```
import warnings
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

df = pd.read_csv('C:\Users\jdeb\Downloads\salaries.csv')
```

```
df.head()
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	2024	SE	FT	AI Engineer	90000	USD	90000	AE	0	AE	L
1	2024	SE	FT	Machine Learning Engineer	205000	USD	205000	US	0	US	M
2	2024	SE	FT	Machine Learning Engineer	90000	USD	90000	US	0	US	M
3	2024	SE	FT	Machine Learning Engineer	150000	USD	150000	AU	0	AU	M
4	2024	SE	FT	Machine Learning Engineer	170000	USD	170000	AU	0	AU	M

```
df.info()
```

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

```
df.describe()
```

	work_year	salary	salary_in_usd	remote_ratio
count	13972.000000	1.397200e+04	13972.000000	13972.000000
mean	2023.139424	1.558544e+05	149874.594544	30.394929
std	0.692133	2.357995e+05	71981.610451	46.169181
min	2020.000000	1.400000e+04	15000.000000	0.000000
25%	2023.000000	1.000000e+05	101570.000000	0.000000
50%	2023.000000	1.410000e+05	141300.000000	0.000000
75%	2024.000000	1.900000e+05	185900.000000	100.000000
max	2024.000000	3.040000e+07	800000.000000	100.000000

```
df.duplicated().sum()
```

```
5178
```

```
df.drop_duplicates(inplace = True)
```

```
df.head()
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	2024	SE	FT	AI Engineer	90000	USD	90000	AE	0	AE	L
1	2024	SE	FT	Machine Learning Engineer	205000	USD	205000	US	0	US	M
2	2024	SE	FT	Machine Learning Engineer	90000	USD	90000	US	0	US	M
3	2024	SE	FT	Machine Learning Engineer	150000	USD	150000	AU	0	AU	M
4	2024	SE	FT	Machine Learning Engineer	170000	USD	170000	AU	0	AU	M

```
df.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
```

```
print('---------- Number of Unique Values----------')
print('Column' , ' Number of Unique Values')
print('-------------------------------------')
for col in df.columns:
    unique_num = len(df[col].value_counts())
    print(col,'-'*(17-len(col)),unique_num)
print('-------------------------------------')
```

```
---------- Number of Unique Values----------
Column  Number of Unique Values
-------------------------------------
work_year ---- 4
experience_level ---- 4
employment_type ---- 4
job_title ---- 149
salary ---- 1
salary_currency ---- 18
salary_in_usd ---- 1
employee_residence ---- 88
remote_ratio ---- 3
company_location ---- 77
company_size ---- 3
-------------------------------------
```

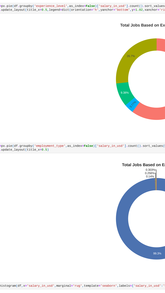
```
df['experience_level'] = df['experience_level'].replace({'EN' :'Entry-Level', 'MI' :'Mid-Level', 'SE' :'Executive Level', 'EX' :'Senior'}).astype(str).str.replace('nan', 'Unknown')
df['employment_type'] = df['employment_type'].replace({'PT' :'Part Time', 'FT' :'Full Time', 'CT' :'Contract', 'FL' :'Freelance'}).astype(str).str.replace('nan', 'Unknown')
```

```
plt.figure(figsize = (8,6))
sns.countplot(data = df, x = 'work_year')
plt.xlabel('Work Year')
plt.ylabel('Count')
plt.title('Work Distribution')
plt.show()
```

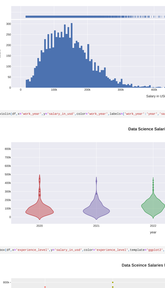

```
plt.figure(figsize = (8,6))
sns.countplot(data = df, x = 'experience_level')
plt.xlabel('Experience Level')
plt.ylabel('Count')
plt.title('Experience Distribution')
plt.show()
```


```
plt.figure(figsize = (8,6))
sns.countplot(data = df, x = 'employment_type')
plt.xlabel('Employment Type')
plt.ylabel('Count')
plt.title('Employment Distribution')
plt.show()
```

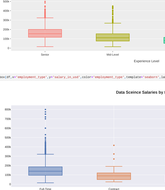

```
plt.figure(figsize=(8, 6))
plt.hist(df['job_title'], bins=20, edgecolor='black')
plt.ylabel('Frequency')
plt.xlabel('Job Title')
plt.title('Job Title Distribution')
plt.show()
```



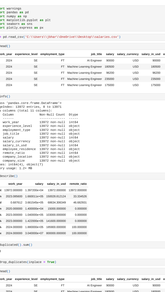

```
plt.figure(figsize=(8, 6))
plt.hist(df['salary'], bins=20, edgecolor='black')
plt.ylabel('Frequency')
plt.xlabel('Salary')
plt.ticklabel_format(style='plain')
plt.title('Salary Distribution')
plt.show()
```



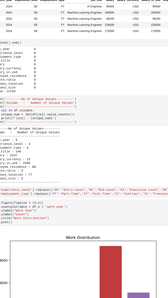

```
sns.set(rc={'figure.figsize':(15,8)})
fig=px.bar(x=df.job_title.value_counts().head(10).values,y=df.job_title.value_counts().index,text=df.job_title.value_counts().head(10).values,labels={'index':'Job title','y':'count'},color='count',template='seaborn',title='<b> Top 10 Popular Roles in Data Science</b>')
fig.show()
```



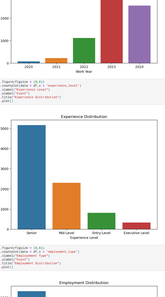

```
fig=px.bar(df.groupby('job_title').mean().sort_values(by='salary_in_usd',ascending=False).head(10),x='job_title',y='salary_in_usd',color='job_title')
fig.show()
```




```
df['job_title'] = df['job_title'].str.replace('Data Engineer', 'DE')
fig=px.bar(df.groupby('job_title').mean().sort_values(by='salary_in_usd',ascending=False).head(10),x='job_title',y='salary_in_usd',color='job_title',labels={'job_title':'Job title','salary_in_usd':'Avg salary in usd'},text='salary_in_usd',template='seaborn')
fig.update_traces(textfont_size=8)
fig.show()
```




```
fig=px.pie(df.groupby('experience_level').sum(),values='salary_in_usd',names=df['experience_level'].value_counts().index,color_discrete_sequence=px.colors.sequential.Rainbow_r,title='Total Jobs Based on Experience Level')
fig.update_layout(title_x=0.5,legend=dict(orientation="v", yanchor="top", y=0.8, xanchor="right", x=1.1))
fig.show()
```




```
fig=px.pie(df.groupby('employment_type').sum(),values='salary_in_usd',names=df['employment_type'].value_counts().index,color_discrete_sequence=px.colors.sequential.Rainbow_r,title='Total Jobs Based on Employee Level')
fig.update_layout(title_x=0.5,legend=dict(orientation="v", yanchor="top", y=0.8, xanchor="right", x=1.1))
fig.show()
```



```
px.histogram(df,x='salary_in_usd',marginal="rug",template='seaborn',labels={'salary_in_usd':'Salary in USD'},title='<b> Salary Distribution')
```



```
px.violin(df,x='work_year',y='salary_in_usd',color='work_year',labels={'work_year':'Year','salary_in_usd':'Salary in usd'},template='seaborn',title='<b>Data Science Salaries by year')
```



```
px.box(df,x='experience_level',y='salary_in_usd',color='experience_level',template='ggplot2',labels={'experience_level':'Experience Level','salary_in_usd':'Salary in usd'},title='<b>Data Science Salaries by Experience')
```



```
px.box(df,x='employment_type',y='salary_in_usd',color='employment_type',template='seaborn',labels={'employment_type':'Employment Type','salary_in_usd':'Salary in usd'},title='<b>Data Science Salaries by type of employee')
```