ASSIGNMENT 3

Descriptive Statistics - Measures of Central Tendency and variablilty ¶

Perform the following operations on any open source dataset :

1. Provide summary statistics(mean,mode,median,min,min,standard deviation) for a dataset.
2. Provide basic statistical details like percentiles of the species 'Iris-Setosa', 'Iris-versicolor' and 'Iris-virginica'

> Importing Required Libraries, Loading the dataset

In [48]:
```
import pandas as pd
import numpy as np
```

In [49]:
```
df = pd.read_csv("Iris.csv")
```

In [50]:
```
df
```

Out[50]:

	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
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

> Data Preprocessing

In [51]:
```
#checks total size(rows*columns)
df.size
```

Out[51]:
```
900
```

In [66]:
```
#checks dimensions of the dataframe
df.shape
```

Out[66]:
```
(150, 6)
```

In [67]:
```
#checks the columns present
df.columns
```

Out[67]:
```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWid
thCm',
       'Species'],
      dtype='object')
```

In [68]:
```
# checks datatypes of each column
df.dtypes
```

Out[68]:
```
Id                int64
SepalLengthCm    float64
SepalWidthCm     float64
PetalLengthCm    float64
PetalWidthCm     float64
Species           object
dtype: object
```

In [69]:
```
#checks initial statistics
df.describe()
```

Out[69]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

In [70]:
```
#prints the information of the dataset
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 [71]:
```
df = df.drop(['Id'],axis=1)
```

In [72]:
```
df
```

Out[72]:

	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
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

Since 'Id' is not required, we drop it.

> Measures of Central Tendency

Measures of central tendency provide a way to summarize the central or typical value of a dataset. Here are some commonly used measures:

Mean: The mean is calculated by summing all the values in a dataset and dividing by the total number of values. It represents the average value of the dataset.

Median: The median is the middle value when the data is sorted in ascending or descending order. It divides the dataset into two equal halves, with 50% of the values above and 50% below.

Mode: The mode is the value that appears most frequently in the dataset. It can be used for both numerical and categorical data.

> Mean

In [73]:
```
np.mean(df['SepalLengthCm'])
```

Out[73]:
```
5.843333333333335
```

In [74]:
```
np.mean(df['SepalWidthCm'])
```

Out[74]:
```
3.0540000000000007
```

In [75]:
```
np.mean(df['PetalLengthCm'])
```

Out[75]:
```
3.7586666666666693
```

In [76]:
```
np.mean(df['PetalWidthCm'])
```

Out[76]:
```
1.1986666666666672
```

In [77]:
```
np.mean(df)
```

Out[77]:
```
SepalLengthCm    5.843333
SepalWidthCm     3.054000
PetalLengthCm    3.758667
PetalWidthCm     1.198667
dtype: float64
```

> Median

In [78]:
```
df.median()
```

Out[78]:
```
SepalLengthCm    5.80
SepalWidthCm     3.00
PetalLengthCm    4.35
PetalWidthCm     1.30
dtype: float64
```

> Mode

In [79]:
```
df.mode().iloc[0]
```

Out[79]:
```
SepalLengthCm             5
SepalWidthCm              3
PetalLengthCm           1.5
PetalWidthCm            0.2
Species         Iris-setosa
Name: 0, dtype: object
```

> Measures of Variabilty

Measures of variability provide information about the spread or dispersion of data points in a dataset. Here are two commonly used measures:

Range: The range is the difference between the maximum and minimum values in a dataset. It gives a simple indication of the overall spread of the data.

Standard Deviation: The standard deviation measures the average amount of variation or dispersion of data points from the mean. It provides a more precise measure of variability and takes into account the differences between individual data points and the mean.

These measures can help understand how spread out the data values are and provide insights into the distribution of the dataset. The range gives a quick overview, while the standard deviation provides a more comprehensive understanding of the variability.

> Standard Deviation

In [17]:
```
np.std(df['SepalLengthCm'])
```

Out[17]:
```
0.8253012917851409
```

In [18]:
```
np.std(df['SepalWidthCm'])
```

Out[18]:
```
0.4321465800705435
```

In [19]:
```
np.std(df['PetalLengthCm'])
```

Out[19]:
```
1.7585291834055201
```

In [20]:
```
np.std(df['PetalWidthCm'])
```

Out[20]:
```
0.760612618588172
```

In [21]:
```
np.std(df)
```

Out[21]:
```
Id               43.300308
SepalLengthCm     0.825301
SepalWidthCm      0.432147
PetalLengthCm     1.758529
PetalWidthCm      0.760613
dtype: float64
```

> Minimum

In [22]:
```
np.min(df['SepalLengthCm'])
```

Out[22]:
```
4.3
```

In [23]:
```
np.min(df['SepalWidthCm'])
```

Out[23]:
```
2.0
```

In [24]:
```
np.min(df['PetalLengthCm'])
```

Out[24]:
```
1.0
```

In [25]:
```
np.min(df['PetalWidthCm'])
```

Out[25]:
```
0.1
```

In [26]:
```
np.min(df)
```

Out[26]:
```
Id                        1
SepalLengthCm           4.3
SepalWidthCm              2
PetalLengthCm             1
PetalWidthCm            0.1
Species         Iris-setosa
dtype: object
```

> Maximum

In [27]:
```
np.max(df['SepalLengthCm'])
```

Out[27]:
```
7.9
```

In [28]:
```
np.max(df['SepalWidthCm'])
```

Out[28]:
```
4.4
```

In [29]:
```
np.max(df['PetalLengthCm'])
```

Out[29]:
```
6.9
```

In [30]:
```
np.max(df['PetalWidthCm'])
```

Out[30]:
```
2.5
```

In [31]:
```
np.max(df)
```

Out[31]:
```
Id                         150
SepalLengthCm              7.9
SepalWidthCm               4.4
PetalLengthCm              6.9
PetalWidthCm               2.5
Species         Iris-virginica
dtype: object
```

> Range

In [83]:
```
# Select only the numerical columns
numerical_columns = ['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'Pe
numerical_df = df[numerical_columns]

# Calculate the range for each numerical column
range_values = numerical_df.max() - numerical_df.min()
```

In [84]:
```
range_values
```

Out[84]:
```
SepalLengthCm    3.6
SepalWidthCm     2.4
PetalLengthCm    5.9
PetalWidthCm     2.4
dtype: float64
```

> Quantile

Quantiles help identify specific values that represent certain percentages of the data and also help understand the distribution and position of data points, while measures of variability provide information about the spread or dispersion of the data values.

In [32]:
```
df.quantile(0.25)
```

Out[32]:
```
Id               38.25
SepalLengthCm     5.10
SepalWidthCm      2.80
PetalLengthCm     1.60
PetalWidthCm      0.30
Name: 0.25, dtype: float64
```

In [33]:
```
df.quantile(0.50)
```

Out[33]:
```
Id               75.50
SepalLengthCm     5.80
SepalWidthCm      3.00
PetalLengthCm     4.35
PetalWidthCm      1.30
Name: 0.5, dtype: float64
```

In [34]:
```
df.quantile(0.75)
```

Out[34]:
```
Id              112.75
SepalLengthCm     6.40
SepalWidthCm      3.30
PetalLengthCm     5.10
PetalWidthCm      1.80
Name: 0.75, dtype: float64
```

> Features of 'Species'

In [36]:
```
print(df.groupby('Species').mean())
```

```
                    Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  \
Species
Iris-setosa       25.5          5.006         3.418          1.464
Iris-versicolor   75.5          5.936         2.770          4.260
Iris-virginica   125.5          6.588         2.974          5.552

                 PetalWidthCm
Species
Iris-setosa             0.244
Iris-versicolor         1.326
Iris-virginica          2.026
```

In [37]:
```
print(df.groupby('Species').min())
```

```
                   Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalW
idthCm
Species
Iris-setosa         1            4.3           2.3            1.0
0.1
Iris-versicolor    51            4.9           2.0            3.0
1.0
Iris-virginica    101            4.9           2.2            4.5
1.4
```

In [38]:
```
print(df.groupby('Species').max())
```

```
                   Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalW
idthCm
Species
Iris-setosa        50            5.8           4.4            1.9
0.6
Iris-versicolor   100            7.0           3.4            5.1
1.8
Iris-virginica    150            7.9           3.8            6.9
2.5
```

In [39]:
```
print(df.groupby('Species').median())
```

```
                    Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  \
Species
Iris-setosa       25.5            5.0           3.4           1.50
Iris-versicolor   75.5            5.9           2.8           4.35
Iris-virginica   125.5            6.5           3.0           5.55

                 PetalWidthCm
Species
Iris-setosa               0.2
Iris-versicolor           1.3
Iris-virginica            2.0
```

In [40]:
```
print(df.groupby('Species').std())
```

```
                      Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  \
Species
Iris-setosa      14.57738       0.352490      0.381024       0.173511
Iris-versicolor  14.57738       0.516171      0.313798       0.469911
Iris-virginica   14.57738       0.635880      0.322497       0.551895

                 PetalWidthCm
Species
Iris-setosa          0.107210
Iris-versicolor      0.197753
Iris-virginica       0.274650
```

In [41]:
```
val = pd.get_dummies(df.Species)
```

In [42]:
```
val
```

Out[42]:

	Iris-setosa	Iris-versicolor	Iris-virginica
0	1	0	0
1	1	0	0
2	1	0	0
3	1	0	0
4	1	0	0
...	...	...	...
145	0	0	1
146	0	0	1
147	0	0	1
148	0	0	1
149	0	0	1

150 rows × 3 columns

In [43]:
```
val.mean()
```

Out[43]:
```
Iris-setosa        0.333333
Iris-versicolor    0.333333
Iris-virginica     0.333333
dtype: float64
```

In [44]:
```
val.quantile(0.25)
```

Out[44]:
```
Iris-setosa        0.0
Iris-versicolor    0.0
Iris-virginica     0.0
Name: 0.25, dtype: float64
```

In [45]:
```
val.quantile(0.50)
```

Out[45]:
```
Iris-setosa        0.0
Iris-versicolor    0.0
Iris-virginica     0.0
Name: 0.5, dtype: float64
```

In [46]:
```
val.quantile(0.75)
```

Out[46]:
```
Iris-setosa        1.0
Iris-versicolor    1.0
Iris-virginica     1.0
Name: 0.75, dtype: float64
```

In [47]:
```
np.percentile(val,75)
```

Out[47]:
```
1.0
```

In [55]:
```
val.std()
```

Out[55]:
```
Iris-setosa        0.472984
Iris-versicolor    0.472984
Iris-virginica     0.472984
dtype: float64
```

In [ ]:
```

```