DATASET = SALARY DATA

Introduction:

In this project, we analyzed a dataset containing salary information for various job titles. Our goal was to explore the dataset, understand the distribution of salaries, and examine the average salaries across different job titles. We used Python libraries such as Pandas, Matplotlib, and Seaborn for data manipulation, visualization, and analysis.

Operations Performed:

1. Data Loading: We started by loading the dataset into a Pandas DataFrame using the `read_csv()` function.
2. Exploratory Data Analysis: We examined the dataset by displaying the first few rows (`head()`) and gathering information about the dataset (`info()`). We also checked for any missing values and duplicates in the data.
3. Salary Distribution Visualization: We visualized the distribution of salaries using a histogram. We used Matplotlib to create the histogram plot and added labels and a title to provide context to the chart.
4. Average Salary by Job Title: We grouped the data by job title using the `groupby()` function and calculated the average salary for each job title. We then created a bar plot using Matplotlib to display the average salary for each job title, with the x-axis representing the job titles and the y-axis representing the average salary. We rotated the x-axis labels to improve readability using `plt.xticks(rotation=90)` .

In [1]:

```
import numpy as np
import pandas as pd
df=pd.read_csv("C:\\Users\\Salary_Data.csv")
```

In [2]:

```
#first 10 datas from dataset?
df.head(10)
```

Out[2]:

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Female	Master's	Data Analyst	3.0	65000.0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0
3	36.0	Female	Bachelor's	Sales Associate	7.0	60000.0
4	52.0	Male	Master's	Director	20.0	200000.0
5	29.0	Male	Bachelor's	Marketing Analyst	2.0	55000.0
6	42.0	Female	Master's	Product Manager	12.0	120000.0
7	31.0	Male	Bachelor's	Sales Manager	4.0	80000.0
8	26.0	Female	Bachelor's	Marketing Coordinator	1.0	45000.0
9	38.0	Male	PhD	Senior Scientist	10.0	110000.0

In [3]:

```
#last 10 datas from database?
df.tail(10)
```

Out[3]:

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
6694	27.0	Male	High School	Digital Marketing Manager	2.0	40000.0
6695	33.0	Female	Bachelor's Degree	Content Marketing Manager	7.0	90000.0
6696	28.0	Male	PhD	Sales Representative	4.0	55000.0
6697	51.0	Female	Master's Degree	Senior Product Marketing Manager	19.0	190000.0
6698	37.0	Male	Bachelor's Degree	Junior Sales Representative	6.0	75000.0
6699	49.0	Female	PhD	Director of Marketing	20.0	200000.0
6700	32.0	Male	High School	Sales Associate	3.0	50000.0
6701	30.0	Female	Bachelor's Degree	Financial Manager	4.0	55000.0
6702	46.0	Male	Master's Degree	Marketing Manager	14.0	140000.0
6703	26.0	Female	High School	Sales Executive	1.0	35000.0

In [4]:

```
#to get information of dataset totally?
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6704 entries, 0 to 6703
Data columns (total 6 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Age                  6702 non-null   float64
 1   Gender               6702 non-null   object 
 2   Education Level      6701 non-null   object 
 3   Job Title            6702 non-null   object 
 4   Years of Experience  6701 non-null   float64
 5   Salary               6699 non-null   float64
dtypes: float64(3), object(3)
memory usage: 314.4+ KB
```

In [5]:

```
#to show the summary statistics of dataset df?
df.describe()
```

Out[5]:

	Age	Years of Experience	Salary
count	6702.000000	6701.000000	6699.000000
mean	33.620859	8.094687	115326.964771
std	7.614633	6.059003	52786.183911
min	21.000000	0.000000	350.000000
25%	28.000000	3.000000	70000.000000
50%	32.000000	7.000000	115000.000000
75%	38.000000	12.000000	160000.000000
max	62.000000	34.000000	250000.000000

In [6]:

```
#find null values in dataset df?the dataset has no null values;
a=pd.isnull(df)
a
```

Out[6]:

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	False
...	...	...	...	...	...	...
6699	False	False	False	False	False	False
6700	False	False	False	False	False	False
6701	False	False	False	False	False	False
6702	False	False	False	False	False	False
6703	False	False	False	False	False	False

6704 rows × 6 columns

In [7]:

```
#to find the location of the particular data in dataset df?
print(df.loc[2])
```

```
Age                                45.0
Gender                             Male
Education Level                     PhD
Job Title                Senior Manager
Years of Experience                15.0
Salary                         150000.0
Name: 2, dtype: object
```

In [8]:

```
# to find the sample of the datas by using sample formula?
df.sample()
```

Out[8]:

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
4745	33.0	Male	Master's Degree	Senior Data Scientist	8.0	120000.0

In [9]:

```
#to check the condition of the null values inside the datas by shorting sum ?
df.isnull().sum()
```

Out[9]:

```
Age                    2
Gender                 2
Education Level        3
Job Title              2
Years of Experience    3
Salary                 5
dtype: int64
```

In [10]:

```
df.memory_usage()
```

Out[10]:

```
Index                    128
Age                    53632
Gender                 53632
Education Level        53632
Job Title              53632
Years of Experience    53632
Salary                 53632
dtype: int64
```

In [11]:

```
#drop_duplicates is used to find thd duplicates in the dataset?
df.drop_duplicates()
```

Out[11]:

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Female	Master's	Data Analyst	3.0	65000.0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0
3	36.0	Female	Bachelor's	Sales Associate	7.0	60000.0
4	52.0	Male	Master's	Director	20.0	200000.0
...	...	...	...	...	...	...
6623	43.0	Female	Master's Degree	Digital Marketing Manager	15.0	150000.0
6624	27.0	Male	High School	Sales Manager	2.0	40000.0
6625	33.0	Female	Bachelor's Degree	Director of Marketing	8.0	80000.0
6628	37.0	Male	Bachelor's Degree	Sales Director	7.0	90000.0
6631	30.0	Female	Bachelor's Degree	Sales Manager	5.0	70000.0

1792 rows × 6 columns

In [12]:

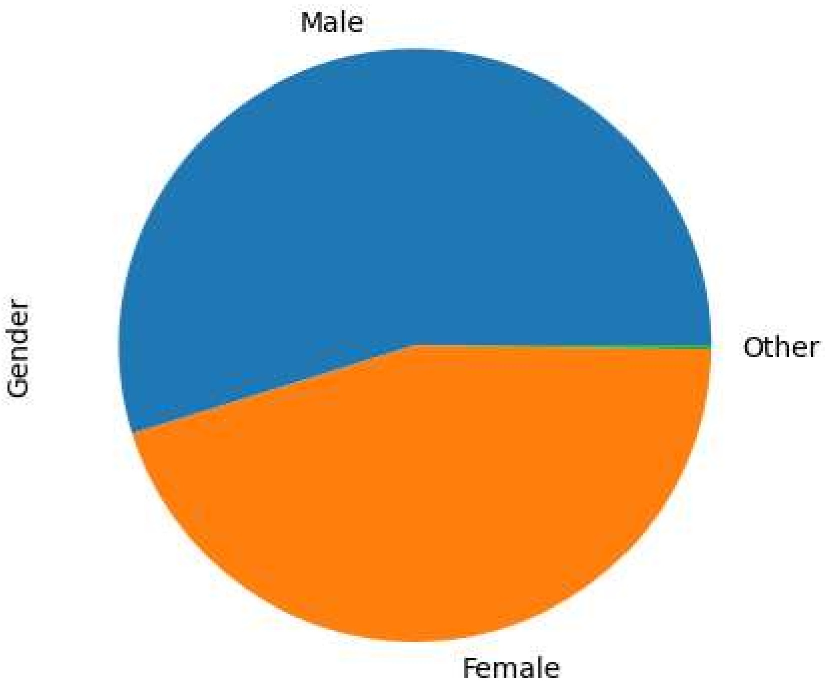
```
import pandas as pd
import matplotlib.pyplot as plt
```

In [13]:

```
#to visualize the data set using matplotlip showing gender varaition in data?
df.Gender.value_counts().plot(kind='pie')
```

Out[13]:

```
<Axes: ylabel='Gender'>
```



In [14]:

```
# to find the values of datas?
df.value_counts()
```

Out[14]:

```
Age   Gender  Education Level    Job Title          Years of Experience  Salary
24.0  Female  High School        Receptionist       0.0                  25000.0     45
27.0  Male    Bachelor's Degree  Software Engineer  3.0                  80000.0     45
32.0  Male    Bachelor's Degree  Product Manager    7.0                  120000.0    45
              Bachelor's         Software Engineer  8.0                  190000.0    39
33.0  Female  Master's           Product Manager    11.0                 198000.0    38
                                                                                     ..
26.0  Female  Bachelor's         Data Analyst       3.0                  120000.0     1
34.0  Female  High School        Sales Executive    5.0                  70000.0      1
              Master's           Business Analyst   5.0                  80000.0      1
                                 Financial Advisor  10.0                 95000.0      1
35.0  Male    PhD                Data Scientist     9.0                  112000.0     1
Length: 1787, dtype: int64
```

In [15]:

```
#using groupby() we can dataframe the 1 or more column in table database?
df.groupby(by="Job Title").Salary
```

Out[15]:

```
<pandas.core.groupby.generic.SeriesGroupBy object at 0x000001FBE03EFB80>
```

In [16]:

```
#its function used returns the values and filled with boolean values truu if value miss;if false the values filled?
df.isna().all()
```

Out[16]:

```
Age                    False
Gender                 False
Education Level        False
Job Title              False
Years of Experience    False
Salary                 False
dtype: bool
```

In [17]:

```
#index is inbuilt function in python searches the given elements from start to end from the list or data ?
df.index
```

Out[17]:

```
RangeIndex(start=0, stop=6704, step=1)
```

In [18]:

```
df[12:17]
```

Out[18]:

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
12	35.0	Male	Bachelor's	Financial Analyst	6.0	65000.0
13	40.0	Female	Master's	Project Manager	14.0	130000.0
14	27.0	Male	Bachelor's	Customer Service Rep	2.0	40000.0
15	44.0	Male	Bachelor's	Operations Manager	16.0	125000.0
16	33.0	Female	Master's	Marketing Manager	7.0	90000.0

In [19]:

```
import numpy as np
import pandas as pd
```

In [20]:

```
# use rename function and can we change the column name from the table:
```

In [21]:

```
df.head()
```

Out[21]:

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Female	Master's	Data Analyst	3.0	65000.0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0
3	36.0	Female	Bachelor's	Sales Associate	7.0	60000.0
4	52.0	Male	Master's	Director	20.0	200000.0

In [22]:

```
df.to_csv("Salary_Data.csv",index=False)
df.dtypes
```

Out[22]:

```
Age                    float64
Gender                  object
Education Level         object
Job Title               object
Years of Experience    float64
Salary                 float64
dtype: object
```

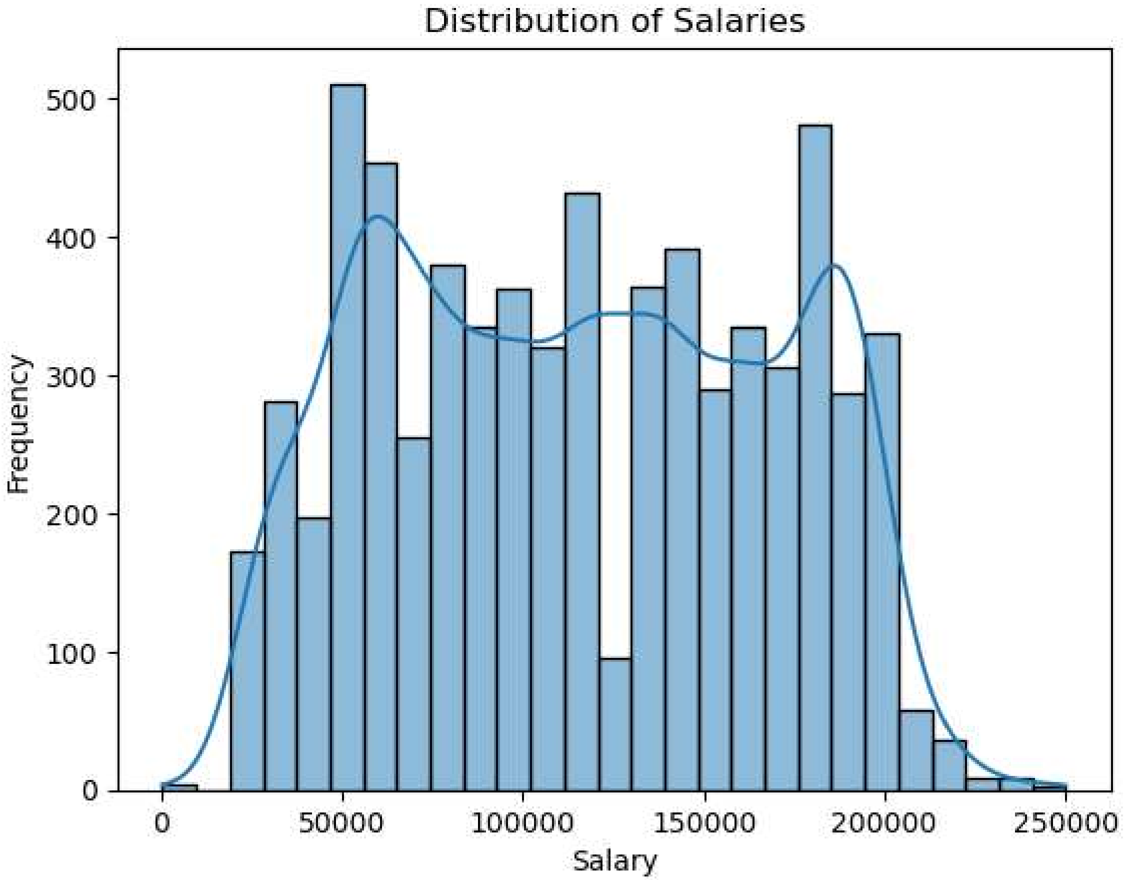
In [35]:

```
import seaborn as sns
import matplotlib.pyplot as plt

# Create the distribution plot using Seaborn
sns.histplot(data=df, x='Salary', kde=True)

# Set the labels and title
plt.xlabel('Salary')
plt.ylabel('Frequency')
plt.title('Distribution of Salaries')

# Display the plot
plt.show()
```



Conclusion:

In this project, we analyzed the salary dataset to gain insights into salary distributions and average salaries across different job titles. We observed the distribution of salaries using a histogram and found that it was slightly right-skewed. We also identified the average salary for each job title and visualized it using a bar plot. This analysis provides a valuable understanding of salary patterns and can be used to make informed decisions regarding salary structures, job market competitiveness, and employee compensation.

```
In [ ]:
```