DSBDAL Assignment 3 -Descriptive Statistics - Measures of Central Tendency and variability

Part 2

Importing required Libraries

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

Reading CSV File

```python
In [2]:  df=pd.read_csv('Iris.csv')
```

Data Preprocessing

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

Out[3]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

```python
In [4]:  print('Our data set contains {} rows and {} columns'.format(df.shape[0],df.sh
```

```
Our data set contains 150 rows and 6 columns
```

```python
In [5]:  df.isnull().sum().sort_values(ascending=False)
```

```
Out[5]:  Id               0
         SepalLengthCm    0
         SepalWidthCm     0
         PetalLengthCm    0
         PetalWidthCm     0
         Species          0
         dtype: int64
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             150 non-null    int64
 1   SepalLengthCm  150 non-null    float64
 2   SepalWidthCm   150 non-null    float64
```

```
   3   PetalLengthCm   150 non-null    float64
   4   PetalWidthCm    150 non-null    float64
   5   Species         150 non-null    object
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

In [7]:
```python
df.drop(['Id'],axis=1,inplace=True)
```

In [8]:
```python
df.describe()
```

Out[8]:

	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 [9]:
```python
df.value_counts("Species")
```

Out[9]:
```
Species
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
dtype: int64
```

Displaying basic stats by grouping according to species

In [11]:
```python
print('Iris-setosa')
setosa = df['Species'] == 'Iris-setosa'
print(df[setosa].describe())
print('\nIris-versicolor')
versicolor = df['Species'] == 'Iris-versicolor'
print(df[versicolor].describe())
print('\nIris-virginica')
virginica = df['Species'] == 'Iris-virginica'
print(df[virginica].describe())
```

```
Iris-setosa
       SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
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

Iris-versicolor
       SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
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


Iris-virginica
        SepalLengthCm   SepalWidthCm   PetalLengthCm   PetalWidthCm
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 [ ]: