Name : Suryal D . Khirade
Roll NO: T190424399
Assignment No :03

Basic Statistics - Measures of Central Tendencies and Variance Perform the following
operations on any open source dataset (eg. 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. Provide the codes with outputs and explain everything that you do in this
step

In [1]:
```
# Import the required libraries
import pandas as pd
import numpy as np
import sklearn
from sklearn import datasets
```

In [2]:
```
iris = datasets.load_iris()
iris
```

Out[2]:
```
{'data': array([[5.1, 3.5, 1.4, 0.2],
        [4.9, 3. , 1.4, 0.2],
        [4.7, 3.2, 1.3, 0.2],
        [4.6, 3.1, 1.5, 0.2],
        [5. , 3.6, 1.4, 0.2],
        [5.4, 3.9, 1.7, 0.4],
        [4.6, 3.4, 1.4, 0.3],
        [5. , 3.4, 1.5, 0.2],
        [4.4, 2.9, 1.4, 0.2],
        [4.9, 3.1, 1.5, 0.1],
        [5.4, 3.7, 1.5, 0.2],
        [4.8, 3.4, 1.6, 0.2],
        [4.8, 3. , 1.4, 0.1],
        [4.3, 3. , 1.1, 0.1],
        [5.8, 4. , 1.2, 0.2],
        [5.7, 4.4, 1.5, 0.4],
        [5.4, 3.9, 1.3, 0.4],
        [5.1, 3.5, 1.4, 0.3],
        [5.7, 3.8, 1.7, 0.3],
```

In [3]:
```
df = pd.DataFrame(iris['data'])
df.head()
```

Out[3]:

	0	1	2	3
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

In [4]:
```
df[4] = iris['target']
df.head()
```

Out[4]:

	0	1	2	3	4
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

In [5]:
```
# Adding column names
df.rename(columns = {0:'SepalLengthCm', 1:'SepalWidthCm', 2:'PetalLengthCm'
df.head()
```

Out[5]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

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

Out[6]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333	1.000000
std	0.828066	0.435866	1.765298	0.762238	0.819232
min	4.300000	2.000000	1.000000	0.100000	0.000000
25%	5.100000	2.800000	1.600000	0.300000	0.000000
50%	5.800000	3.000000	4.350000	1.300000	1.000000
75%	6.400000	3.300000	5.100000	1.800000	2.000000
max	7.900000	4.400000	6.900000	2.500000	2.000000

In [7]:
```
df.shape
```

Out[7]:
```
(150, 5)
```

MEAN

In [8]:
```
df.mean()
```

Out[8]:
```
SepalLengthCm    5.843333
SepalWidthCm     3.057333
PetalLengthCm    3.758000
PetalWidthCm     1.199333
Species          1.000000
dtype: float64
```

MEDIAN

In [10]:
```
df.median()
```

Out[10]:
```
SepalLengthCm    5.80
SepalWidthCm     3.00
PetalLengthCm    4.35
PetalWidthCm     1.30
Species          1.00
dtype: float64
```

MODE

In [11]:
```
# Calculated only for categorical data
df.Species.mode()
```

Out[11]:
```
0    0
1    1
2    2
Name: Species, dtype: int32
```

In [12]:
```
df.groupby(['Species']).count()
```

Out[12]:

Species	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	50	50	50	50
1	50	50	50	50
2	50	50	50	50

STANDARD DEVIATION

In [13]:
```
df.SepalLengthCm.std()
```

Out[13]:
```
0.8280661279778629
```

In [14]:
```
df.SepalWidthCm.std()
```

Out[14]:
```
0.435866284936698
```

In [15]:
```
df.PetalLengthCm.std()
```

Out[15]:
```
1.7652982332594667
```

In [16]:
```
df.PetalWidthCm.std()
```

Out[16]:
```
0.7622376689603465
```