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.csv dataset.**

OUTPUT:

```
In [3]: import pandas as pd
        import numpy as np

In [4]: iris=pd.read_csv('Iris.csv')

In [5]: iris.head()
```

Out[5]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	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 [6]: iris.describe()
```

Out[6]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

```
In [7]: iris.shape
Out[7]: (150, 5)

In [10]: iris.mean()
C:\Users\user\AppData\Local\Temp\i
me.mean is deprecated. In a future
ted. Select only valid columns or
  iris.mean()
Out[10]: SepalLengthCm    5.843333
SepalWidthCm     3.054000
PetalLengthCm    3.758667
PetalWidthCm     1.198667
dtype: float64

In [11]: iris.median()
C:\Users\user\AppData\Local\Temp\ipyk
ame.median is deprecated. In a future
ecated. Select only valid columns or
  iris.median()
Out[11]: SepalLengthCm    5.80
SepalWidthCm     3.00
PetalLengthCm    4.35
PetalWidthCm     1.30
dtype: float64

In [12]: iris.SepalLengthCm.mode()
Out[12]: 0    5.0
Name: SepalLengthCm, dtype: float64

In [13]: iris.SepalWidthCm.mode()
Out[13]: 0    3.0
Name: SepalWidthCm, dtype: float64

In [14]: iris.groupby(['Species']).count()
Out[14]:
```

Species	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
Iris-setosa	50	50	50	50
Iris-versicolor	50	50	50	50
Iris-virginica	50	50	50	50

```
In [15]: iris.Species.mode()

Out[15]: 0        Iris-setosa
         1    Iris-versicolor
         2     Iris-virginica
         Name: Species, dtype: object

In [17]: iris.SepalLengthCm.std()

Out[17]: 0.8280661279778629

In [18]: iris.SepalWidthCm.std()

Out[18]: 0.4335943113621737

In [19]: iris.PetalLengthCm.std()

Out[19]: 1.7644204199522617

In [20]: iris.PetalWidthCm.std()

Out[20]: 0.7631607417008414
```