LAB 1: Exploratory Data Analysis and Data Visualization in Python

In [1]:

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

In [3]:

```
df = pd.read_csv(r"D:\SO\Salary_Data.csv")
```

In [4]:

```
df.head()
```

Out[4]:

	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 [5]:

```
df.tail()
```

Out[5]:

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
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 [6]:

```
df.shape
```

Out[6]:

```
(6704, 6)
```

In [7]:

```
df.columns
```

Out[7]:

```
Index(['Age', 'Gender', 'Education Level', 'Job Title', 'Years of Experience',
       'Salary'],
      dtype='object')
```

In [8]:

```
df.dtypes
```

Out[8]:

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

In [9]:

```
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
```

Descriptive statastics

In [10]:

```
df.describe()
```

Out[10]:

	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 [11]:

```
df.describe(include = 'object')
```

Out[11]:

	Gender	Education Level	Job Title
count	6702	6701	6702
unique	3	7	193
top	Male	Bachelor's Degree	Software Engineer
freq	3674	2267	518

In [12]:

```
df['Gender'].value_counts()
```

Out[12]:

```
Gender
Male      3674
Female    3014
Other       14
Name: count, dtype: int64
```

In [13]:

```
df['Job Title'].value_counts()
```

Out[13]:

```
Job Title
Software Engineer                   518
Data Scientist                      453
Software Engineer Manager           376
Data Analyst                        363
Senior Project Engineer             318
                                   ...
Junior Social Media Specialist        1
Senior Software Architect             1
Developer                             1
Social M                              1
Social Media Man                      1
Name: count, Length: 193, dtype: int64
```

In [14]:

```
df['Gender'].value_counts(normalize=True)*100
```

Out[14]:

```
Gender
Male      54.819457
Female    44.971650
Other      0.208893
Name: proportion, dtype: float64
```

In [15]:

```
df.iloc[0:15,0:4]
```

Out[15]:

	Age	Gender	Education Level	Job Title
0	32.0	Male	Bachelor's	Software Engineer
1	28.0	Female	Master's	Data Analyst
2	45.0	Male	PhD	Senior Manager
3	36.0	Female	Bachelor's	Sales Associate
4	52.0	Male	Master's	Director
5	29.0	Male	Bachelor's	Marketing Analyst
6	42.0	Female	Master's	Product Manager
7	31.0	Male	Bachelor's	Sales Manager
8	26.0	Female	Bachelor's	Marketing Coordinator
9	38.0	Male	PhD	Senior Scientist
10	29.0	Male	Master's	Software Developer
11	48.0	Female	Bachelor's	HR Manager
12	35.0	Male	Bachelor's	Financial Analyst
13	40.0	Female	Master's	Project Manager

14	27.0	Male	Bachelor's	Customer Service Rep

In [16]:

```
df.loc[0:14, ['Age', 'Gender', 'Education Level']]
```

Out[16]:

	Age	Gender	Education Level
0	32.0	Male	Bachelor's
1	28.0	Female	Master's
2	45.0	Male	PhD
3	36.0	Female	Bachelor's
4	52.0	Male	Master's
5	29.0	Male	Bachelor's
6	42.0	Female	Master's
7	31.0	Male	Bachelor's
8	26.0	Female	Bachelor's
9	38.0	Male	PhD
10	29.0	Male	Master's
11	48.0	Female	Bachelor's
12	35.0	Male	Bachelor's
13	40.0	Female	Master's
14	27.0	Male	Bachelor's

In [17]:

```
df[df['Age'] == df[df['Gender'] == 'Male']['Age'].max()]['Job Title']
```

Out[17]:

```
1225    Software Engineer Manager
1236    Software Engineer Manager
1258    Software Engineer Manager
1304    Software Engineer Manager
1305    Software Engineer Manager
Name: Job Title, dtype: object
```

Sorting

In [18]:

```
df.sort_values(by='Job Title').head()
```

Out[18]:

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
46	32.0	Male	Bachelor's	Account Manager	5.0	75000.0
31	31.0	Female	Bachelor's	Accountant	4.0	55000.0
135	39.0	Female	Bachelor's	Administrative Assistant	10.0	55000.0
43	36.0	Female	Bachelor's	Administrative Assistant	8.0	45000.0
1933	34.0	Male	Master's Degree	Back end Developer	8.0	140000.0

In [19]:

```
df.sort_values(by='Age', ascending=False).head()
```

Out[19]:

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
1236	62.0	Male	PhD	Software Engineer Manager	20.0	200000.0
1305	62.0	Male	PhD	Software Engineer Manager	19.0	200000.0
1304	62.0	Male	PhD	Software Engineer Manager	20.0	200000.0
1258	62.0	Male	PhD	Software Engineer Manager	19.0	200000.0
1225	62.0	Male	PhD	Software Engineer Manager	19.0	200000.0

Replacing Values in columns

In [20]:

```
d = {'Male': 0, 'Female': 1}  # Create dictionary
print('Before replacement:')
print(df['Gender'].head())
df['Gender'] = df['Gender'].map(d)
print('After replacement:')
print(df['Gender'].head())
```

```
Before replacement:
0      Male
1    Female
2      Male
3    Female
4      Male
Name: Gender, dtype: object
After replacement:
0    0.0
1    1.0
2    0.0
3    1.0
4    0.0
Name: Gender, dtype: float64
```

In [21]:

```
df.groupby(by='Gender')['Age'].describe()
```

Out[21]:

	count	mean	std	min	25%	50%	75%	max
Gender								
0.0	3674.0	34.415895	7.977857	22.0	28.0	32.0	40.0	62.0
1.0	3014.0	32.624088	6.976065	21.0	28.0	31.0	36.0	60.0

In [22]:

```
pd.crosstab(df['Gender'], df['Education Level'], normalize=True)
```

Out[22]:

Education Level	Bachelor's	Bachelor's Degree	High School	Master's	Master's Degree	PhD	phD
Gender							
0.0	0.070585	0.202183	0.027666	0.013309	0.104980	0.130402	0.00015
1.0	0.042470	0.136833	0.037536	0.029759	0.129954	0.074174	0.00000

In [23]:

```
df.pivot_table(['Age', 'Salary'], ['Gender'], aggfunc='mean')
```

Out[23]:

Gender	Age	Salary
0.0	34.415895	121389.870915
1.0	32.624088	107888.998672

In [24]:

```
df.pivot_table(['Age', 'Salary'], ['Gender'], aggfunc='max')
```

Out[24]:

Gender	Age	Salary
0.0	62.0	250000.0
1.0	60.0	220000.0

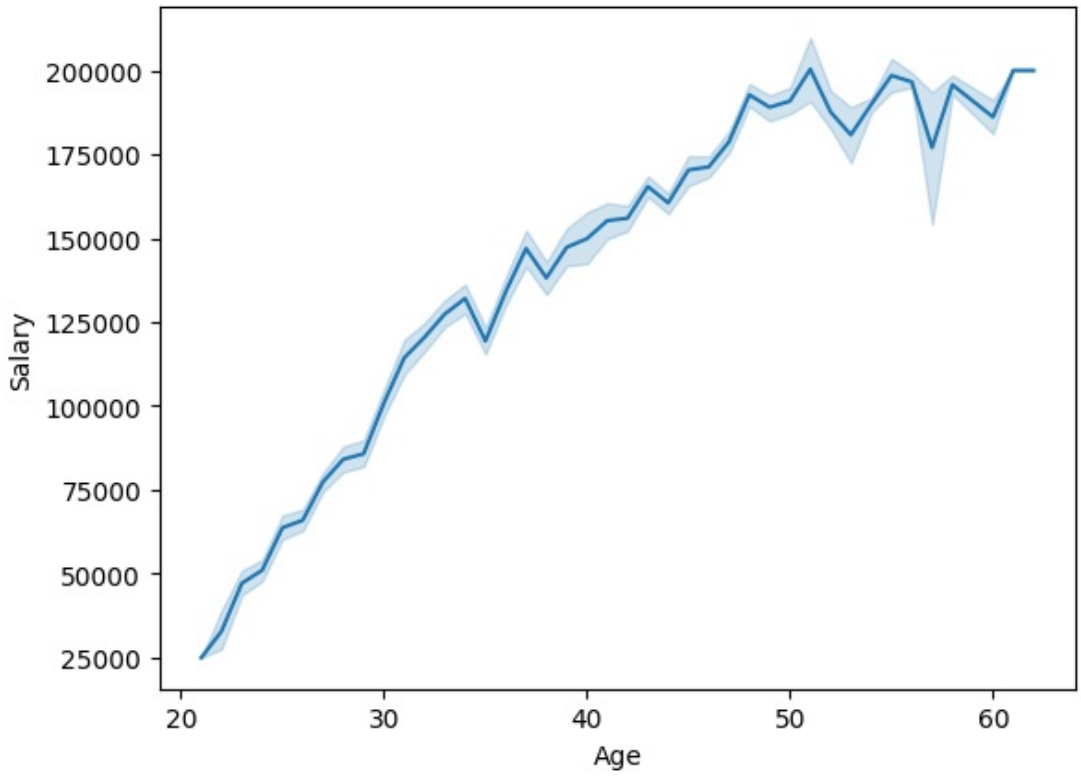
Data Visualization

In [25]:

```
sns.lineplot(x='Age', y='Salary', data=df)
```

Out[25]:

```
<Axes: xlabel='Age', ylabel='Salary'>
```

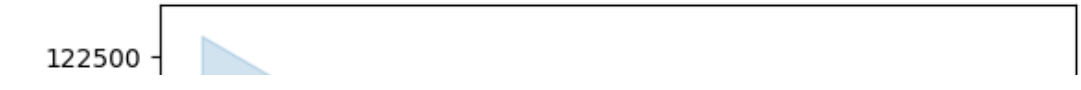


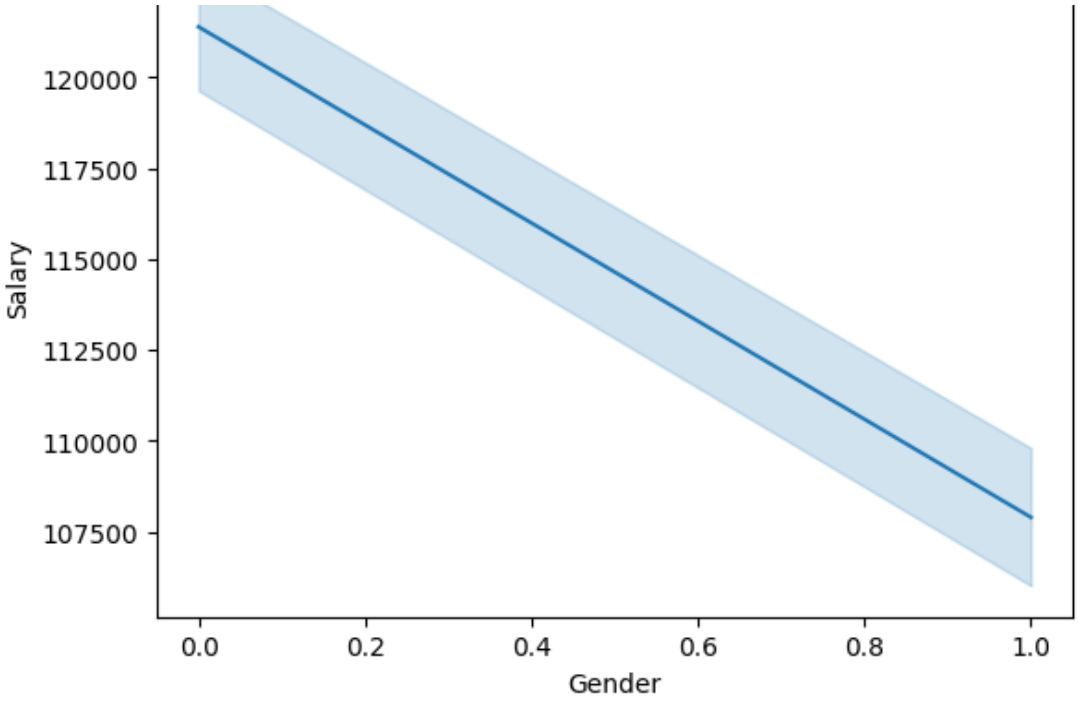
In [26]:

```
sns.lineplot(x='Gender', y='Salary', data=df)
```

Out[26]:

```
<Axes: xlabel='Gender', ylabel='Salary'>
```



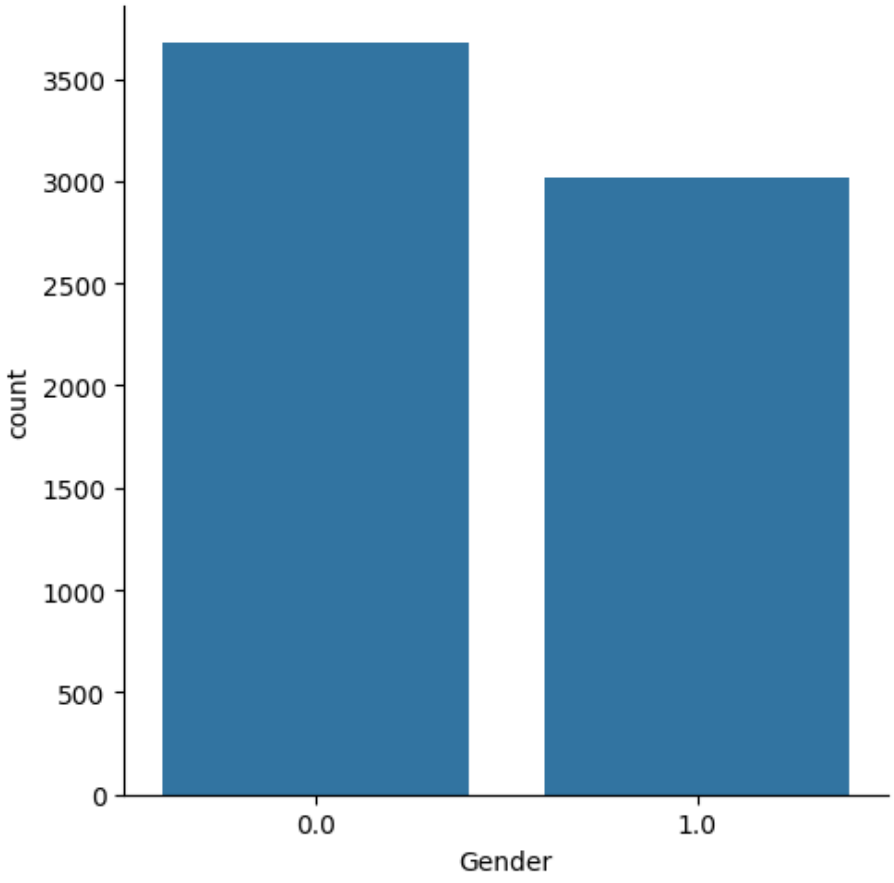


In [27]:

```
sns.catplot(x='Gender', data=df, kind='count')
```

Out[27]:

```
<seaborn.axisgrid.FacetGrid at 0x1d3b0027d40>
```

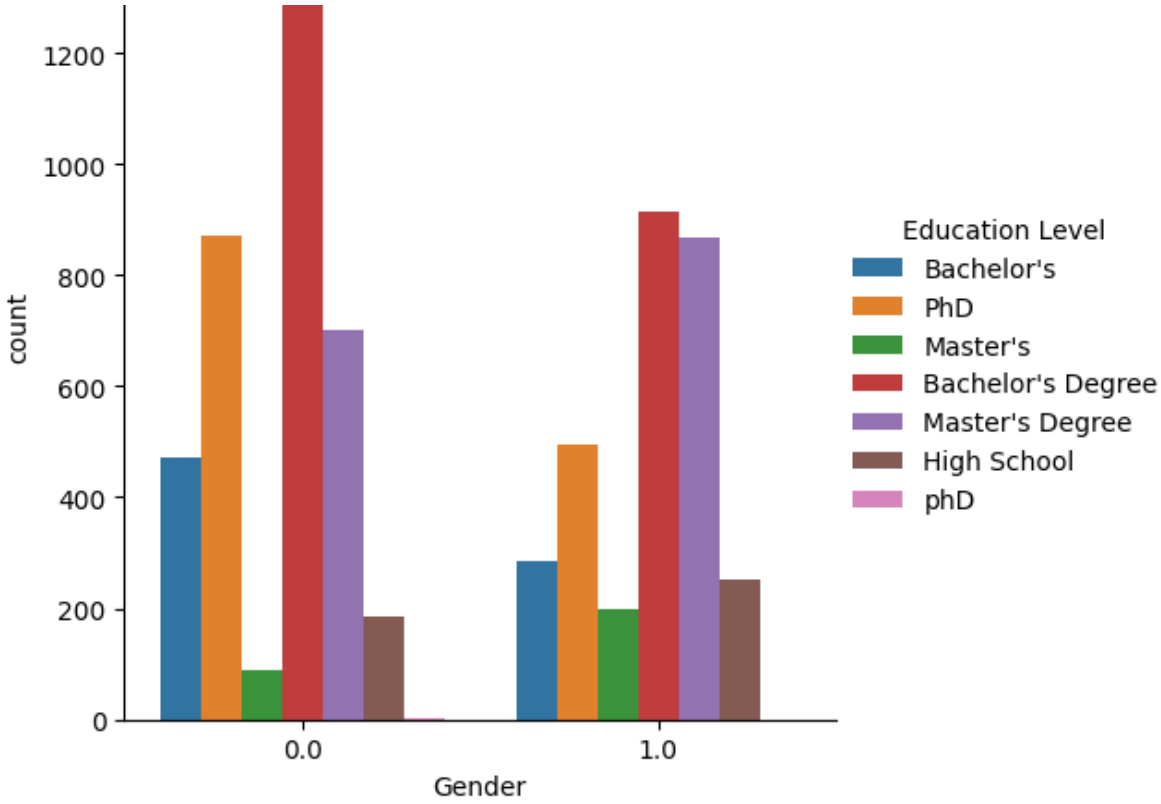


In [28]:

```
sns.catplot(x='Gender', data=df, kind='count', hue='Education Level')
```

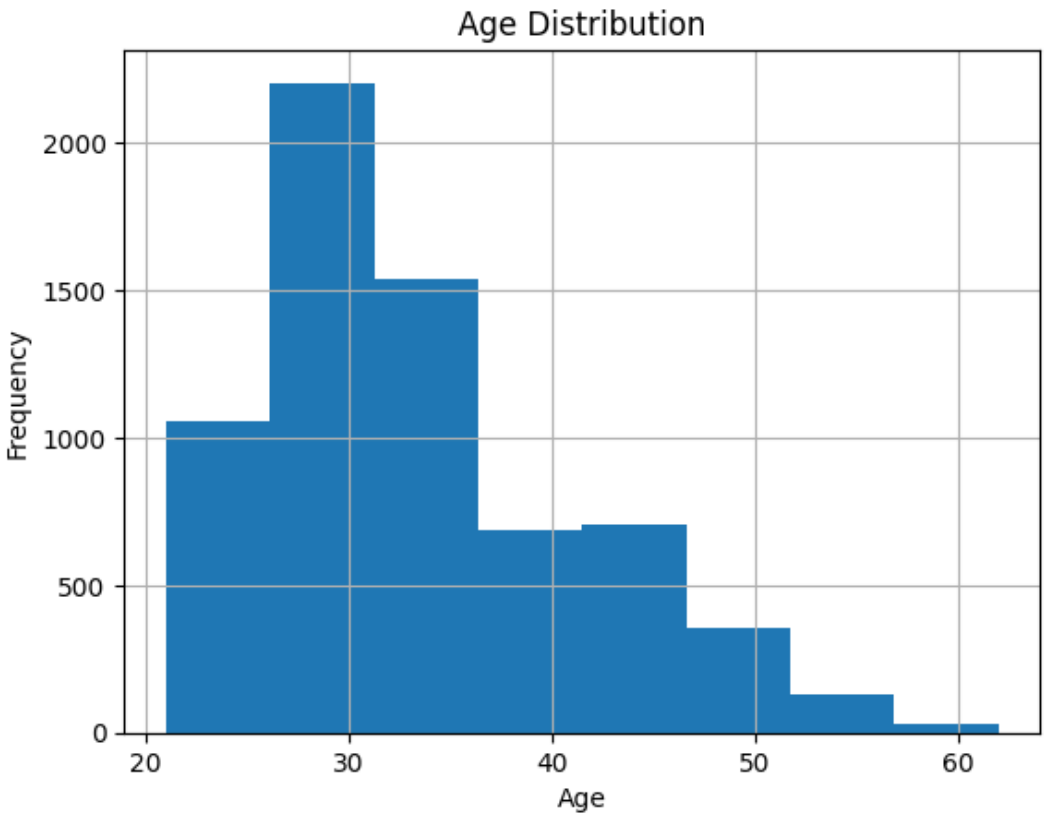
Out[28]:

```
<seaborn.axisgrid.FacetGrid at 0x1d3aff087d0>
```



In [29]:

```
df['Age'].hist(bins=8)
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.title('Age Distribution')
plt.show()
```



In [31]:

```
sizes = df["Gender"].value_counts()
fig1, ax1 = plt.subplots()
ax1.pie(sizes,
labels=sizes.index,
autopct='%1.2f%%',
shadow=True)
```

```
plt.title("Gender Distribution")
plt.show()
```