Assignment A1

Title - Analysis on Iris flower Dataset.

Problem statement :-

Download the iris flower dataset or any other dataset into a dataframe. Use Python / R and perform following :

1. How many features are there and what are their types ?
2. Compare & display summary statistics for each features available in dataset (e.g min, max, mean, std-dev, variance, percentile)
3. Data visualization - create a histogram for each feature in the dataset to illustrate feature distribution.
4. Create a box plot for each feature in the dataset. All of the box plots should be combined into a single plot. compare distributions and find outliers.

Objectives -

- To learn the concept & terminologies in data analytics.
- To learn how to display summary set statistics & charts for each feature.

Outcomes - We will be able to -

- learn the concepts in data analytics.
- learn how to summarize & plot charts.

Theory -

A) Iris flower dataset -

- The dataset is a multivariate dataset introduced by the British statician & bio chemist Ronald fisher in 1936.

- Dataset Consist of 50 samples from each of 3 species of Iris , which are Sentosa , virginica & versicolor.
- four features measured from each Sample are length and width of sepals & petals in mm.

B) Summary statistics :-

1. Mean :- It identifies the average value of set of values

$$\bar{x} = \frac{\Sigma x_i}{n} \quad \text{where } x_i = \text{value of attributes.} \quad n = \text{total no. of items}$$

2. Range - It shows the mathematical model between the lowest & highest values in the dataset , it measures the variability of dataset.

Range = Max - Min .

3. Standard deviation :- It measures the variability of dataset like range . The smaller standard deviation indicates less Variability.

$$\sigma = \sqrt{\frac{\Sigma (x_i - \bar{x})^2}{n}}$$

4. Variance - It measures ~~the~~ how far the data is spread ou

$$\sigma^2 = \frac{\overset{n}{\Sigma} (x_i - \bar{x})^2}{n}.$$

c) Applications -

1. Histogram -
- It is suitable for visualizing distribution of numeric data over a continuous interval or a certain time perio

- The histogram organises large amount of data & provides a visualization quickly, using a single dimension

2. Box plot -

- It allows quick graphical examination of one or more dataset. It may seem primitive than a histogram but they do have some advantages.
- They take up space & are particularly useful for comparing distributions between several groups of data.

3. Data visualization

- It quickly creates insightful data visuals.
- They allow anyone to organise & present information quickly

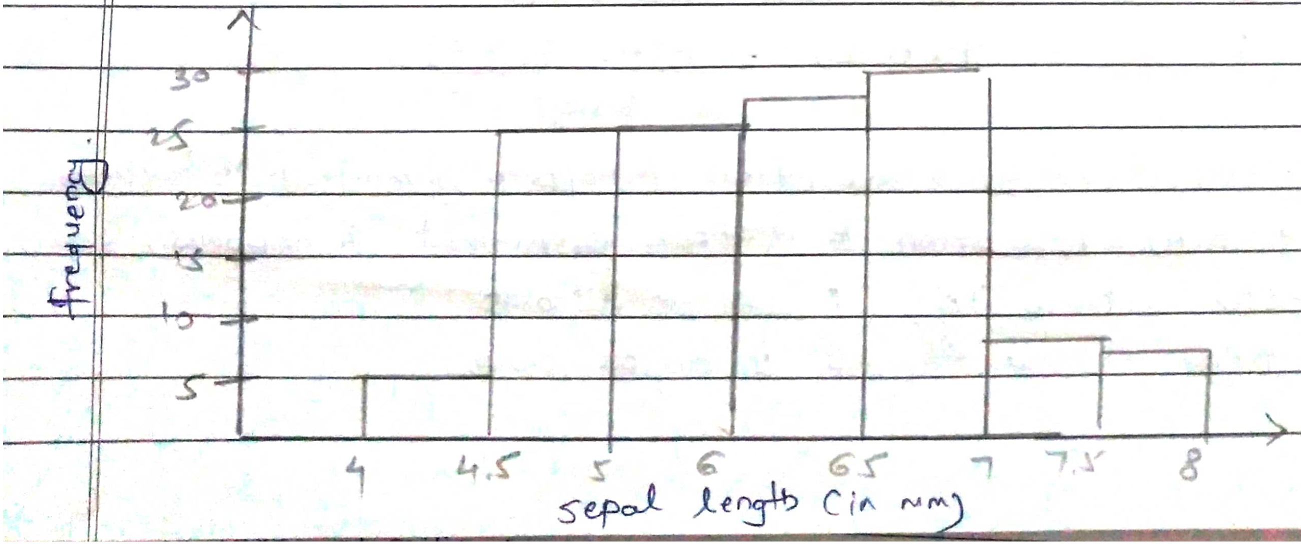
Conclusion -

Thus, we studied about concepts in data analytics & the dataset. we also presented the data in charts & box plots

Test case -

Input	Output
Column of sepal length	Mean = 5.843 mm.

Histogram of sepal length.



In [3]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [4]:

```
df=pd.read_csv(r"C:\Users\Viraj Shinde\Desktop\LP1\iris.data")
```

In [5]:

```
df.head()
```

Out[5]:

	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 [6]:

```
df.tail()
```

Out[6]:

	Sepal length	Sepal width	Petal length	Petal width	Species
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

In [7]:

```
X= df.drop('Species', axis = 1)
```

In [8]:

```
df.shape
list(df.columns)
```

Out[8]:

```
['Sepal length', 'Sepal width', 'Petal length', 'Petal width', 'Species']
```

In [9]:

```
df.dtypes
```

Out[9]:

```
Sepal length    float64
Sepal width     float64
Petal length    float64
Petal width     float64
Species          object
dtype: object
```

In [10]:
```
df['Sepal length'].describe()
```

Out[10]:
```
count    150.000000
mean       5.843333
std        0.828066
min        4.300000
25%        5.100000
50%        5.800000
75%        6.400000
max        7.900000
Name: Sepal length, dtype: float64
```

In [11]:
```
df['Sepal width'].describe()
```

Out[11]:
```
count    150.000000
mean       3.054000
std        0.433594
min        2.000000
25%        2.800000
50%        3.000000
75%        3.300000
max        4.400000
Name: Sepal width, dtype: float64
```

In [12]:
```
df['Petal length'].describe()
```

Out[12]:
```
count    150.000000
mean       3.758667
std        1.764420
min        1.000000
25%        1.600000
50%        4.350000
75%        5.100000
max        6.900000
Name: Petal length, dtype: float64
```

In [13]:
```
df['Petal width'].describe()
```

Out[13]:
```
count    150.000000
mean       1.198667
std        0.763161
min        0.100000
25%        0.300000
50%        1.300000
75%        1.800000
max        2.500000
Name: Petal width, dtype: float64
```

In [14]:
```
df['Species'].describe()
```

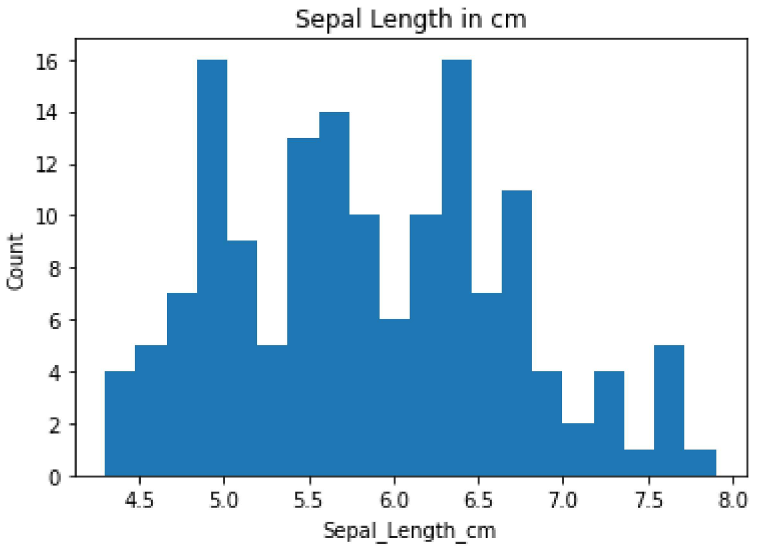
Out[14]:
```
count                 150
unique                  3
top       Iris-versicolor
freq                   50
Name: Species, dtype: object
```

In [15]:

```
x = df["Sepal length"]
plt.hist(x, bins = 20)
plt.title("Sepal Length in cm")
plt.xlabel("Sepal_Length_cm")
plt.ylabel("Count")
```

Out[15]:

```
Text(0, 0.5, 'Count')
```

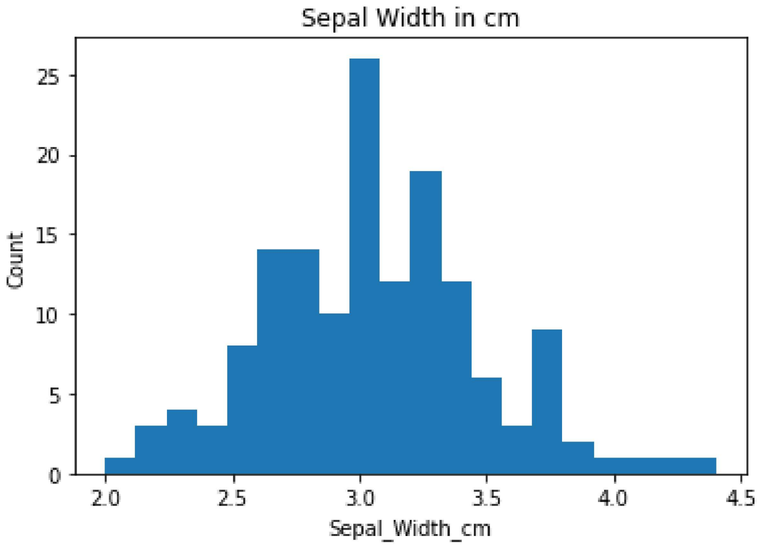


In [16]:

```
x = df["Sepal width"]
plt.hist(x, bins = 20)
plt.title("Sepal Width in cm")
plt.xlabel("Sepal_Width_cm")
plt.ylabel("Count")
```

Out[16]:

```
Text(0, 0.5, 'Count')
```

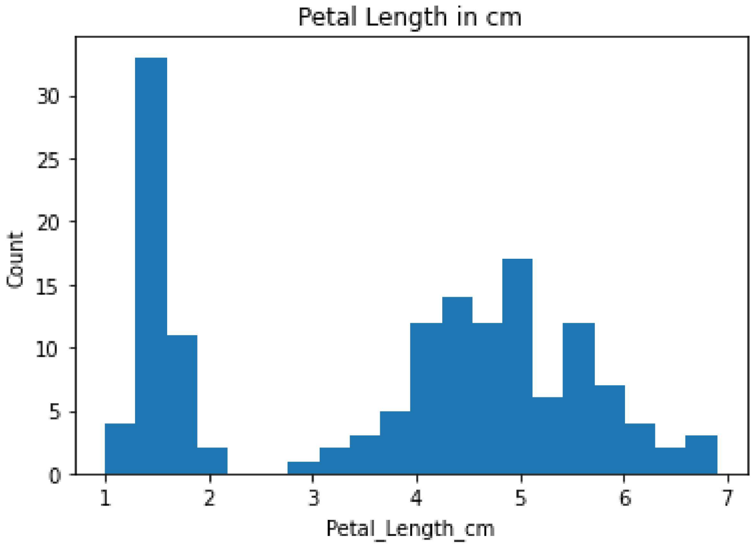


In [17]:

```
x = df["Petal length"]
plt.hist(x, bins = 20)
plt.title("Petal Length in cm")
plt.xlabel("Petal_Length_cm")
plt.ylabel("Count")
```

Out[17]:

```
Text(0, 0.5, 'Count')
```

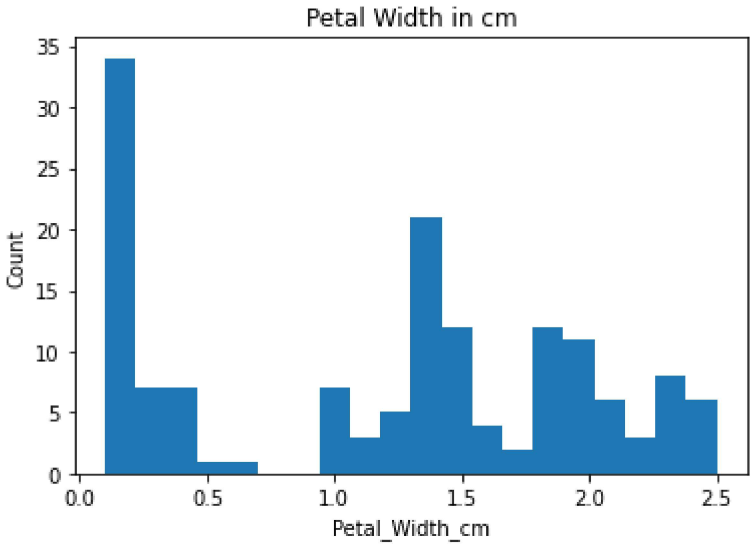


In [18]:

```
x = df["Petal width"]
plt.hist(x, bins = 20)
plt.title("Petal Width in cm")
plt.xlabel("Petal_Width_cm")
plt.ylabel("Count")
```

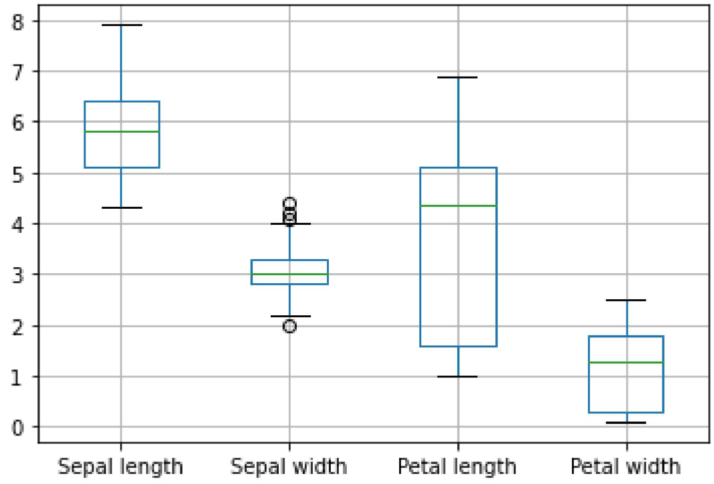
Out[18]:

```
Text(0, 0.5, 'Count')
```



```
In [19]: X.boxplot()
```

```
Out[19]: <AxesSubplot:>
```



```
In [20]: df.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 [ ]:
```