Department of Computer Science and Engineering

29th Batch

Lab Report 6

Course title	: Artificial Intelligence Lab
Course Code	: CSE - 414

Submitted By	Submitted To
Name : Md. Nahid Hasan ID : 221311131 Section : D Semester : 8th Batch : 29th	Name : Md. Mahfujur Rahman Designation : Lecturer, Varendra University, Rajshahi. Name : D.M. Asadujjaman Designation : Lecturer, Varendra University, Rajshahi.

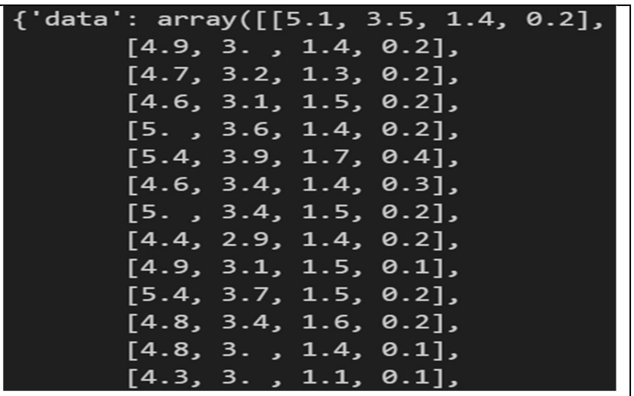
Signature

Signature

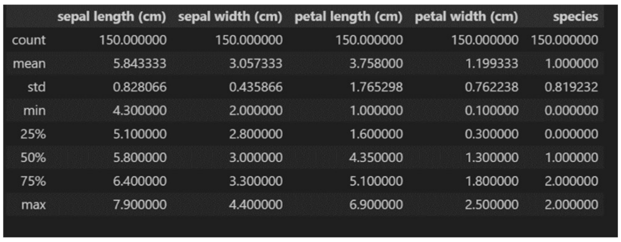
➢ Question: Data processing with pandas, scikit-learn, seaborn, matplotlib and implement 7 plots.

❖ Solution(Code & Output):

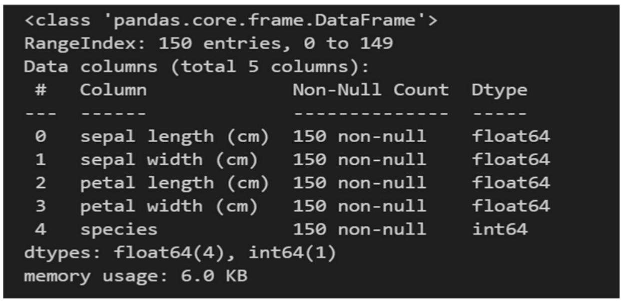
```
import pandas as pd
from sklearn.datasets import load_iris
import seaborn as sns
import matplotlib.pyplot as plt

#Load Data
iris = load_iris()
print(iris)

iris.feature_names
iris.data

df = pd.DataFrame(iris.data,
columns=iris.feature_names)
df['species'] = iris.target

df.head(10)
df.info()
df.shape
df.columns
df.describe()
print(df['species'].value_counts())
```



```
{'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],
```



	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	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

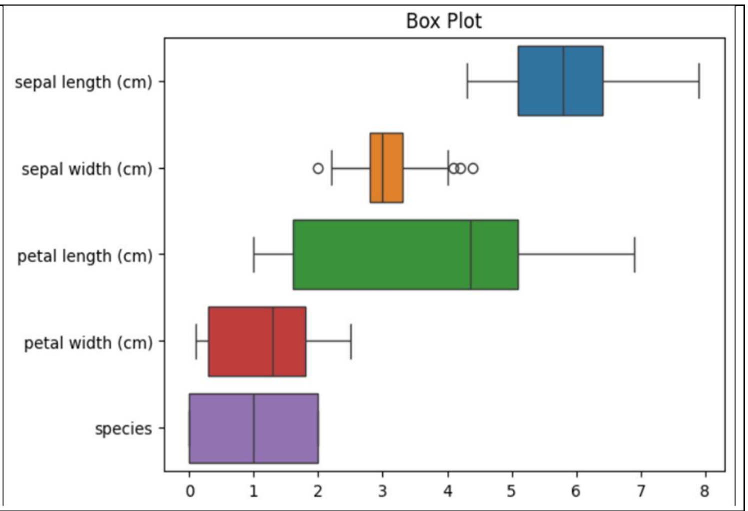


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   sepal length (cm)  150 non-null    float64
 1   sepal width (cm)   150 non-null    float64
 2   petal length (cm)  150 non-null    float64
 3   petal width (cm)   150 non-null    float64
 4   species            150 non-null    int64  
dtypes: float64(4), int64(1)
memory usage: 6.0 KB
```

Box Plot:

```
plt.figure(figsize=(15,10))
sns.boxplot(data=df, orient='h')
plt.title("Box Plot")
```

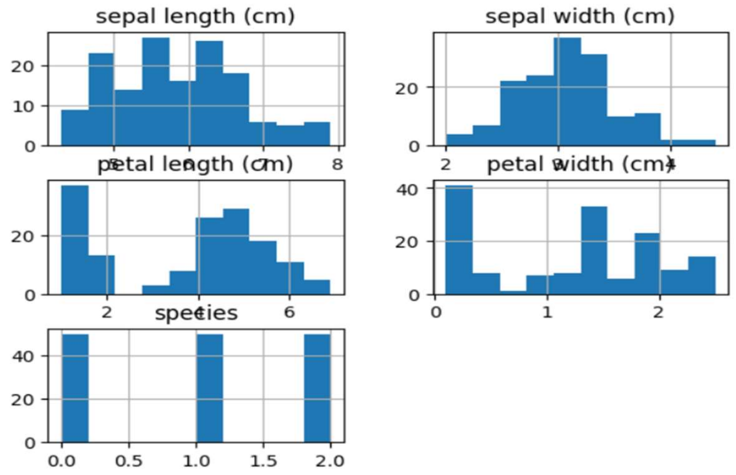
Shows data spread, median, and outliers across multiple variables. It's useful for comparing distributions side by side.



Histogram:

```
df.hist(figsize=(10,6))
plt.title('Histogram')
plt.show()
```

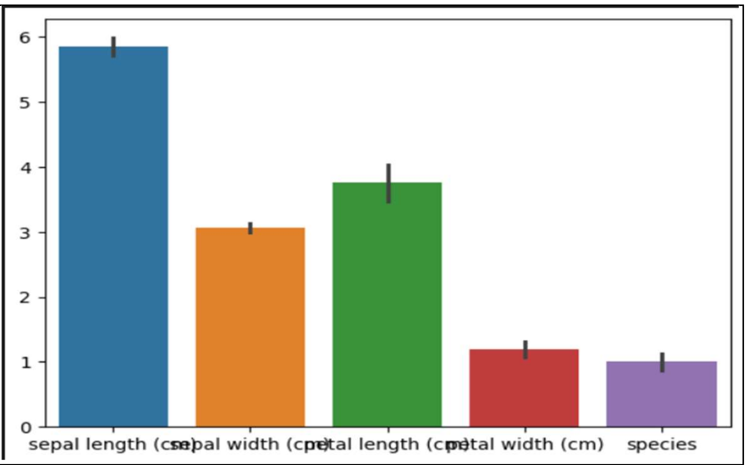
visualizes how often values occur within set ranges (bins), showing distribution shape like skewness or modality.



Bar Plot:

```
plt.figure(figsize=(15,10))
sns.boxplot(data=df, orient='h')
plt.title("Box Plot")
```

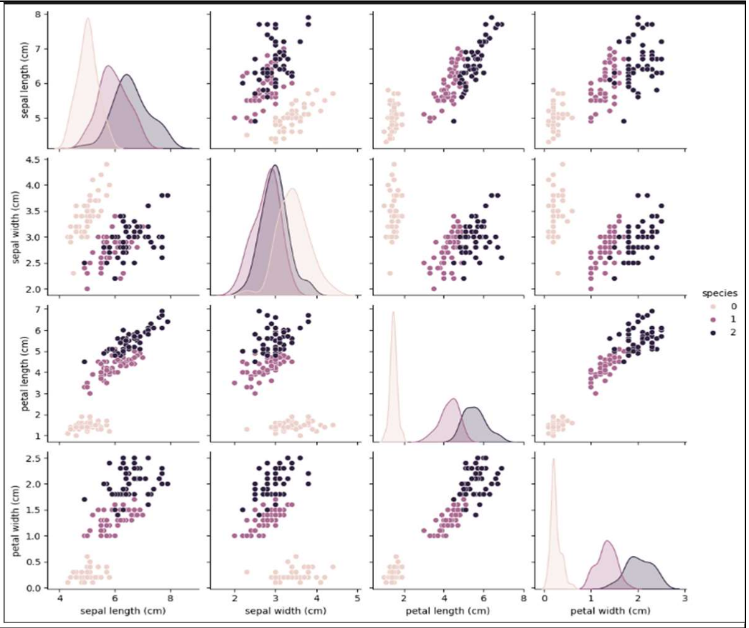
shows the mean (or another aggregation) of data for each category, with optional error bars.



Pair Plot:

```
sns.pairplot(df,hue='species')
plt.title(pair Plot')
plt.show()
```

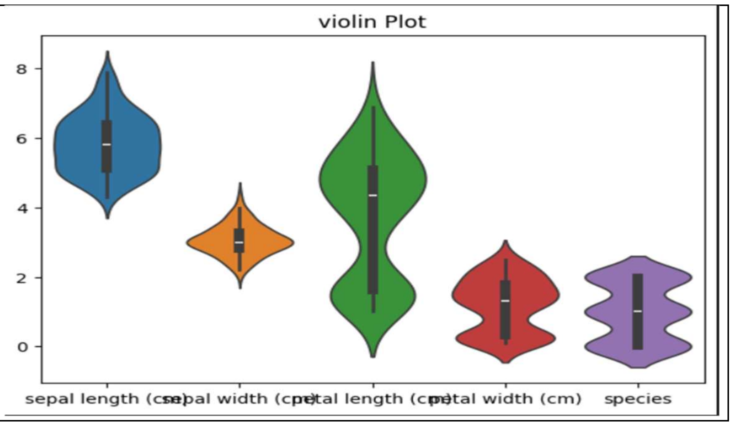
Pair Plot creates a grid of scatter plots for each variable pair, useful for spotting trends or clustering



Violin Plot:

```
sns.violinplot(df)
plt.title('violin Plot')
plt.show()
```

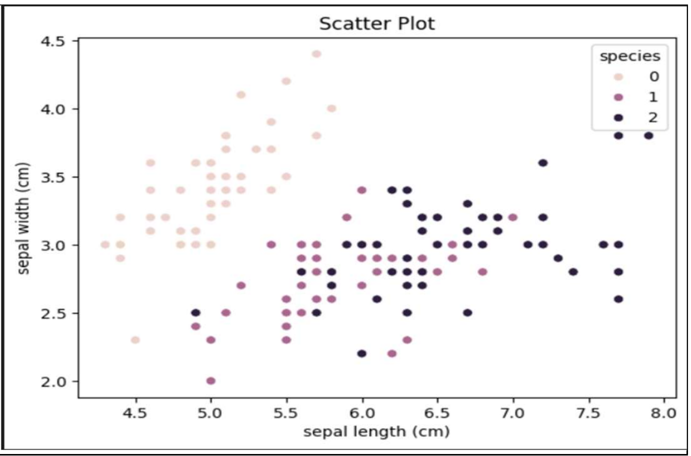
Violin Plot merges a box plot with a density plot, showing distribution shape and summary stats across categories.



Scatter Plot:

```
sns.scatterplot(x='sepal length (cm)',
y='sepal width (cm)', data=df)
plt.title("Scatter Plot")
plt.show()
```

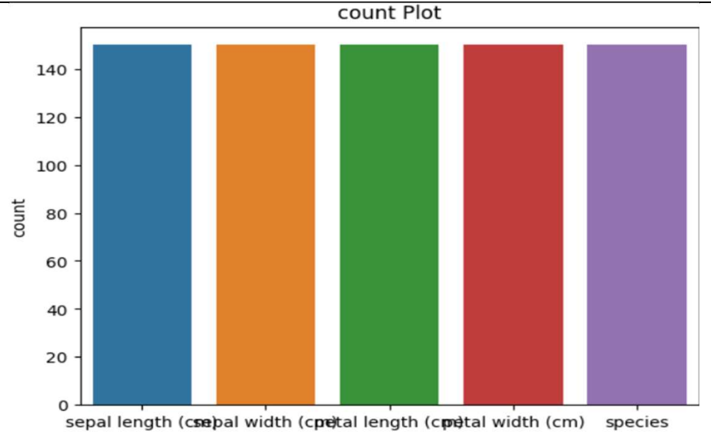
displays the relationship between two numeric variables. Adding color (hue) helps distinguish categories like species.



Count Plot:

```
sns.countplot(df)
plt.title('count Plot')
plt.show()
```

Shows data spread, median, and outliers across multiple variables. It's useful for comparing distributions side by side.



❖ Conclusion:

In this lab, I learned how to visualize data using different types of plots. It helped me understand patterns and relationships in the dataset more clearly. I practiced using pandas, seaborn, matplotlib, and scikit-learn for data analysis and visualization.