In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv("Employee_Salary_Dataset.csv")
df2=pd.read_csv("IRIS.csv")
```

In [4]:
```
df
```

Out[4]:

	ID	Experience_Years	Age	Gender	Salary
0	1	5	28	Female	250000
1	2	1	21	Male	50000
2	3	3	23	Female	170000
3	4	2	22	Male	25000
4	5	1	17	Male	10000
5	6	25	62	Male	5001000
6	7	19	54	Female	800000
7	8	2	21	Female	9000
8	9	10	36	Female	61500
9	10	15	54	Female	650000
10	11	4	26	Female	250000
11	12	6	29	Male	1400000
12	13	14	39	Male	6000550
13	14	11	40	Male	220100
14	15	2	23	Male	7500
15	16	2	21	Male	15000
16	17	10	34	Female	930000
17	18	15	54	Female	7900000
18	19	2	21	Male	15000
19	20	10	36	Male	330000
20	21	15	54	Male	6570000
21	22	4	26	Male	25000
22	23	5	29	Male	6945000
23	24	1	21	Female	6000
24	25	4	23	Female	8900
25	26	3	22	Female	20000
26	27	1	18	Male	3000
27	28	27	62	Female	10000000
28	29	19	54	Female	5000000
29	30	2	21	Female	6100
30	31	10	34	Male	80000
31	32	15	54	Male	900000
32	33	25	55	Female	1540000
33	34	19	53	Female	9300000
34	35	16	49	Male	7600000

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

Out[5]:

	ID	Experience_Years	Age	Salary
count	35.000000	35.00000	35.000000	3.500000e+01
mean	18.000000	9.20000	36.485714	2.059147e+06
std	10.246951	7.55236	14.643052	3.170124e+06
min	1.000000	1.00000	17.000000	3.000000e+03
25%	9.500000	2.50000	22.500000	2.250000e+04
50%	18.000000	6.00000	29.000000	2.500000e+05
75%	26.500000	15.00000	53.500000	3.270000e+06
max	35.000000	27.00000	62.000000	1.000000e+07

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35 entries, 0 to 34
Data columns (total 5 columns):
#   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
0   ID               35 non-null     int64 
1   Experience_Years 35 non-null     int64 
2   Age              35 non-null     int64 
3   Gender           35 non-null     object
4   Salary           35 non-null     int64 
dtypes: int64(4), object(1)
memory usage: 1.5+ KB
```

In [7]:
```
ggmf=ggmf.groupby("Gender")
ggmm=ggm.get_group("Male")
ggm
```

Out[7]:

	ID	Experience_Years	Age	Gender	Salary
1	2	1	21	Male	50000
3	4	2	22	Male	25000
4	5	1	17	Male	10000
5	6	25	62	Male	5001000
11	12	6	29	Male	1400000
12	13	14	39	Male	6000550
13	14	11	40	Male	220100
14	15	2	23	Male	7500
18	19	2	21	Male	15000
19	20	10	36	Male	330000
20	21	15	54	Male	6570000
21	22	4	26	Male	25000
22	23	5	29	Male	6945000
26	27	1	18	Male	3000
30	31	10	34	Male	80000
31	32	15	54	Male	900000
34	35	16	49	Male	7600000

In [8]:
```
ggf=gg.get_group('Female')
ggf
```

Out[8]:

	ID	Experience_Years	Age	Gender	Salary
0	1	5	28	Female	250000
2	3	3	23	Female	170000
6	7	19	54	Female	800000
7	8	2	21	Female	9000
8	9	10	36	Female	61500
9	10	15	54	Female	650000
10	11	4	26	Female	250000
15	16	4	27	Female	87000
16	17	10	34	Female	930000
17	18	15	54	Female	7900000
23	24	1	21	Female	6000
24	25	4	23	Female	8900
25	26	3	22	Female	20000
27	28	27	62	Female	10000000
28	29	19	54	Female	5000000
29	30	2	21	Female	6100
32	33	25	55	Female	1540000
33	34	19	53	Female	9300000

In [12]:
```
ggm.describe()
```

Out[12]:

	ID	Experience_Years	Age	Salary
count	16.000000	16.000000	16.000000	1.600000e+01
mean	19.277778	10.111111	37.111111	2.054617e+06
std	10.648422	8.123234	15.449698	3.405129e+06
min	1.000000	1.000000	21.000000	6.000000e+03
25%	9.250000	3.250000	23.000000	3.037500e+04
50%	17.500000	7.500000	31.000000	2.500000e+05
75%	27.500000	16.000000	54.000000	1.367500e+06
max	34.000000	27.000000	62.000000	1.000000e+07

In [13]:
```
ggf.describe()
```

Out[13]:

	ID	Experience_Years	Age	Salary
count	19.000000	19.000000	19.000000	1.900000e+01
mean	18.277778	10.111111	37.111111	2.054617e+06
std	10.648422	8.123234	15.449698	3.405129e+06
min	1.000000	1.000000	21.000000	6.000000e+03
25%	9.250000	3.250000	23.000000	3.037500e+04
50%	17.500000	7.500000	31.000000	2.500000e+05
75%	27.500000	16.000000	54.000000	1.367500e+06
max	34.000000	27.000000	62.000000	1.000000e+07

In [14]:
```
ggm["Experience_Years","Age","Salary"].mean()
```

Out[14]:
```
Experience_Years    1.0111111e+01
Age                 3.7111111e+01
Salary              2.0549167e+06
dtype: float64
```

In [15]:
```
ggf["Experience_Years","Age","Salary"].mean()
```

Out[15]:
```
Experience_Years    1.0111111e+01
Age                 3.7111111e+01
Salary              2.0549167e+06
dtype: float64
```

In [21]:
```
ggm["Experience_Years","Age","Salary"].median()
```

Out[21]:
```
Experience_Years      7.5
Age                  31.0
Salary           250000.0
dtype: float64
```

In [22]:
```
ggf["Experience_Years","Age","Salary"].median()
```

Out[22]:
```
Experience_Years      7.5
Age                  31.0
Salary           250000.0
dtype: float64
```

In [24]:
```
ggf["Experience_Years","Age","Salary"].max()
```

Out[24]:
```
Experience_Years         27
Age                      62
Salary             10000000
dtype: int64
```

In [46]:
```
ggm["Experience_Years","Age","Salary"].min()
```

Out[46]:
```
Experience_Years      1
Age                  17
Salary             3000
dtype: int64
```

In [28]:
```
ggm["Experience_Years","Age"].mode()
```

Out[28]:

	Experience_Years	Age
0	4	54.0
1	19	NaN

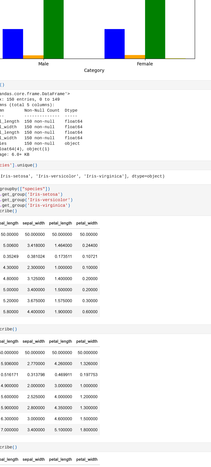
In [29]:
```
ggm["Age"].mode()
```

Out[29]:

	Age
0	54

In [30]:
```
ggmf.groupby("Gender")
ggm=gg.get_group('Male')
ggf=gg.get_group('Female')
salary_data = {'Male': ggm["Salary"].mean() , 'Female':ggf["Salary"].mean()}
category = list(salary_data.keys())
avg_salary = list(salary_data.values())
```

In [33]:
```
male_mean = ggm["Salary"].mean()
male_median = ggm["Salary"].median()
male_max = ggm["Salary"].max()
male_min = ggm["Salary"].min()

female_mean = ggf["Salary"].mean()
female_median = ggf["Salary"].median()
female_max = ggf["Salary"].max()
female_min = ggf["Salary"].min()

categories = ['Male', 'Female']
means = [male_mean, female_mean]
medians = [male_median, female_median]
max_salaries = [male_max, female_max]
min_salaries = [male_min, female_min]

plt.figure(figsize=(8, 6))
colors = ['blue', 'orange', 'green', 'yellow']

bar_width = 0.2
index = range(len(categories))

plt.bar(index, means, color=colors[0], width=bar_width, label='Mean Salary')
plt.bar([i + bar_width for i in index], medians, color=colors[1], width=bar_width, label='Median Salary')
plt.bar([i + 2*bar_width for i in index], max_salaries, color=colors[2], width=bar_width, label='Max Salary')
plt.bar([i + 3*bar_width for i in index], min_salaries, color=colors[3], width=bar_width, label='Min Salary')

plt.xticks([i + 1.5*bar_width for i in index], categories)
plt.xlabel('Category')
plt.ylabel('Salary')
plt.legend()
plt.title('Statistics of Salaries by Gender')

plt.show()
```



In [3]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

In [9]:
```
df2['species'].unique()
```

Out[9]:
```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

In [10]:
```
gp2=df2.groupby("species")
gps1=gp2.get_group('Iris-setosa')
gps2=gp2.get_group('Iris-versicolor')
gps3=gp2.get_group('Iris-virginica')
gps1.describe()
```

Out[10]:

	sepal_length	sepal_width	petal_length	petal_width
count	50.00000	50.000000	50.000000	50.00000
mean	5.00600	3.418000	1.464000	0.24400
std	0.35249	0.381024	0.173511	0.10721
min	4.30000	2.300000	1.000000	0.10000
25%	4.80000	3.125000	1.400000	0.20000
50%	5.00000	3.400000	1.500000	0.20000
75%	5.20000	3.675000	1.575000	0.30000
max	5.80000	4.400000	1.900000	0.60000

In [12]:
```
gps2.describe()
```

Out[12]:

	sepal_length	sepal_width	petal_length	petal_width
count	50.000000	50.000000	50.000000	50.000000
mean	5.936000	2.770000	4.260000	1.326000
std	0.516171	0.313798	0.469911	0.197753
min	4.900000	2.000000	3.000000	1.000000
25%	5.600000	2.525000	4.000000	1.200000
50%	5.900000	2.800000	4.350000	1.300000
75%	6.300000	3.000000	4.600000	1.500000
max	7.000000	3.400000	5.100000	1.800000

In [14]:
```
gps3.describe()
```

Out[14]:

	sepal_length	sepal_width	petal_length	petal_width
count	50.00000	50.000000	50.000000	50.00000
mean	6.58800	2.974000	5.552000	2.02600
std	0.63588	0.322497	0.551895	0.27465
min	4.90000	2.200000	4.500000	1.40000
25%	6.22500	2.800000	5.100000	1.80000
50%	6.50000	3.000000	5.550000	2.00000

75%	6.90000	3.175000	5.875000	2.30000
max	7.90000	3.800000	6.900000	2.50000