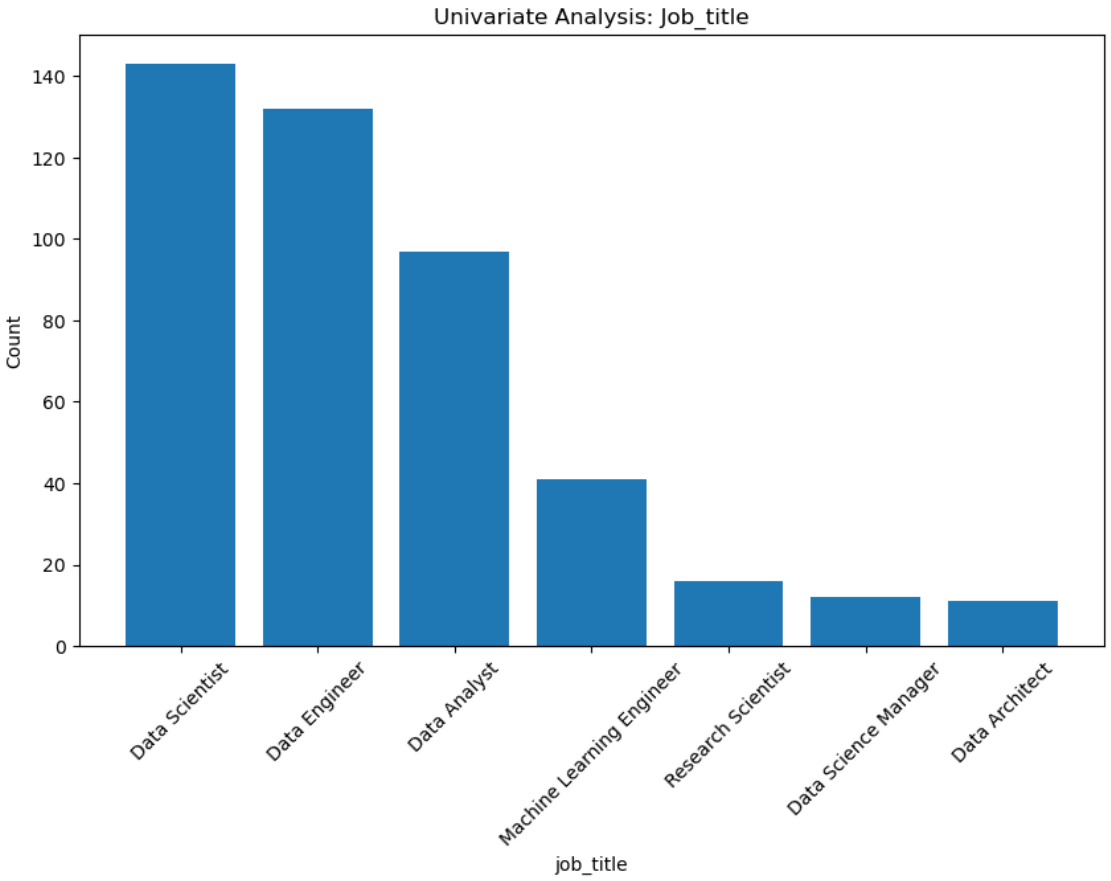
OBSERVATION : The analysis of employment residence and location reveals a clear hierarchy in terms of the number of individuals across various countries. The United States emerges as the predominant residence for the dataset, exhibiting the highest count. Following this, the United Kingdom ranks second in terms of employment residence. Subsequently, individuals from India, Canada, Germany, France, and Spain which are represented, forming the successive tiers in terms of count. This distribution underscores the dominance of the United States and the sequential representation of other countries, highlighting a diverse international presence within the dataset.

[ ]:

[ ]:

[20]:
```
# Perform univariate analysis on 'Job_title'
job_titl_counts = df['job_title'].value_counts()
job_title_counts = job_titl_counts.nlargest(7)
plt.figure(figsize=(10, 6))
plt.bar(job_title_counts.index, job_title_counts.values)
plt.title('Univariate Analysis: Job_title')
plt.xlabel('job_title')
plt.ylabel('Count')
```

```
plt.xticks(rotation=45)
plt.show()
```

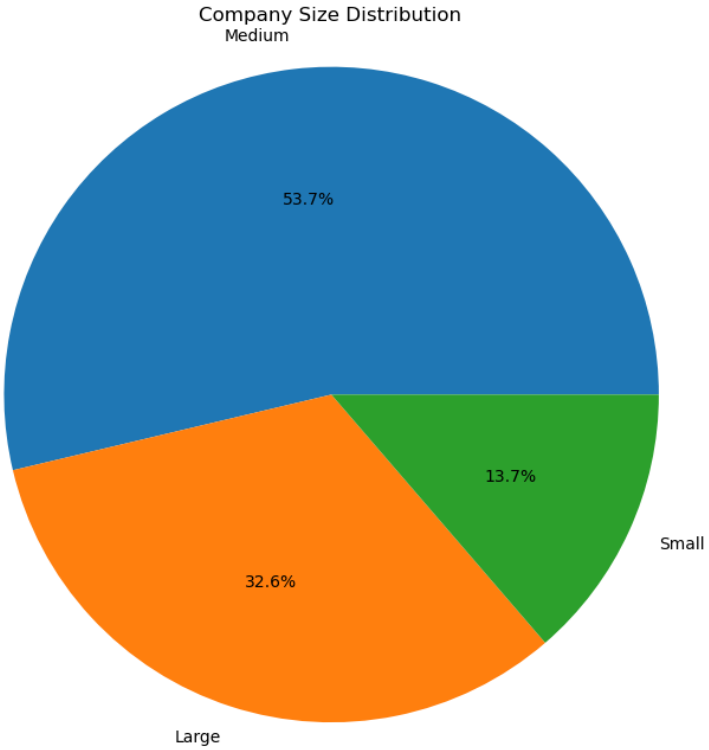


OBSERVATION : The analysis of job titles unveils a distinct hierarchy in terms of the prevalence of different roles within the dataset. Notably, the position of "Data Scientist" stands out as the most common job title, exhibiting the highest count. Following closely, "Data Engineer" emerges as the second most frequently held role, with "Data Analyst" coming in third. The sequence continues with "Machine Learning Engineer," "Research Scientist," "Data Science Manager," and "Data Architect." This distribution illustrates the dominance of "Data Scientist" positions and the subsequent representation of other roles. The prominence of data-centric roles, ranging from data engineering to machine learning and research, underscores the data-driven landscape of the dataset, with diverse specializations contributing to its composition.

[ ]:

[ ]:

[21]:
```
# Calculate the frequency counts of company sizes
company_size_counts = df['company_size'].value_counts()

# Plot a pie chart
plt.figure(figsize=(12, 8))
plt.pie(company_size_counts, labels=company_size_counts.index, autopct='%1.
↪1f%%')
plt.title('Company Size Distribution')
plt.axis('equal')
plt.show()
```

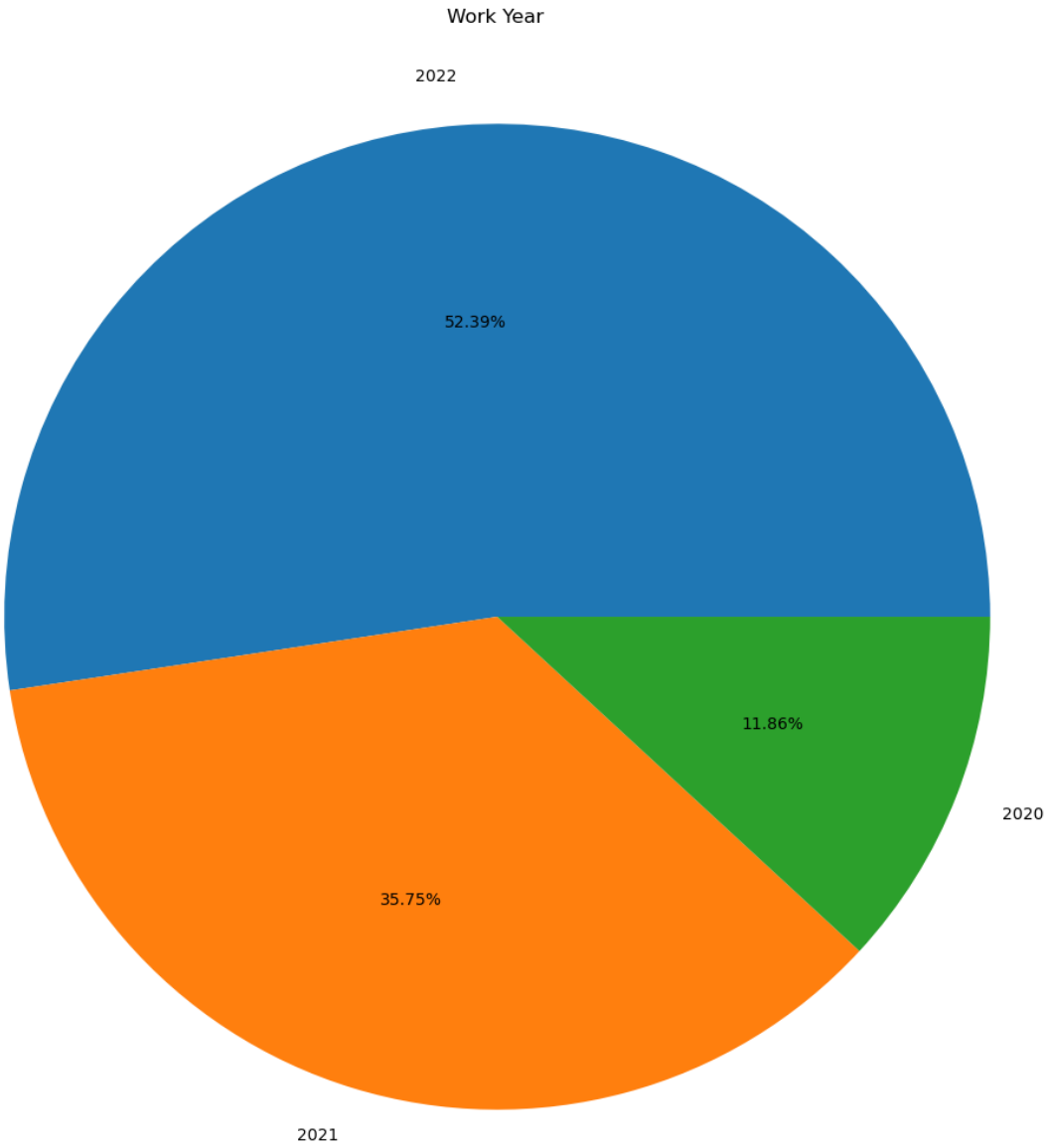


OBSERVATION : Upon analyzing the distribution of company sizes, a discernible pattern emerges, showcasing distinct proportions among different size categories. Notably, "Medium" companies constitute the most prevalent category, representing a substantial percentage of 53.7% within the dataset. Following closely, "Large" companies hold a significant share, accounting for 32.6% of the distribution. Lastly, the category of "Small" companies contributes to the composition with a percentage of 13.7%. This distribution underscores a notable prevalence of medium-sized enterprises, followed by large corporations, while smaller entities round off the distribution. It's apparent that the dataset predominantly comprises medium and large companies, with a modest representation of small companies.

[ ]:

[ ]:

[22]:
```
# Calculate the frequency counts of Work year
work_year = df['work_year'].value_counts()

# Plot a pie chart
plt.figure(figsize=(12, 13))
plt.pie(work_year, labels=work_year.index, autopct='%2.2f%%')
plt.title('Work Year')
plt.axis('equal')
plt.show()
```

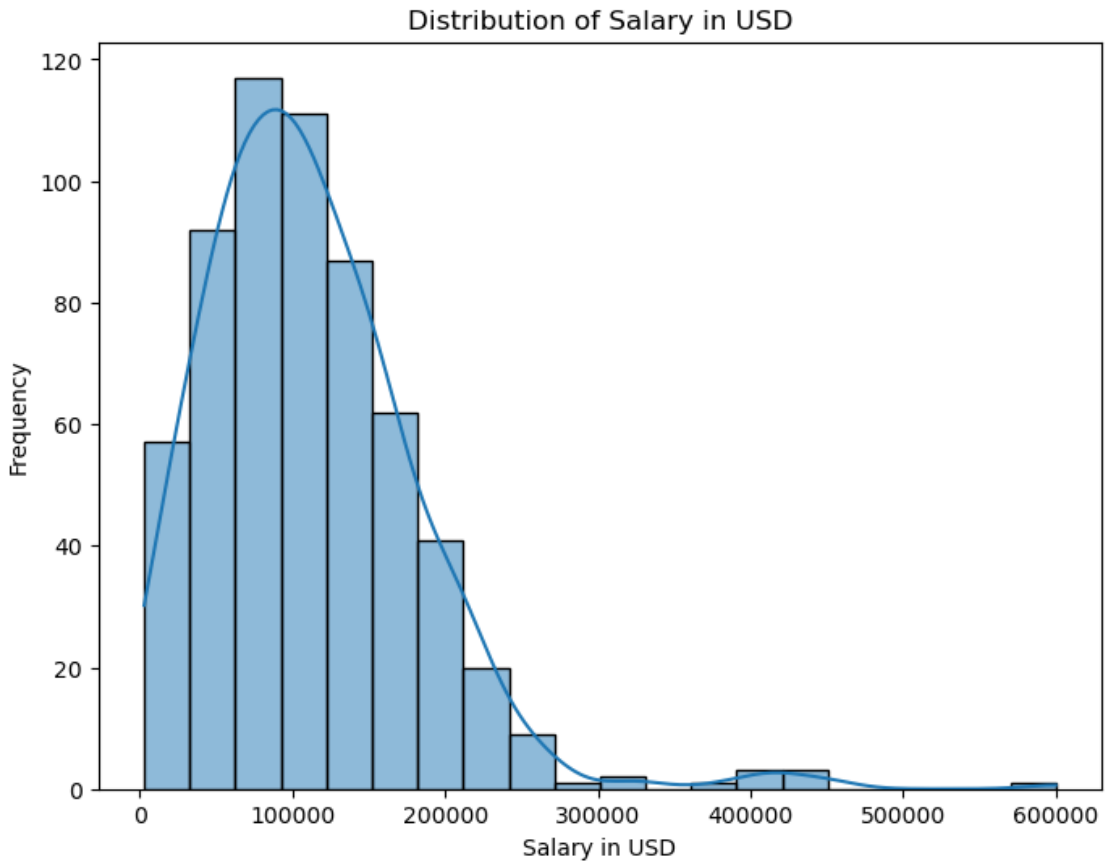


OBSERVATION : Examining the distribution of work years reveals a distinct pattern in the dataset, delineating the prevalence of different work year categories. Notably, the year 2022 emerges as the predominant period, encompassing a substantial percentage of 52.39%. Following closely, the year 2021 captures a significant share, accounting for 35.75% of the distribution. Concluding the spectrum, the year 2020 constitutes the remaining percentage, contributing 11.86% to the distribution. This distribution accentuates the prevalence of work experience garnered in the year 2022, with the year 2021 closely following suit. The representation of work years in the year 2020 underscores a smaller yet noteworthy presence. It is evident that the dataset prominently spans work experiences primarily from the years 2022 and 2021.

[ ]:

[ ]:

[23]:
```
# Histogram
plt.figure(figsize=(8, 6))
sns.histplot(df['salary_in_usd'], bins=20, kde=True)
plt.title('Distribution of Salary in USD')
plt.xlabel('Salary in USD')
plt.ylabel('Frequency')
plt.show()
```



From the histogram diagram it displays a tail that stretches towards the right side. This elongated tail is indicative of a relatively small number of data points with exceptionally high values, which can be considered outliers. These outliers greatly exceed the majority of data points, which are clustered towards the lower end of the distribution. This distribution pattern suggests the presence of a few extreme cases that significantly deviate from the general trend of the data.

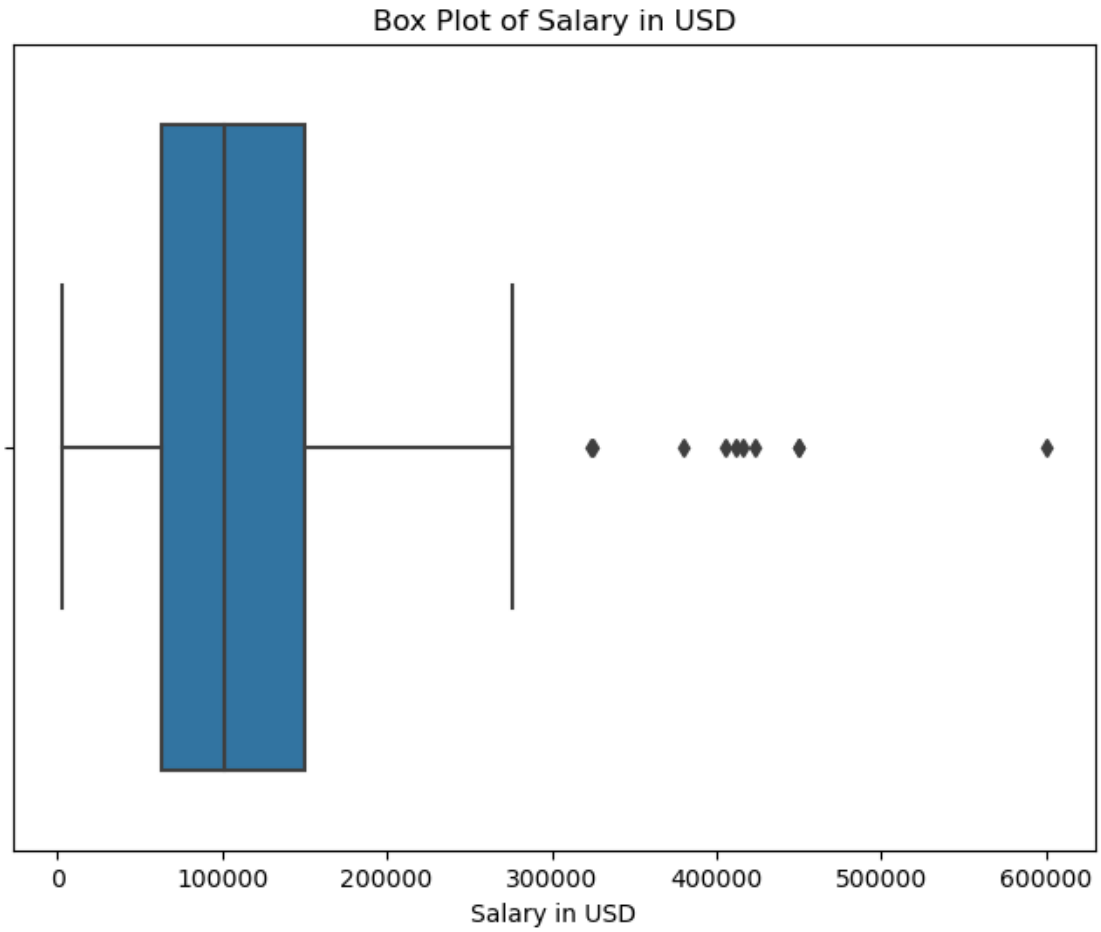
[ ]:

[ ]:

[24]:
```
# Box plot
plt.figure(figsize=(8, 6))
sns.boxplot(df['salary_in_usd'])
plt.title('Box Plot of Salary in USD')
plt.xlabel('Salary in USD')
plt.show()
```

```
C:\Users\MaiNoot\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
```

```
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



OBSERVATION : An interesting observation in the salary distribution is the presence of several outliers in the executive-level and senior level positions. These outliers represent top-tier executives within the organization, occupying roles that require strategic decision-making and leadership. The salaries associated with these positions are significantly higher compared to the rest of the dataset, reflecting the premium placed on experience, expertise, and the responsibilities inherent in executive roles.

[ ]:

[ ]:

[25]:
```
# Group by country and find the maximum salary
max_salary_by_country = df.groupby('employee_residence')['salary_in_usd'].max()

# Find the country with the highest pay
```

```
country_with_highest_pay = max_salary_by_country.idxmax()
highest_pay = max_salary_by_country.max()

print(f"The country with the highest pay is {country_with_highest_pay} with a␣
 ↪salary of ${highest_pay:.2f}.")
```

```
The country with the highest pay is United States with a salary of $600000.00.
```

[26]:
```
min_salary_by_country = df.groupby('employee_residence')['salary_in_usd'].min()
# Find the country with the lowest pay
country_with_lowest_pay = min_salary_by_country.idxmin()
lowest_pay = min_salary_by_country.min()

print(f"The country with the lowest pay is {country_with_lowest_pay} with a␣
 ↪salary of ${lowest_pay:.2f}.")
```

```
The country with the lowest pay is Mexico with a salary of $2859.00.
```

[27]:
```
# Check the number of unique features in the "remote ratio" column
num_features = df['remote_ratio'].nunique()

print(f"The 'remote ratio' column has {num_features} unique features.")
```

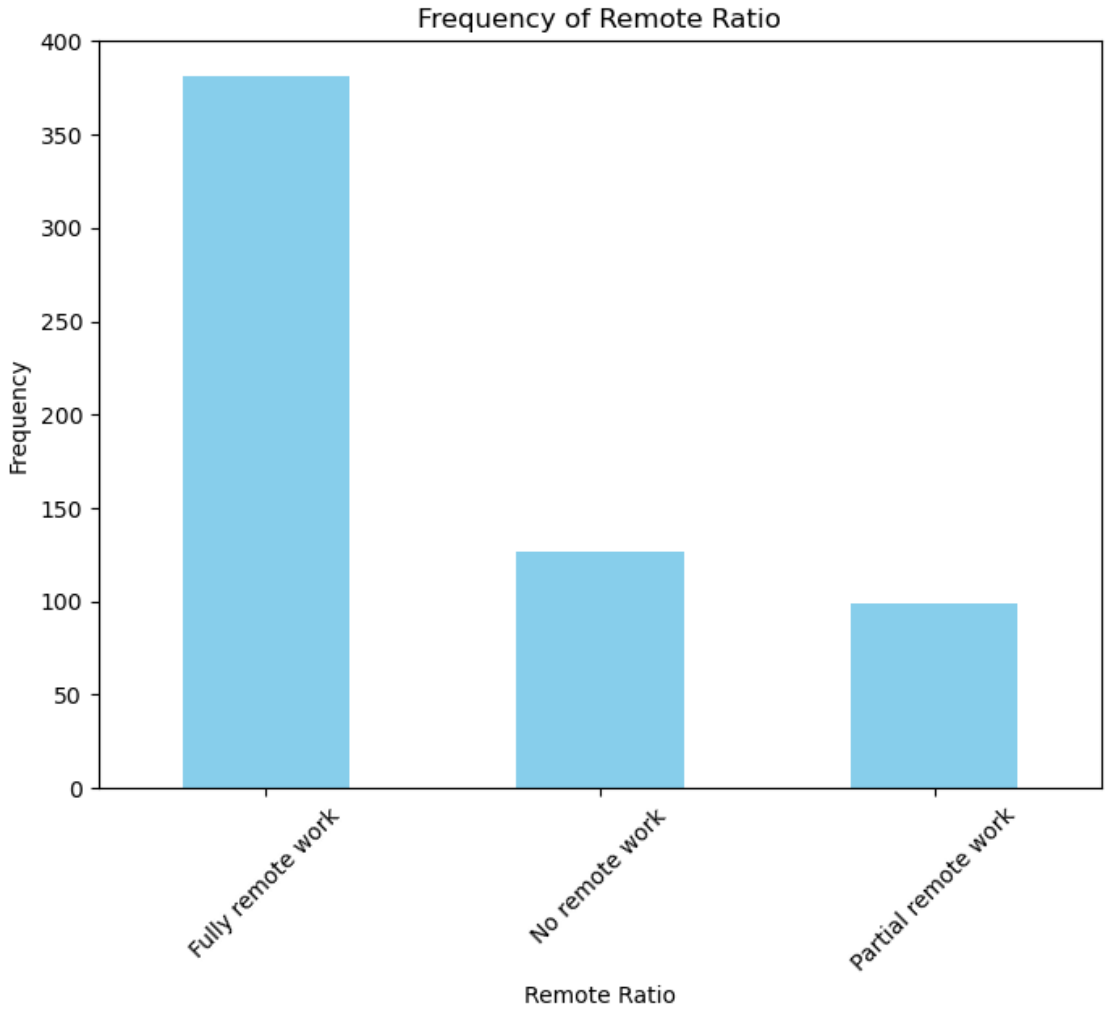
```
The 'remote ratio' column has 3 unique features.
```

[28]:
```
# Define a mapping dictionary for numerical values to words
mapping_dict = {
    0: 'No remote work',
    50: 'Partial remote work',
    100: 'Fully remote work'
    # Add more mappings as needed
}

# Replace numerical values with corresponding words using the mapping dictionary
df['remote_ratio'] = df['remote_ratio'].replace(mapping_dict)

# Print the updated DataFrame with words instead of numbers in the␣
 ↪"remote_ratio" column
print(df)
```

```
     work_year experience_level employment_type                   job_title  \
0         2020        Mid-level       Full-Time              Data Scientist
1         2020     Senior-level       Full-Time  Machine Learning Scientist
2         2020     Senior-level       Full-Time           Big Data Engineer
3         2020        Mid-level       Full-Time        Product Data Analyst
4         2020     Senior-level       Full-Time   Machine Learning Engineer
..         …              …              …                          …
602       2022     Senior-level       Full-Time               Data Engineer
603       2022     Senior-level       Full-Time               Data Engineer
```

```
604       2022     Senior-level       Full-Time                Data Analyst
605       2022     Senior-level       Full-Time                Data Analyst
606       2022        Mid-level       Full-Time                AI Scientist

     salary salary_currency  salary_in_usd employee_residence  \
0     70000             EUR          79833            Germany
1    260000             USD         260000              Japan
2     85000             GBP         109024     United Kingdom
3     20000             USD          20000           Honduras
4    150000             USD         150000      United States
..      …               …              …                 …
602  154000             USD         154000      United States
603  126000             USD         126000      United States
604  129000             USD         129000      United States
605  150000             USD         150000      United States
606  200000             USD         200000              India

            remote_ratio company_location company_size
0         No remote work          Germany        Large
1         No remote work            Japan        Small
2    Partial remote work   United Kingdom       Medium
3         No remote work         Honduras        Small
4    Partial remote work    United States        Large
..                   …              …          …
602    Fully remote work    United States       Medium
603    Fully remote work    United States       Medium
604       No remote work    United States       Medium
605    Fully remote work    United States       Medium
606    Fully remote work    United States        Large

[607 rows x 11 columns]
```

[29]:
```python
# Create a bar chart for the "remote_ratio" column
plt.figure(figsize=(8, 6))
df['remote_ratio'].value_counts().plot(kind='bar', color='skyblue')
plt.title('Frequency of Remote Ratio')
plt.xlabel('Remote Ratio')
plt.ylabel('Frequency')
plt.xticks(rotation=45)
plt.show()
```



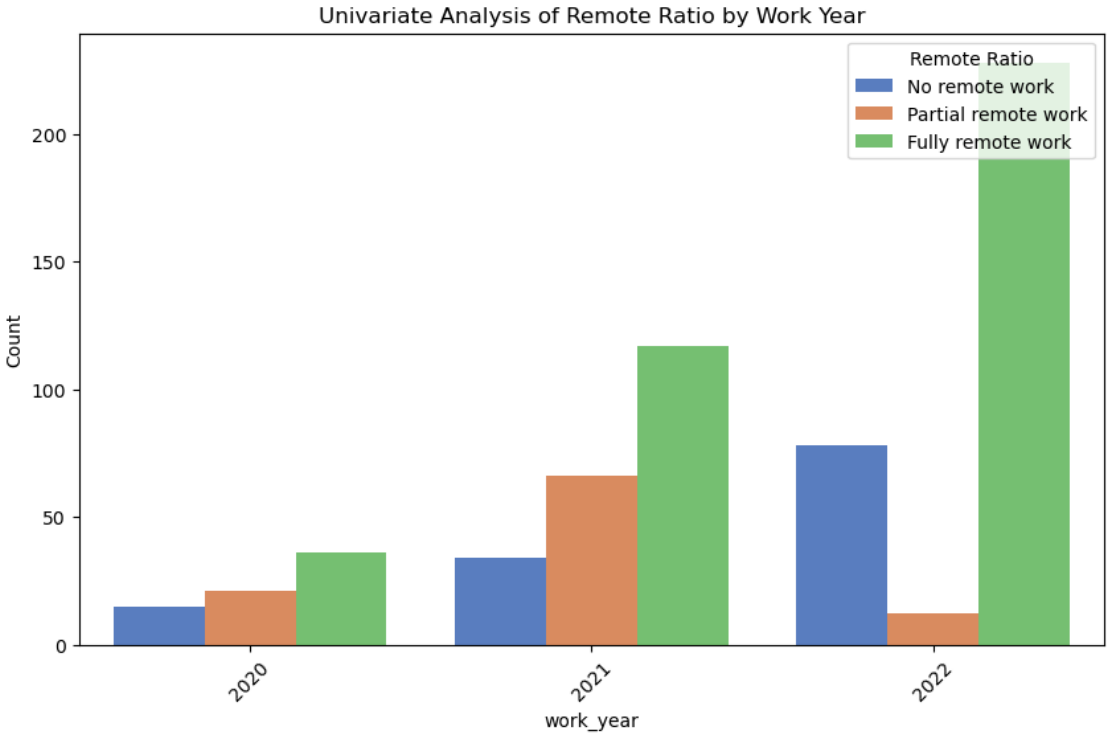
OBSERVATION : Intriguing insights emerge from the analysis of remote work options across job positions. The data reveals distinct preferences among job postings in terms of remote work arrangements. Notably, the frequency of fully remote job postings stands out, signifying a significant trend toward flexible work environments. This preference is followed closely by job postings that offer no remote work, indicating that while remote options are popular, some roles require in-person presence.

[ ]:

[ ]:

[30]:
```
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='work_year', hue='remote_ratio', palette='muted')
plt.title('Univariate Analysis of Remote Ratio by Work Year')
plt.xlabel('work_year')
```

```
plt.ylabel('Count')
plt.legend(title='Remote Ratio', loc='upper right', labels=list(mapping_dict.
 ↪values()))
plt.xticks(rotation=45)
plt.show()
```



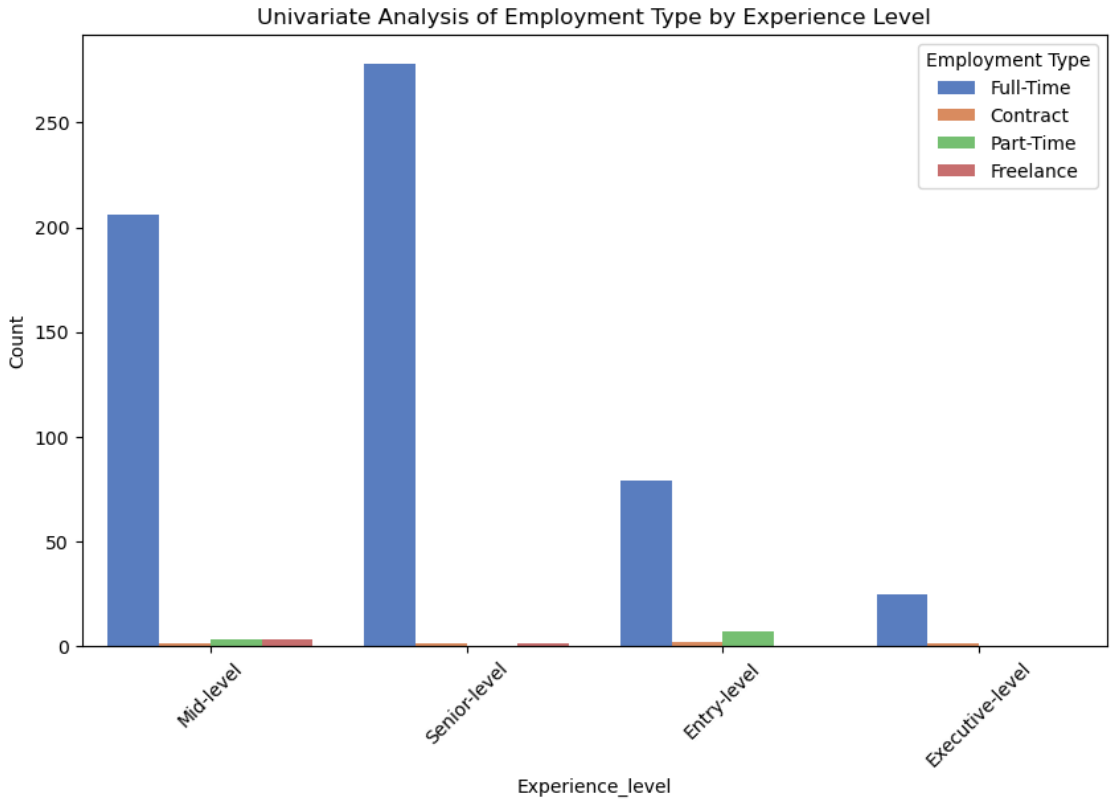
OBERVATION : Overall, the three-year analysis showcases the persistent demand for remote work, with fully remote positions consistently taking the lead. This trend not only highlights the transformation in work dynamics but also suggests a changing paradigm in how organizations and employees approach work flexibility

[ ]:

[ ]:

[31]:
```
#Univariate analysis by experience level and employment type
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='experience_level', hue='employment_type',␣
 ↪palette='muted')
plt.title('Univariate Analysis of Employment Type by Experience Level')
plt.xlabel('Experience_level')
plt.ylabel('Count')
plt.legend(title='Employment Type', loc='upper right')
plt.xticks(rotation=45)
```

```
plt.show()
```



OBSERVATION : From the observation the full-time experience is the highest across all groups (mid-level, executive-level, senior-level, and entry-level), while contract, part-time, and freelance experiences are very low. This observation indicates that full-time experience is the dominant type of experience across these groups.

[ ]:

[ ]:

[32]:
```
# Group the data by "Experience Level" and find the top 3 job titles within␣
 ↪each group
top_job_titles_by_experience = df.groupby('experience_level')['job_title'].
 ↪apply(lambda x: x.value_counts().nlargest(3))

print(top_job_titles_by_experience)
```

```
experience_level
Entry-level       Data Scientist              22
                  Data Analyst                12
                  Data Engineer               12
```

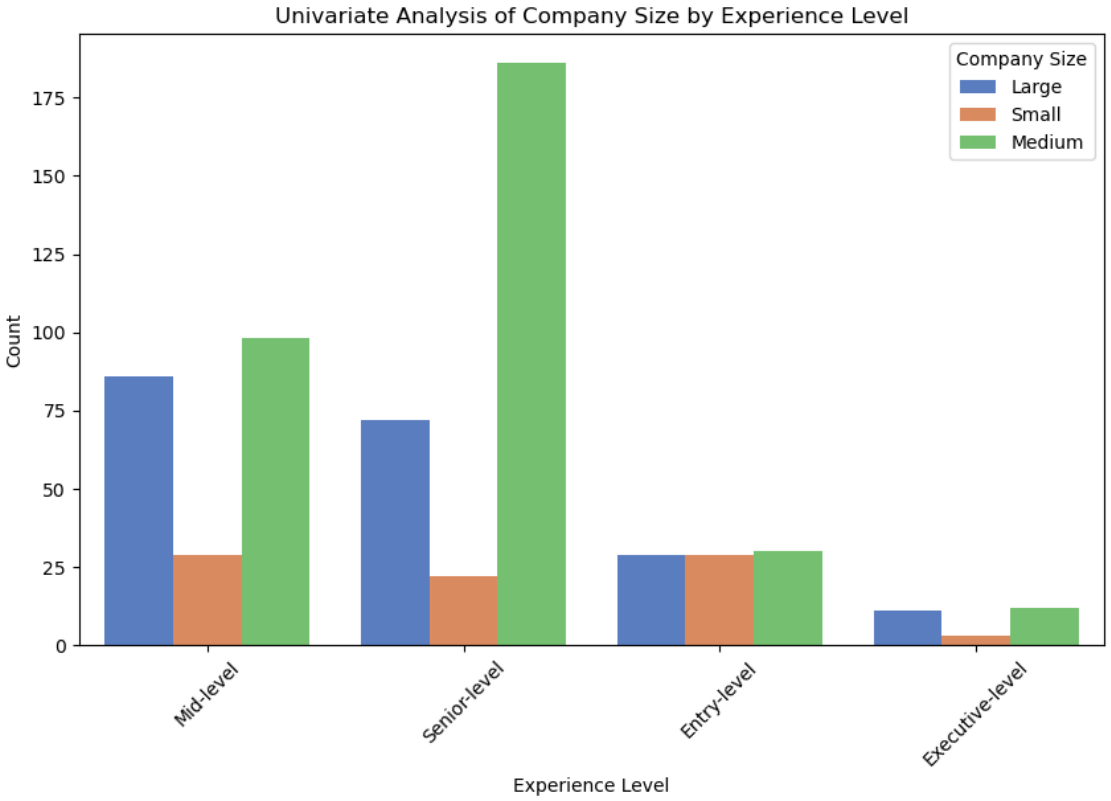
```
Executive-level   Director of Data Science     6
                  Data Engineer                4
                  Head of Data Science         3
Mid-level         Data Scientist              60
                  Data Engineer               53
                  Data Analyst                29
Senior-level      Data Engineer               63
                  Data Scientist              61
                  Data Analyst                54
Name: job_title, dtype: int64
```

OBSERVATION : The majority of "Data Scientist" roles are in the "Mid-level" and "Senior-level" categories, with 60 and 61 positions, respectively. #### "Data Engineer" positions are also prominent across "Mid-level" and "Senior-level" with 53 and 63 positions, respectively. #### "Data Analyst" positions are relatively evenly distributed across "Entry-level," "Mid-level," and "Senior-level," with 12, 29, and 54 positions, respectively. #### "Executive-level" positions are less common overall, with fewer positions in "Director of Data Science," "Head of Data Science," and "Data Engineer" compared to the other levels.

[ ]:

[ ]:

[33]:
```
# Univariate analysis by experience level and company size
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='experience_level', hue='company_size',␣
 ↪palette='muted')
plt.title('Univariate Analysis of Company Size by Experience Level')
plt.xlabel('Experience Level')
plt.ylabel('Count')
plt.legend(title='Company Size', loc='upper right')
plt.xticks(rotation=45)
plt.show()
```

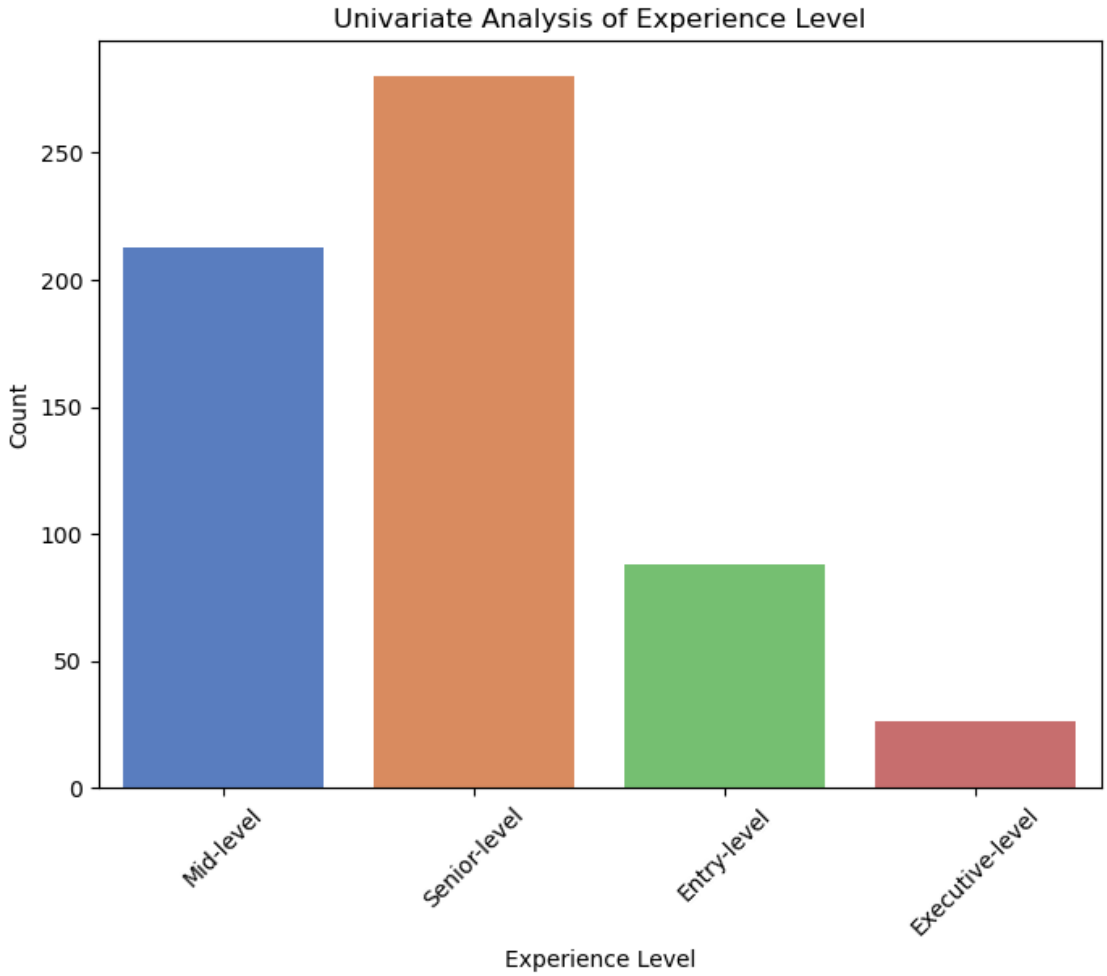


OBSERVATION :Based on the univariate analysis of company size by experience level, it appears that senior-level and mid-level positions have a higher count of employees in medium-sized companies compared to large and small companies.

[ ]:

[ ]:

[34]:
```
# Univariate analysis of experience level
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='experience_level', palette='muted')
plt.title('Univariate Analysis of Experience Level')
plt.xlabel('Experience Level')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



OBSERVATION

Senior Level:Senior-level positions have the highest count, indicating that there are more individuals with senior-level experience in the dataset than any other experience level.

Mid Level:Mid-level positions come next in terms of count, suggesting that there are fewer mid-level individuals compared to senior-level individuals but more than the entry and executive levels.

Entry Level:Entry-level positions have a lower count than both senior and mid-level positions but more than executive-level positions. This indicates the presence of individuals with entry-level experience in the dataset.

Executive Level:Executive-level positions have the lowest count, implying that there are the fewest individuals with executive-level experience in the dataset.

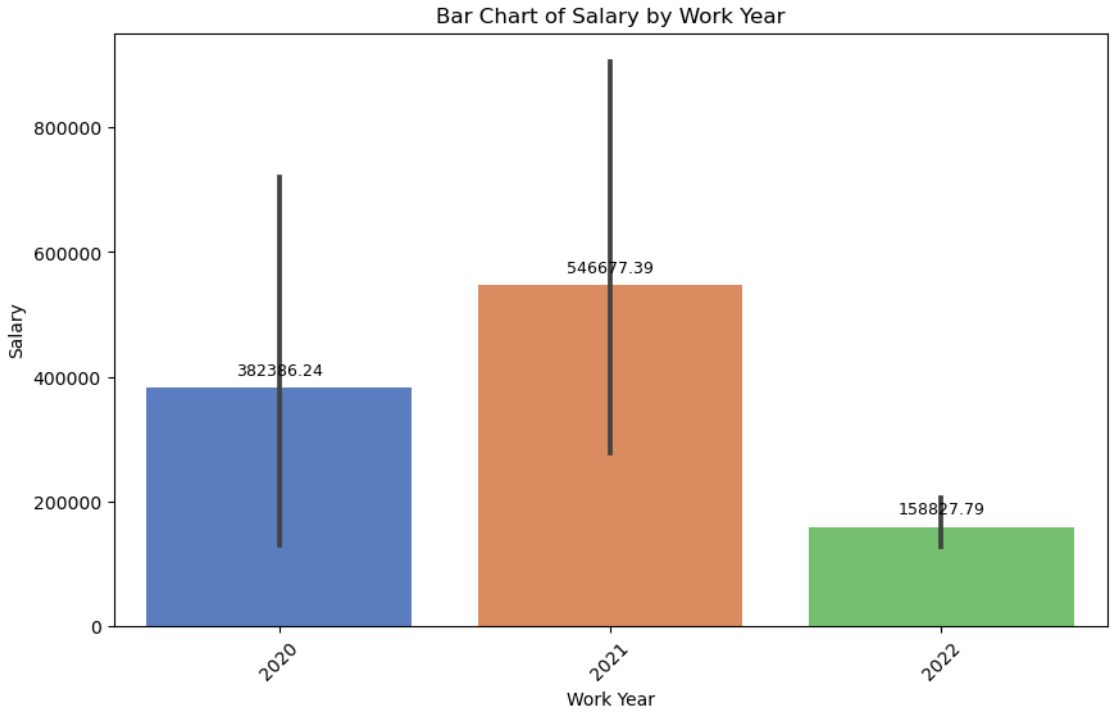
[ ]:

[ ]:

[35]:
```
# Group data by Work Year and calculate descriptive statistics for Salary
descriptive_stats = df.groupby('work_year')['salary'].describe()

# Display the descriptive statistics
print(descriptive_stats)
```

```
           count           mean           std      min       25%       50%  \
work_year
2020        72.0  382386.236111  1.412566e+06   8000.0  54249.75   94500.0
2021       217.0  546677.387097  2.396277e+06   4000.0  60000.00  100000.0
2022       318.0  158827.786164  3.712070e+05  10000.0  80416.50  123000.0

                75%         max
work_year
2020       151750.0  11000000.0
2021       174000.0  30400000.0
2022       160060.0   6000000.0
```

[36]:
```
# Plot the bar chart of Salary by Work Year
plt.figure(figsize=(10, 6))
ax = sns.barplot(data=df, x='work_year', y='salary', palette='muted')
plt.title('Bar Chart of Salary by Work Year')
plt.xlabel('Work Year')
plt.ylabel('Salary')
plt.xticks(rotation=45)

# Add figures (values) above each bar
for p in ax.patches:
    ax.annotate(f"{p.get_height():.2f}", (p.get_x() + p.get_width() / 2., p.
 ↪get_height()),
                ha='center', va='bottom', fontsize=9, color='black', xytext=(0,␣
 ↪5),
                textcoords='offset points')

plt.show()
```



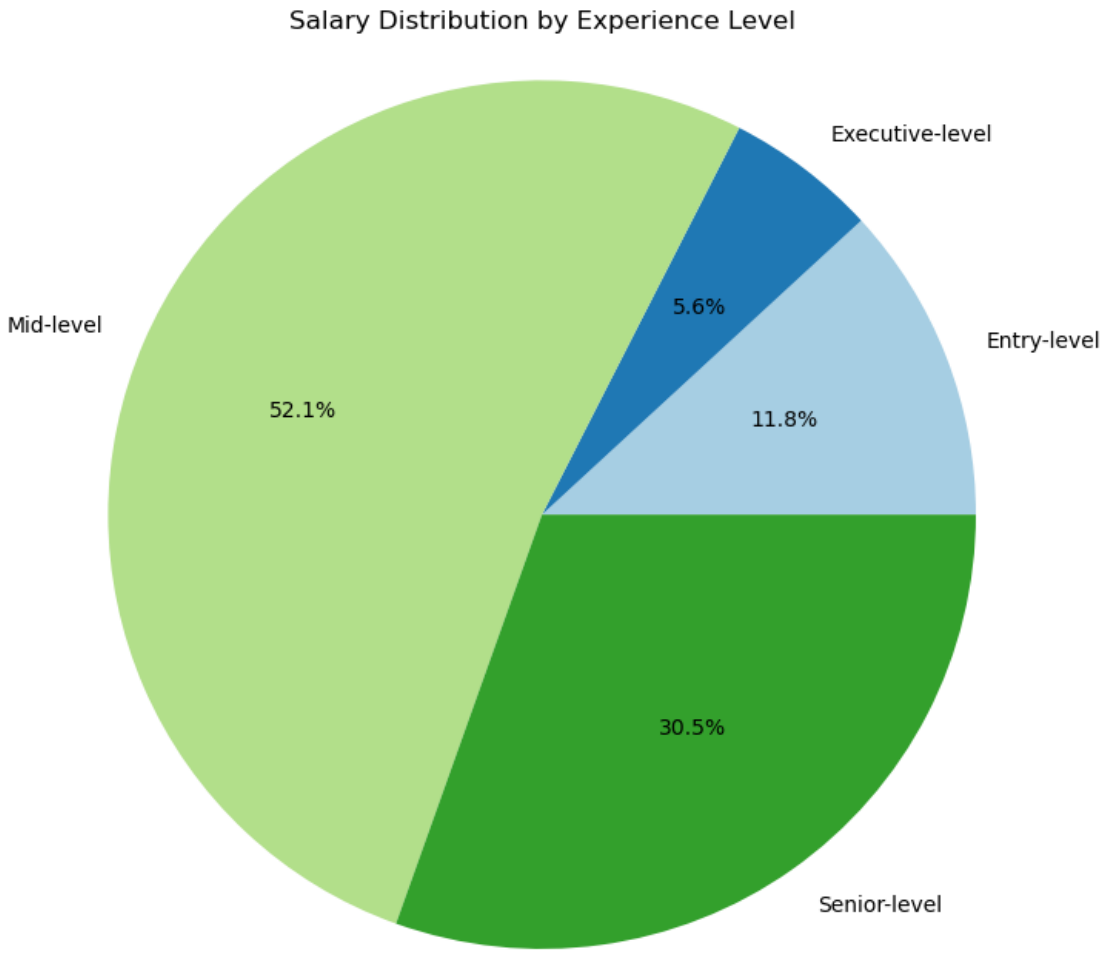
OBSERVATION : From the analysis 2021 had the highest salary of 547777.39 while 2020 had the second highest salary of 382386.24 and lastly year 2022 was the lowest with 158827.79

[ ]:

[ ]:

[ ]:

[37]:
```
# Group the data by experience level and calculate the total salary for each group
experience_salary = df.groupby('experience_level')['salary'].sum()

# Create a pie chart to show the proportion of each experience level's salary
plt.figure(figsize=(8, 8))
plt.pie(experience_salary, labels=experience_salary.index, autopct='%1.1f%%', colors=plt.cm.Paired.colors)
plt.title('Salary Distribution by Experience Level')
plt.axis('equal')
plt.show()
```



OBSERVATION

Mid-Level Positions Dominate Salary Distribution: Mid-level positions account for the largest share of the salary distribution, comprising approximately 52.1% of the total.This indicates that a significant proportion of individuals in the dataset hold mid-level positions in terms of experience.

Senior-Level Positions Are the Second Most Common:Senior-level positions represent a substantial portion of the salary distribution, making up about 30.5% of the total.While mid-level positions are more prevalent, senior-level positions still have a significant presence in the dataset.

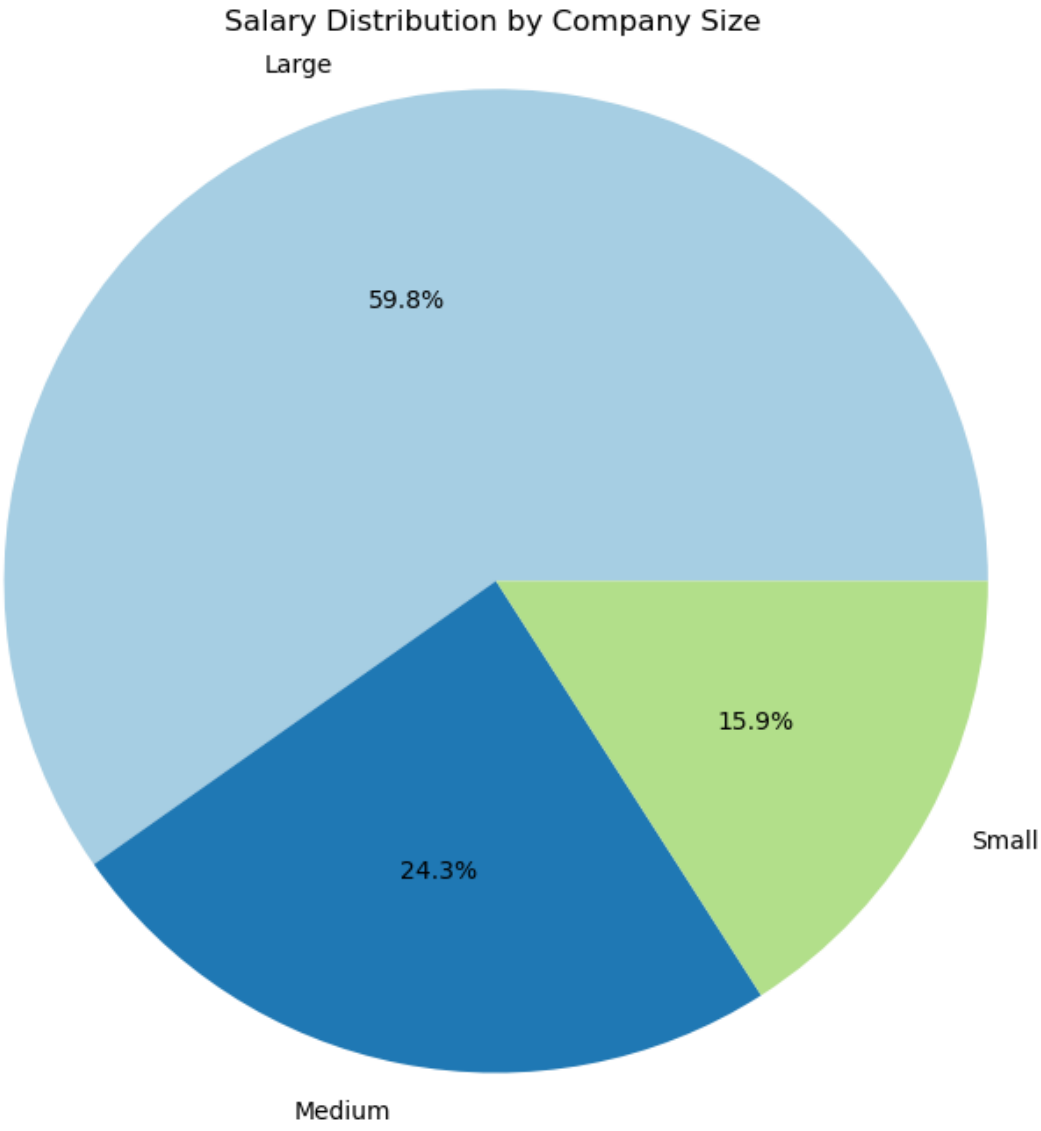
Entry-Level Positions Constitute a Smaller Share:Entry-level positions have a smaller share of the salary distribution, at approximately 11.8%.This suggests that fewer individuals in the dataset are in entry-level positions compared to mid-level and senior-level positions.

Executive-Level Positions Have a Minority Share:Executive-level positions, while important, constitute a minority of the salary distribution, accounting for around 5.6% of the total.This observation indicates that executive-level roles are less common in the dataset compared to other experience levels.

[ ]:

[ ]:

[38]:
```
# Group the data by company size and calculate the total salary for each group
company_salary = df.groupby('company_size')['salary'].sum()

# Create a pie chart to show the proportion of each company size's salary
plt.figure(figsize=(8, 8))
plt.pie(company_salary, labels=company_salary.index, autopct='%1.1f%%',␣
 ↪colors=plt.cm.Paired.colors)
plt.title('Salary Distribution by Company Size')
plt.axis('equal')
plt.show()
```



OBSERVATION: #### Large companies represent the majority of the salary distribution, accounting for approximately 59.8% of the total.This observation indicates that a significant portion of individuals in the dataset work for large companies. #### Medium-sized companies make up about 24.3% of the salary distribution. While they are not as prevalent as large companies, they still constitute a substantial portion of the dataset. #### Small companies have the smallest share of the salary distribution, at approximately 15.9%.This suggests that fewer individuals in the dataset are employed by small companies compared to large and medium-sized ones.

[ ]:

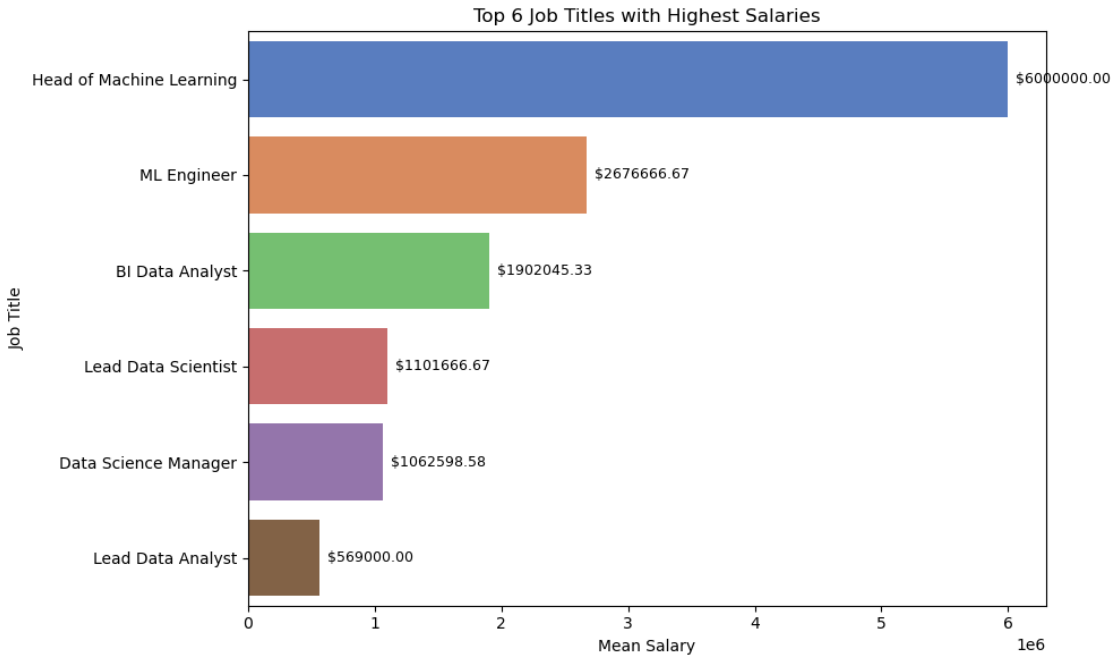
[39]:
```
# Group the data by job title and find the maximum salary for each group
highest_salaries = df.groupby('job_title')['salary'].max()
```

```
# Display the highest salaries by job title
print(highest_salaries)
```

```
job_title
3D Computer Vision Researcher                  400000
AI Scientist                                  1335000
Analytics Engineer                             205300
Applied Data Scientist                         380000
Applied Machine Learning Scientist             423000
BI Data Analyst                              11000000
Big Data Architect                             125000
Big Data Engineer                             1672000
Business Data Analyst                         1400000
Cloud Data Engineer                            160000
Computer Vision Engineer                       180000
Computer Vision Software Engineer              150000
Data Analyst                                   450000
Data Analytics Engineer                        110000
Data Analytics Lead                            405000
Data Analytics Manager                         150260
Data Architect                                 266400
Data Engineer                                 4450000
Data Engineering Manager                       174000
Data Science Consultant                        423000
Data Science Engineer                          159500
Data Science Manager                          7000000
Data Scientist                               30400000
Data Specialist                                165000
Director of Data Engineering                   200000
Director of Data Science                       325000
ETL Developer                                   50000
Finance Data Analyst                            45000
Financial Data Analyst                         450000
Head of Data                                   235000
Head of Data Science                           224000
Head of Machine Learning                      6000000
Lead Data Analyst                             1450000
Lead Data Engineer                             276000
Lead Data Scientist                           3000000
Lead Machine Learning Engineer                  80000
ML Engineer                                   8500000
Machine Learning Developer                     100000
Machine Learning Engineer                     4900000
Machine Learning Infrastructure Engineer       195000
Machine Learning Manager                       157000
Machine Learning Scientist                     260000
Marketing Data Analyst                          75000
```

```
NLP Engineer                                240000
Principal Data Analyst                      170000
Principal Data Engineer                     600000
Principal Data Scientist                    416000
Product Data Analyst                        450000
Research Scientist                          450000
Staff Data Scientist                        105000
Name: salary, dtype: int64
```

[40]:
```python
# Calculate the mean salary for each job title
mean_salaries = df.groupby('job_title')['salary'].mean().reset_index()

# Sort the data in descending order based on mean salary
top_6_job_titles = mean_salaries.sort_values(by='salary', ascending=False).
 ↪head(6)

# Create a horizontal bar plot
plt.figure(figsize=(10, 6))
ax = sns.barplot(data=top_6_job_titles, x='salary', y='job_title',␣
 ↪palette='muted')
plt.title('Top 6 Job Titles with Highest Salaries')
plt.xlabel('Mean Salary')
plt.ylabel('Job Title')

# Add figures (values) to the bars
for p in ax.patches:
    ax.annotate(f"${p.get_width():.2f}", (p.get_width(), p.get_y() + p.
 ↪get_height() / 2.),
                ha='left', va='center', fontsize=9, color='black', xytext=(5,␣
 ↪0),
                textcoords='offset points')

plt.tight_layout()
plt.show()
```



OBSERVATION: The highest salary for the different job title is Head of Machine learning then machine learning engineer and BI data analyst followed by others as seen above.

[ ]: