Practical No. : 03

Perform the following operations on any open source dataset (e.g., data.csv)

1. Provide summary statistics (mean, median, minimum, maximum, standard deviation) for a dataset (age, income etc.) with numeric variables grouped by one of the qualitative (categorical) variable. For example, if your categorical variable is age groups and quantitative variable is income, then provide summary statistics of income grouped by the age groups. Create a list that contains a numeric value for each response to the categorical variable.
2. Write a Python program to display some basic statistical details like percentile, mean, standard deviation etc. of the species of 'Iris-setosa', 'Iris-versicolor' and 'Iris-versicolor' of iris.csvdataset.

Import all the required Python Libraries.

In [1]:
```
import pandas as pd
import numpy as np
```

Reading the dataset and loading into pandas dataframe

In [2]:
```
df = pd.read_csv("Employee_Salary_Dataset.csv")
```

In [3]:
```
df.head()
```

Out[3]:

	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

In [4]:
```
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
```

1) Provide summary statistics (mean, median, minimum, maximum, standard deviation) for a dataset (age, income etc.) with numeric variables grouped by one of the qualitative (categorical) variable. For example, if your categorical variable is age groups and quantitative variable is income, then provide summary statistics of income grouped by the age groups. Create a list that contains a numeric value for each response to the categorical variable.

In [5]:
```
df.groupby('Gender')['Salary'].describe()
```

Out[5]:

	count	mean	std	min	25%	50%	75%	max
Gender								
Female	18.0	2.054917e+06	3.450120e+06	6000.0	30375.0	250000.0	1387500.0	10000000.0
Male	17.0	2.063626e+06	2.950974e+06	3000.0	25000.0	220100.0	5001000.0	7600000.0

In [6]:
```
df.groupby('Gender')['Salary'].mean()
```

Out[6]:
```
Gender
Female    2.054917e+06
Male      2.063626e+06
Name: Salary, dtype: float64
```

In [7]:
```
df.groupby('Gender')['Salary'].median()
```

Out[7]:
```
Gender
Female    250000.0
Male      220100.0
Name: Salary, dtype: float64
```

In [8]:
```
df.groupby('Gender')['Salary'].std()
```

Out[8]:
```
Gender
Female    3.450120e+06
Male      2.950974e+06
Name: Salary, dtype: float64
```

In [9]:
```
df.groupby('Gender')['Salary'].min()
```

Out[9]:
```
Gender
Female    6000
Male      3000
Name: Salary, dtype: int64
```

In [10]:
```
df.groupby('Gender')['Salary'].max()
```

Out[10]:
```
Gender
Female    10000000
Male       7600000
Name: Salary, dtype: int64
```

In [11]:
```
df.groupby('Gender')['Salary'].quantile(0.25)
```

Out[11]:
```
Gender
Female    30375.0
Male      25000.0
Name: Salary, dtype: float64
```

In [12]:
```
df.groupby('Gender')['Salary'].quantile(0.50)
```

Out[12]:
```
Gender
Female    250000.0
Male      220100.0
Name: Salary, dtype: float64
```

In [13]:
```
df.groupby('Gender')['Salary'].quantile(0.75)
```

Out[13]:
```
Gender
Female    1387500.0
Male      5001000.0
Name: Salary, dtype: float64
```

Reading the dataset and loading into new pandas dataframe

In [14]:
```
df1 = pd.read_csv("iris dataset.csv")
```

In [15]:
```
df1.head()
```

Out[15]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [16]:
```
df1.shape
```

Out[16]:
```
(150, 5)
```

2) Write a Python program to display some basic statistical details like percentile, mean, standard deviation etc. of the species of ‘Iris-setosa’, ‘Iris-versicolor’ and ‘Iris-versicolor’ of iris.csvdataset.

In [17]:
```
df1[df1['species'] == "Iris-setosa"].describe()
```

Out[17]:

	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 [18]:
```
df1[df1['species'] == "Iris-versicolor"].describe()
```

Out[18]:

	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 [19]:
```
df1[df1['species'] == "Iris-virginica"].describe()
```

Out[19]:

	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

In [20]:
```
df1.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 [21]:
```
df1['species'].unique()
```

Out[21]:
```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

In [22]:
```
df1.groupby("species").mean()
```

Out[22]:

species	sepal_length	sepal_width	petal_length	petal_width
Iris-setosa	5.006	3.418	1.464	0.244
Iris-versicolor	5.936	2.770	4.260	1.326
Iris-virginica	6.588	2.974	5.552	2.026

In [23]:
```
df1.groupby('species').median()
```

Out[23]:

species	sepal_length	sepal_width	petal_length	petal_width
Iris-setosa	5.0	3.4	1.50	0.2
Iris-versicolor	5.9	2.8	4.35	1.3
Iris-virginica	6.5	3.0	5.55	2.0

In [24]:
```
df1.groupby('species').min()
```

Out[24]:

species	sepal_length	sepal_width	petal_length	petal_width
Iris-setosa	4.3	2.3	1.0	0.1
Iris-versicolor	4.9	2.0	3.0	1.0
Iris-virginica	4.9	2.2	4.5	1.4

In [25]:
```
df1.groupby('species').max()
```

Out[25]:

species	sepal_length	sepal_width	petal_length	petal_width
Iris-setosa	5.8	4.4	1.9	0.6
Iris-versicolor	7.0	3.4	5.1	1.8
Iris-virginica	7.9	3.8	6.9	2.5

In [26]:
```
df1.groupby('species').std()
```

Out[26]:

species	sepal_length	sepal_width	petal_length	petal_width
Iris-setosa	0.352490	0.381024	0.173511	0.107210
Iris-versicolor	0.516171	0.313798	0.469911	0.197753
Iris-virginica	0.635880	0.322497	0.551895	0.274650

In [27]:
```
df1.groupby('species').quantile(0.25)
```

Out[27]:

species	sepal_length	sepal_width	petal_length	petal_width
Iris-setosa	4.800	3.125	1.4	0.2
Iris-versicolor	5.600	2.525	4.0	1.2
Iris-virginica	6.225	2.800	5.1	1.8

In [28]:
```
df1.groupby('species').quantile(0.50)
```

Out[28]:

species	sepal_length	sepal_width	petal_length	petal_width
Iris-setosa	5.0	3.4	1.50	0.2
Iris-versicolor	5.9	2.8	4.35	1.3
Iris-virginica	6.5	3.0	5.55	2.0

In [29]:
```
df1.groupby('species').quantile(0.75)
```

Out[29]:

species	sepal_length	sepal_width	petal_length	petal_width
Iris-setosa	5.2	3.675	1.575	0.3
Iris-versicolor	6.3	3.000	4.600	1.5
Iris-virginica	6.9	3.175	5.875	2.3