Job Role Salary and Trends Analysis

Project Description

- The objective of the Job Role Salary and Trends Analysis project is to analyze job roles and their corresponding salaries using a dataset that includes various attributes such as job title, experience level, employment type, employee residence, remote ratio, company location, and company size. This involves cleaning and preparing the data, conducting descriptive statistics, and performing exploratory data analysis (EDA) to reveal patterns and trends. Visualizations such as bar plots, box plots, and heatmaps are utilized to highlight salary distributions and the impact of different factors on salaries. The project aims to provide valuable insights into how experience, remote work, company size, and location influence job role salaries, offering guidance for career planning, recruitment strategies, and market positioning.

Import Libraries

```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
```

Import .csv File

```
data = pd.read_csv("/content/03_Data Science Salaries 2023 Analysis.csv")
```

Data Preprocessing

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3755 entries, 0 to 3754
Data columns (total 11 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   work_year           3755 non-null   int64 
 1   experience_level    3755 non-null   object
 2   employment_type     3755 non-null   object
 3   job_title           3755 non-null   object
 4   salary              3755 non-null   int64 
 5   salary_currency     3755 non-null   object
 6   salary_in_usd       3755 non-null   int64 
 7   employee_residence  3755 non-null   object
 8   remote_ratio        3755 non-null   int64 
 9   company_location    3755 non-null   object
 10  company_size        3755 non-null   object
dtypes: int64(4), object(7)
memory usage: 322.8+ KB
```

```
print(data.isnull().sum())
```

```
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
```

```
data.shape
```

```
(3755, 11)
```

```
data.dtypes
```

```
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
```

```
data["job_title"].unique()
```

```
array(['Principal Data Scientist', 'ML Engineer', 'Data Scientist',
       'Applied Scientist', 'Data Analyst', 'Data Modeler',
       'Research Engineer', 'Analytics Engineer',
       'Business Intelligence Engineer', 'Machine Learning Engineer',
       'Data Strategist', 'Data Engineer', 'Computer Vision Engineer',
       'Data Quality Analyst', 'Compliance Data Analyst',
       'Data Architect', 'Applied Machine Learning Engineer',
       'AI Developer', 'Research Scientist', 'Data Analytics Manager',
       'Business Data Analyst', 'Applied Data Scientist',
       'Staff Data Analyst', 'ETL Engineer', 'Data DevOps Engineer',
       'Head of Data', 'Data Science Manager', 'Data Manager',
       'Machine Learning Researcher', 'Big Data Engineer',
       'Data Specialist', 'Lead Data Analyst', 'BI Data Engineer',
       'Director of Data Science', 'Machine Learning Scientist',
       'MLOps Engineer', 'AI Scientist', 'Autonomous Vehicle Technician',
       'Applied Machine Learning Scientist', 'Lead Data Scientist',
       'Cloud Database Engineer', 'Financial Data Analyst',
       'Data Infrastructure Engineer', 'Software Data Engineer',
       'AI Programmer', 'Data Operations Engineer', 'BI Developer',
       'Data Science Lead', 'Deep Learning Researcher', 'BI Analyst',
       'Data Science Consultant', 'Data Analytics Specialist',
       'Machine Learning Infrastructure Engineer', 'BI Data Analyst',
       'Head of Data Science', 'Insight Analyst',
       'Deep Learning Engineer', 'Machine Learning Software Engineer',
       'Big Data Architect', 'Product Data Analyst',
       'Computer Vision Software Engineer', 'Azure Data Engineer',
       'Marketing Data Engineer', 'Data Analytics Lead', 'Data Lead',
       'Data Science Engineer', 'Machine Learning Research Engineer',
       'NLP Engineer', 'Manager Data Management',
       'Machine Learning Developer', '3D Computer Vision Researcher',
       'Principal Machine Learning Engineer', 'Data Analytics Engineer',
       'Data Analytics Consultant', 'Data Management Specialist',
       'Data Science Tech Lead', 'Data Scientist Lead',
       'Cloud Data Engineer', 'Data Operations Analyst',
       'Marketing Data Analyst', 'Power BI Developer',
       'Product Data Scientist', 'Principal Data Architect',
       'Machine Learning Manager', 'Lead Machine Learning Engineer',
       'ETL Developer', 'Cloud Data Architect', 'Lead Data Engineer',
       'Head of Machine Learning', 'Principal Data Analyst',
       'Principal Data Engineer', 'Staff Data Scientist',
       'Finance Data Analyst'], dtype=object)
```

```
data.columns = data.columns.str.lower()
```

```
pd.get_dummies(data, columns=['experience_level', 'employment_type', 'employee_residence', 'company_location', 'company_size'], drop_fi
```

	work_year	job_title	salary	salary_currency	salary_in_usd	remote_ratio	e
0	2023	Principal Data Scientist	80000	EUR	85847	100	
1	2023	ML Engineer	30000	USD	30000	100	
2	2023	ML Engineer	25500	USD	25500	100	
3	2023	Data Scientist	175000	USD	175000	100	
4	2023	Data Scientist	120000	USD	120000	100	
...	...	...	...	...	...	...	
3750	2020	Data Scientist	412000	USD	412000	100	
3751	2021	Principal Data Scientist	151000	USD	151000	100	
3752	2020	Data Scientist	105000	USD	105000	100	
3753	2020	Business Data Analyst	100000	USD	100000	100	
3754	2021	Data Science Manager	7000000	INR	94665	50	

3755 rows × 162 columns

```
salary_bins = [0, 50000, 100000, 150000, 200000, np.inf]
salary_labels = ['Low', 'Medium', 'High', 'Very High', 'Top']
data['salary_bin'] = pd.cut(data['salary_in_usd'], bins=salary_bins, labels=salary_labels)
```

```
data.head()
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency
0	2023	SE	FT	Principal Data Scientist	80000	EUR
1	2023	MI	CT	ML Engineer	30000	USD
2	2023	MI	CT	ML Engineer	25500	USD
3	2023	SE	FT	Data Scientist	175000	USD
4	2023	SE	FT	Data Scientist	120000	USD

```
print(data.isnull().sum())
```

```
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
salary_bin            0
dtype: int64
```

Data Analysis

```
role_salary = data.groupby('job_title')['salary_in_usd'].mean().reset_index()
role_salary.columns = ['Job_Title', 'Average_Salary_in_USD']

print(role_salary)
```

```
                        Job_Title  Average_Salary_in_USD
0   3D Computer Vision Researcher           21352.250000
1                    AI Developer          136666.090909
2                   AI Programmer           55000.000000
3                    AI Scientist          110120.875000
4              Analytics Engineer          152368.631068
..                            ...                    ...
88              Research Engineer          163108.378378
89             Research Scientist          161214.195122
90         Software Data Engineer           62510.000000
91             Staff Data Analyst           15000.000000
92           Staff Data Scientist          105000.000000

[93 rows x 2 columns]
```

```
experience_salary = data.groupby(['job_title', 'experience_level'])['salary_in_usd'].mean().unstack().reset_index()
print(experience_salary)
```

```
experience_level                      job_title             EN        EX  \
0                   3D Computer Vision Researcher   35000.000000       NaN
1                                AI Developer  130884.500000       NaN
2                               AI Programmer   55000.000000       NaN
3                                AI Scientist   52781.285714  200000.0
4                          Analytics Engineer  130000.000000  175125.0
..                                        ...            ...       ...
88                          Research Engineer  130000.000000       NaN
89                         Research Scientist  118280.888889   84053.0
90                     Software Data Engineer            NaN       NaN
91                         Staff Data Analyst            NaN   15000.0
92                       Staff Data Scientist            NaN       NaN

experience_level             MI             SE
0                    5409.000000   10000.000000
1                  137510.000000  147666.666667
2                            NaN            NaN
3                  117726.200000  201278.000000
4                  102480.230769  158404.024691
..                           ...            ...
88                 178000.000000  174773.181818
89                 141575.086957  179892.979592
90                  75020.000000   50000.000000
91                           NaN            NaN
92                           NaN  105000.000000

[93 rows x 5 columns]
```

```
employment_salary = data.groupby(['job_title', 'employment_type'])['salary_in_usd'].mean().unstack().reset_index()
print(employment_salary)
```

```
employment_type                       job_title        CT       FL  \
0                   3D Computer Vision Researcher       NaN      NaN
1                                AI Developer       NaN      NaN
2                               AI Programmer       NaN      NaN
3                                AI Scientist       NaN      NaN
4                          Analytics Engineer    7500.0      NaN
..                                        ...       ...      ...
88                          Research Engineer       NaN      NaN
89                         Research Scientist       NaN      NaN
90                     Software Data Engineer       NaN  50000.0
91                         Staff Data Analyst       NaN      NaN
92                       Staff Data Scientist  105000.0      NaN

employment_type             FT       PT
0                    26666.666667   5409.0
1                   136666.090909      NaN
2                    55000.000000      NaN
3                   124138.142857  12000.0
4                   153788.911765      NaN
..                            ...      ...
88                  163108.378378      NaN
89                  161214.195122      NaN
90                   75020.000000      NaN
91                   15000.000000      NaN
92                            NaN      NaN

[93 rows x 5 columns]
```

```
remote_salary = data.groupby(['job_title', 'remote_ratio'])['salary_in_usd'].mean().unstack().reset_index()
print(remote_salary)
```

```
remote_ratio                      job_title             0            50  \
0              3D Computer Vision Researcher   20000.000000   7704.500000
1                              AI Developer   98118.166667  166666.666667
2                             AI Programmer   70000.000000           NaN
3                              AI Scientist  191278.000000   94842.333333
4                         Analytics Engineer  160663.369565   68750.000000
..                                       ...           ...           ...
88                         Research Engineer  173395.133333           NaN
89                        Research Scientist  174970.777778   97190.000000
90                    Software Data Engineer           NaN   50000.000000
91                        Staff Data Analyst   15000.000000           NaN
92                      Staff Data Scientist           NaN           NaN

remote_ratio            100
0              50000.000000
1             207309.000000
2              40000.000000
3              90357.300000
4             148471.890909
..                      ...
88            119022.285714
89            158944.235294
90             75020.000000
91                      NaN
92            105000.000000

[93 rows x 4 columns]
```

```
company_size_salary = data.groupby(['job_title', 'company_size'])['salary_in_usd'].mean().unstack().reset_index()
print(company_size_salary)
```

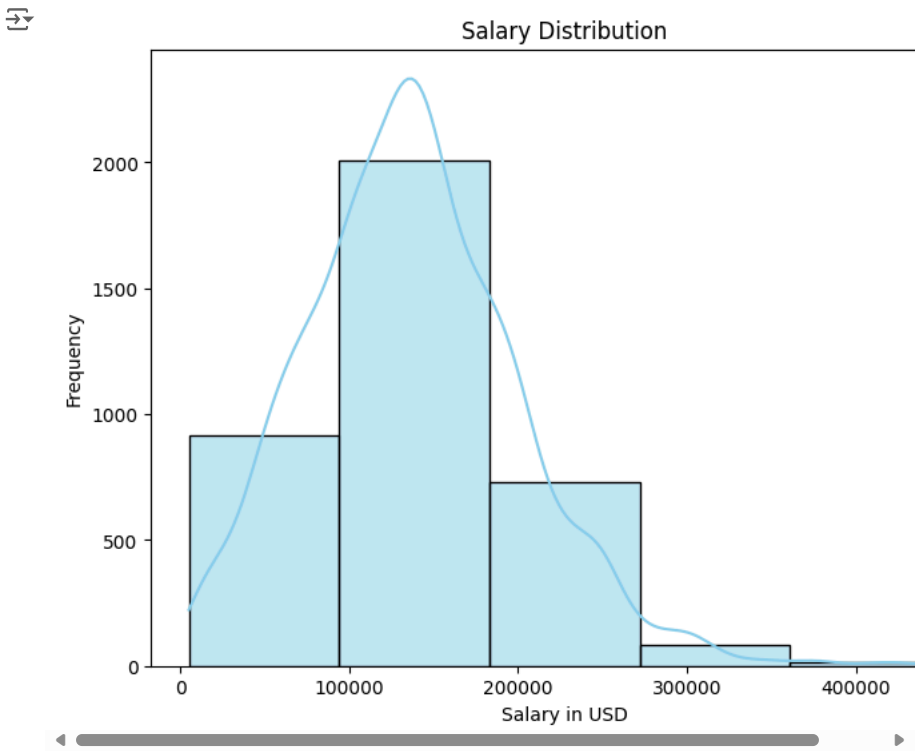
```
company_size                      job_title             L             M  \
0              3D Computer Vision Researcher           NaN   12704.500000
1                              AI Developer  257309.000000   97900.833333
2                             AI Programmer   70000.000000   40000.000000
3                              AI Scientist  173744.166667   82704.000000
4                         Analytics Engineer  130000.000000  153623.455446
..                                       ...           ...           ...
88                         Research Engineer           NaN  166910.114286
89                        Research Scientist  101745.800000  180579.460317
90                    Software Data Engineer   75020.000000   50000.000000
91                        Staff Data Analyst           NaN   15000.000000
92                      Staff Data Scientist           NaN  105000.000000

company_size          S
0              30000.00
1             133768.00
2                   NaN
3              61189.80
4              48000.00
..                  ...
88             96578.00
89             79217.75
90                  NaN
91                  NaN
92                  NaN

[93 rows x 4 columns]
```

```
new_var = data.describe()
new_var
```

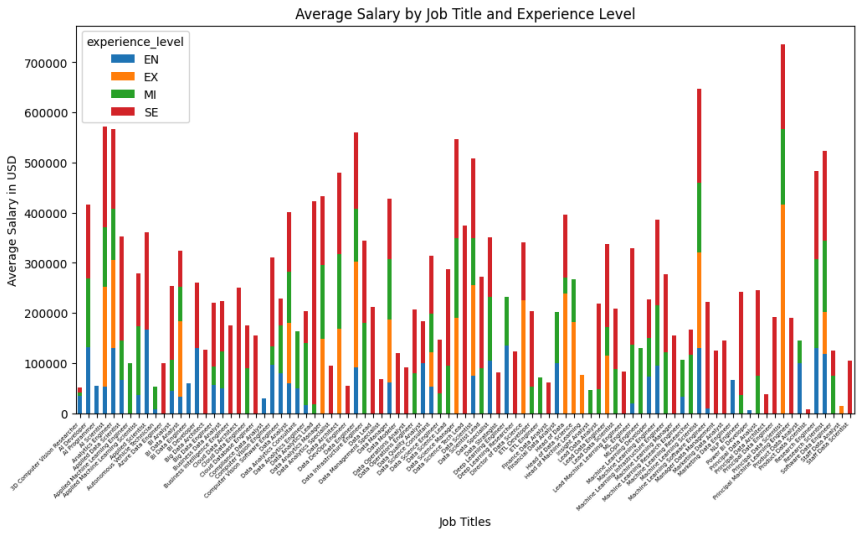
	work_year	salary	salary_in_usd	remote_ratio
count	3755.000000	3.755000e+03	3755.000000	3755.000000
mean	2022.373635	1.906956e+05	137570.389880	46.271638
std	0.691448	6.716765e+05	63055.625278	48.589050
min	2020.000000	6.000000e+03	5132.000000	0.000000
25%	2022.000000	1.000000e+05	95000.000000	0.000000
50%	2022.000000	1.380000e+05	135000.000000	0.000000
75%	2023.000000	1.800000e+05	175000.000000	100.000000
max	2023.000000	3.040000e+07	450000.000000	100.000000

Visualizations

```
plt.figure(figsize=(8, 6))
sns.histplot(data['salary_in_usd'], bins=5, kde=True, color='skyblue')
plt.title('Salary Distribution')
plt.xlabel('Salary in USD')
plt.ylabel('Frequency')
plt.show()
```

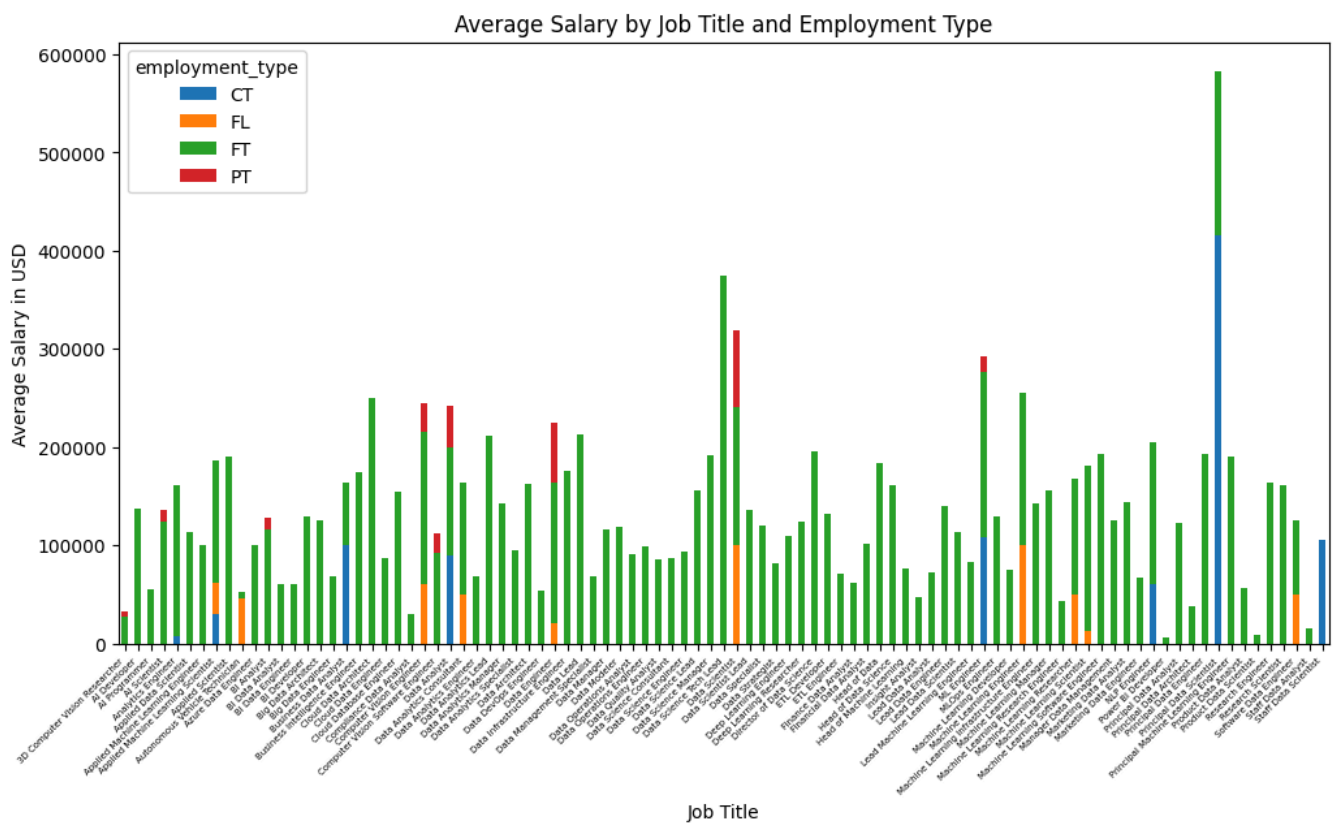



```
experience_salary = data.groupby(['job_title', 'experience_level'])['salary_in_usd'].mean().unstack().reset_index()
experience_salary.plot(kind='bar', x='job_title', stacked=True, figsize=(12, 6))
plt.title('Average Salary by Job Title and Experience Level')
plt.xlabel('Job Titles')
plt.ylabel('Average Salary in USD')
plt.xticks(rotation=45, ha='right', fontsize = 5)
plt.show()
plt.tight_layout()
```



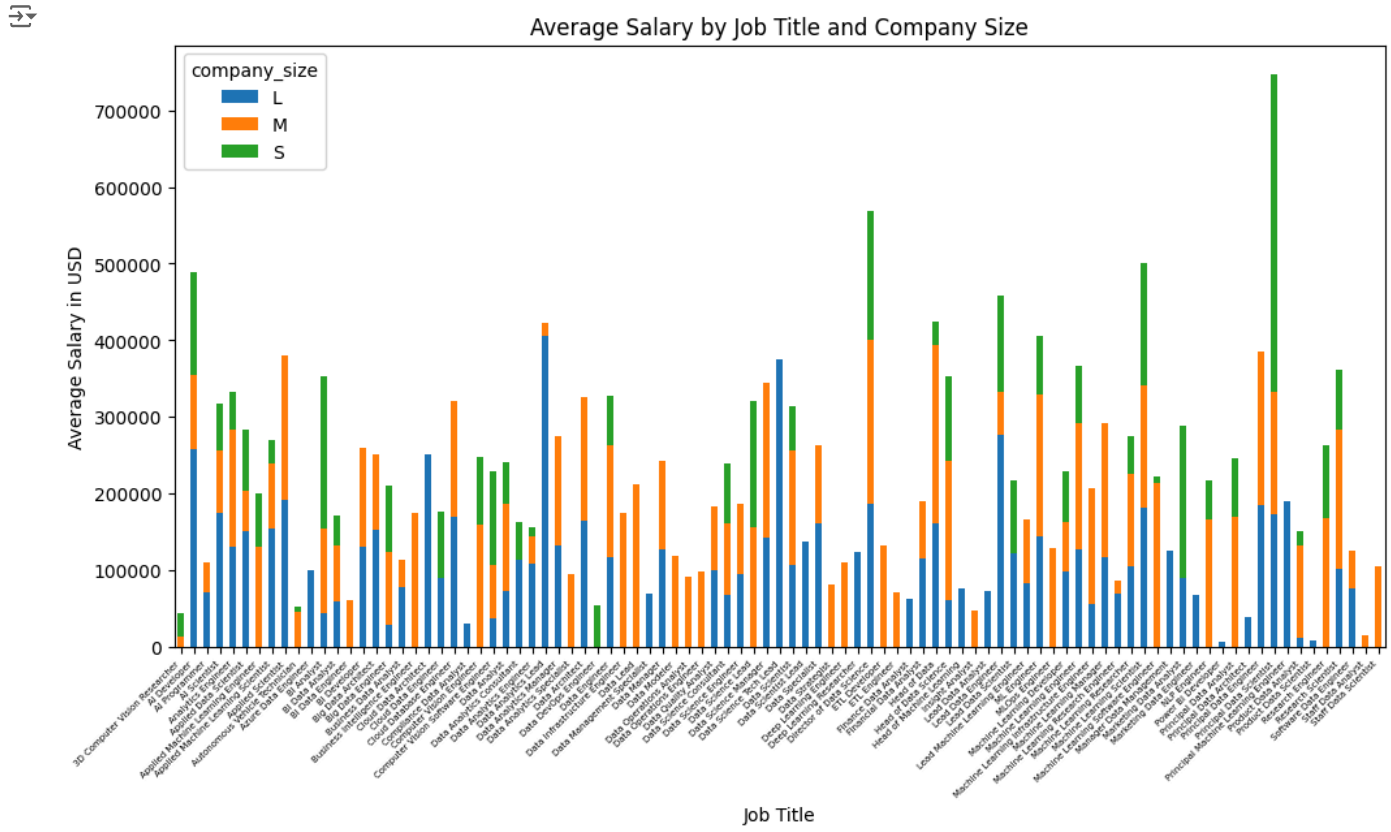

```
<Figure size 640x480 with 0 Axes>
```

```
employment_salary = data.groupby(['job_title', 'employment_type'])['salary_in_usd'].mean().unstack().reset_index()
employment_salary.plot(kind='bar', x='job_title', stacked=True, figsize=(12, 6))
plt.title('Average Salary by Job Title and Employment Type')
plt.xlabel('Job Title')
plt.ylabel('Average Salary in USD')
plt.xticks(rotation=45, ha='right', fontsize = 5)
plt.show()
```

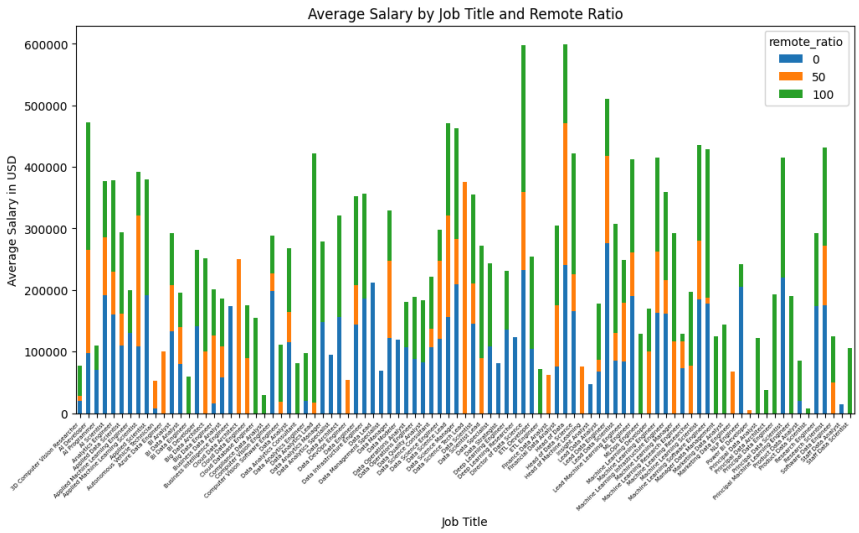



```
company_size_salary = data.groupby(['job_title', 'company_size'])['salary_in_usd'].mean().unstack().reset_index()
```

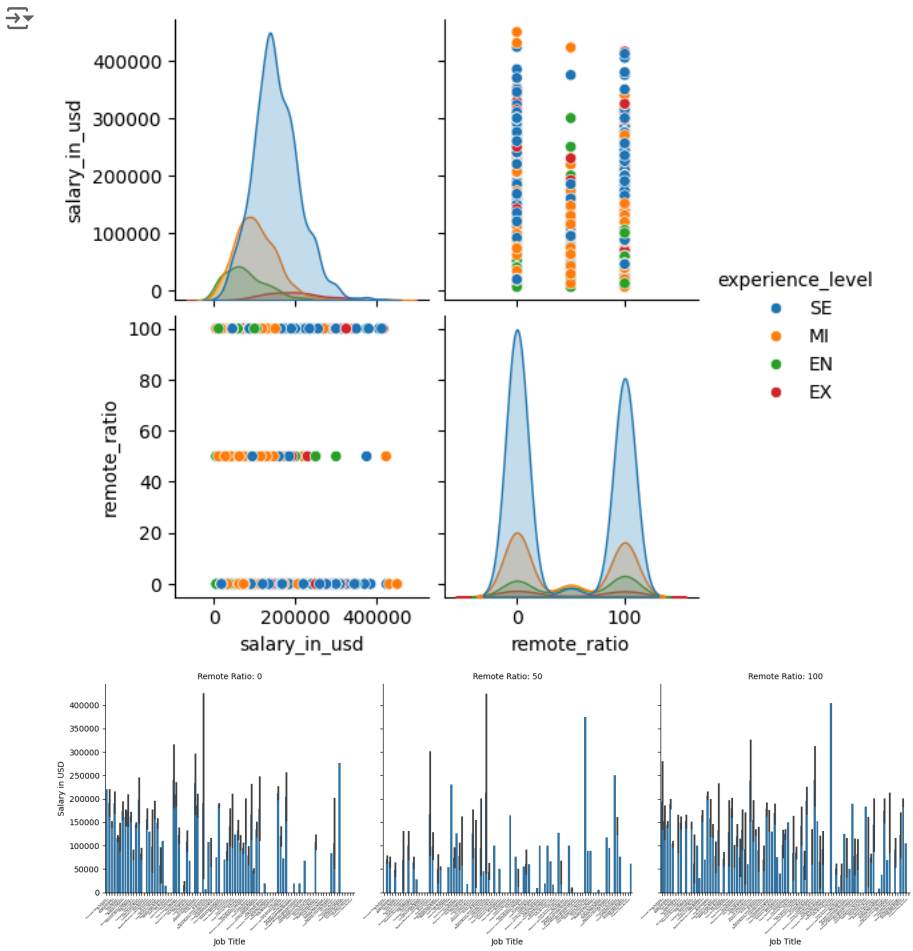
```
company_size_salary.plot(kind='bar', x='job_title', stacked=True, figsize=(12, 6))
plt.title('Average Salary by Job Title and Company Size')
plt.xlabel('Job Title')
plt.ylabel('Average Salary in USD')
plt.xticks(rotation=45, ha='right', fontsize = 5)
plt.show()
```



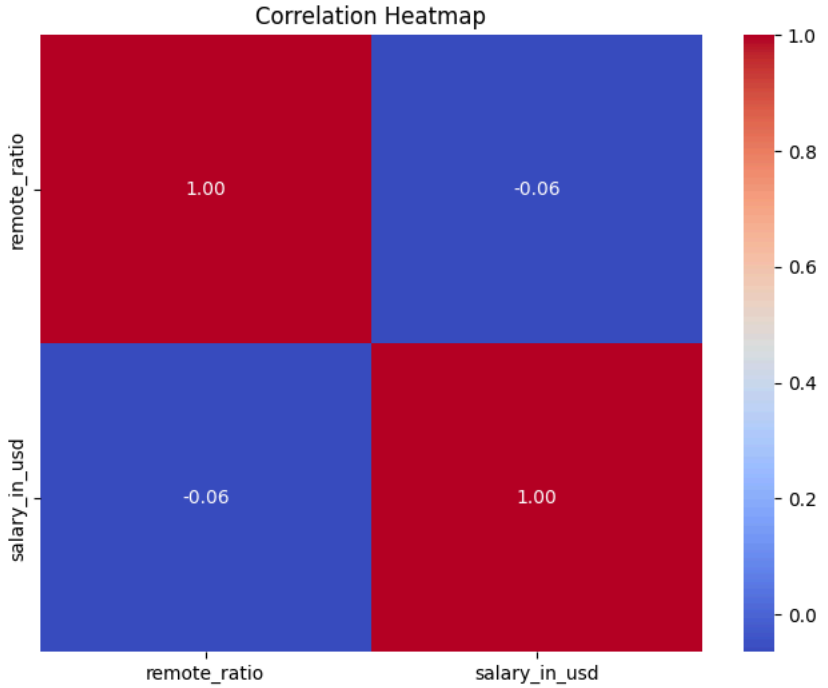

```
remote_salary = data.groupby(['job_title', 'remote_ratio'])['salary_in_usd'].mean().unstack().reset_index()
remote_salary.plot(kind='bar', x='job_title', stacked=True, figsize=(12, 6))
plt.title('Average Salary by Job Title and Remote Ratio')
plt.xlabel('Job Title')
plt.ylabel('Average Salary in USD')
plt.xticks(rotation=45, ha='right', fontsize = 5)
plt.show()
```



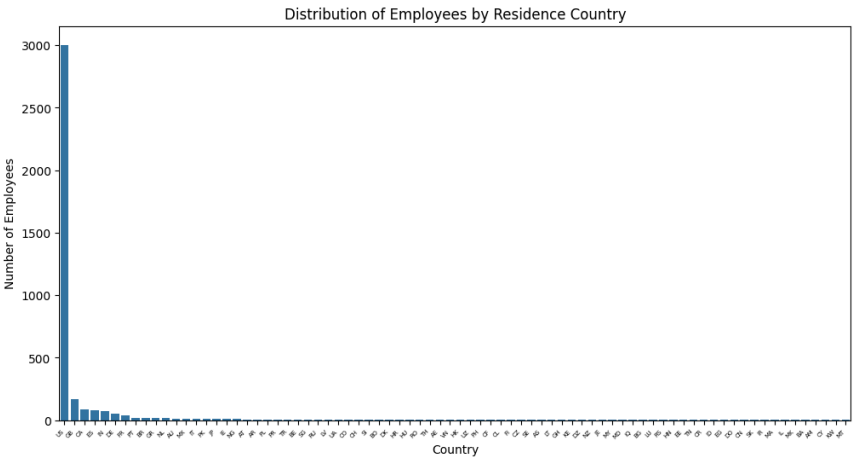

```
sns.pairplot(data[['salary_in_usd', 'experience_level', 'remote_ratio', 'company_size']], hue='experience_level')
plt.show()

g = sns.FacetGrid(data, col='remote_ratio', height=5, aspect=1)

job_order = data['job_title'].unique()

g.map(sns.barplot, 'job_title', 'salary_in_usd', order=job_order)
for ax in g.axes.flat:
    for label in ax.get_xticklabels():
        label.set_rotation(45)
        label.set_ha('right')
        label.set_fontsize(3)

g.set_axis_labels('Job Title', 'Salary in USD')
g.set_titles('Remote Ratio: {col_name}')

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




```
corr = data[['remote_ratio', 'salary_in_usd']].corr()

plt.figure(figsize=(8, 6))
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap')
plt.show()
```




```
df = pd.DataFrame(data)

plt.figure(figsize=(12, 6))
sns.countplot(data=df, x='employee_residence', order=df['employee_residence'].value_counts().index)
plt.title('Distribution of Employees by Residence Country')
plt.xlabel('Country')
plt.ylabel('Number of Employees')
plt.xticks(rotation=45, ha='right', fontsize = 5)
plt.show()
```



```
def classify_region(company_location):
    if company_location in {"NG", "GH", "KE", "MA", "DZ", "EG", "CF"}:
        return "Africa"
    if company_location in {"IN", "HK", "SG", "TH", "VN", "AM", "PK", "IR", "ID", "AE", "MY", "JP", "IQ", "CN"}:
        return "Asia"
    if company_location in {"ES", "DE", "GB", "NL", "CH", "FR", "FI", "UA", "IE", "IL", "SE", "SI", "PT", "RU", "HR", "EE", "BA", "GR",
        return "Europe"
    if company_location in {"US", "CA", "MX", "CR", "BS", "PR", "HN"}:
        return "North America"
    if company_location in {"CO", "BR", "AR", "BO", "CL"}:
        return "South America"
    if company_location in {"AU", "NZ", "AS"}:
        return "Oceania"
    return "Other"

df['region'] = df['company_location'].apply(classify_region)

print(df[['company_location', 'region']])
```

```
      company_location         region
0                   ES         Europe
1                   US  North America
2                   US  North America
```

```
import plotly.express as px


region_counts = df['region'].value_counts().reset_index()
region_counts.columns = ['region', 'count']

fig = px.bar(
    region_counts,
    x='region',
    y='count',
    title='Number of Employees by Region',
    labels={'count': 'Number of Employees', 'region': 'Region'},
    color='region'
)
fig.show()
```