Artificial Intelligence and Machine Learning

LAB 1

Name: PUNEETH L

USN: 1BM24MC069

1. Consider a dataset and perform exploratory data analysis.

```
from sklearn.datasets import load_iris
import pandas as pd

iris = load_iris()

df = pd.DataFrame(data=iris.data, columns=iris.feature_names)

print(df.head())
```

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

i. Identify the dimension, structure, and summary of the data set

```
[2]: print(f"Shape of the dataset: {df.shape}")
     print(f"Dataset Structure: \n{df.info()}")
     print(f"Summary statistics: \n{df.describe()}")
```

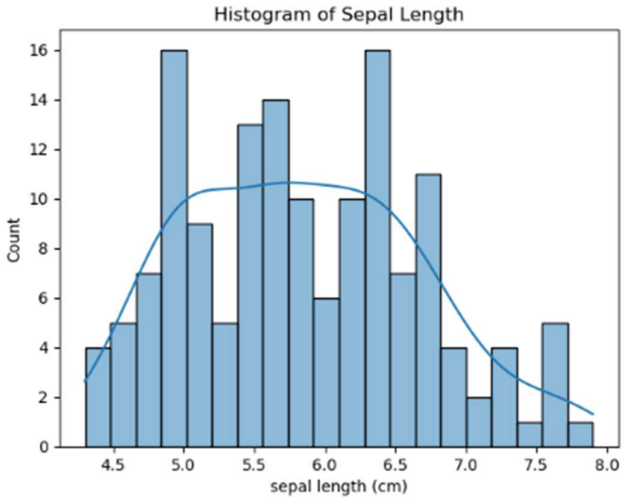
```
Shape of the dataset: (150, 4)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 4 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   sepal length (cm)  150 non-null    float64
 1   sepal width (cm)   150 non-null    float64
 2   petal length (cm)  150 non-null    float64
 3   petal width (cm)   150 non-null    float64
dtypes: float64(4)
memory usage: 4.8 KB
Dataset Structure:
None
Summary statistics:
       sepal length (cm)  sepal width (cm)  petal length (cm)  \
count         150.000000        150.000000         150.000000
mean            5.843333          3.057333           3.758000
std             0.828066          0.435866           1.765298
min             4.300000          2.000000           1.000000
25%             5.100000          2.800000           1.600000
50%             5.800000          3.000000           4.350000
75%             6.400000          3.300000           5.100000
max             7.900000          4.400000           6.900000

       petal width (cm)
count        150.000000
mean           1.199333
std            0.762238
min            0.100000
25%            0.300000
50%            1.300000
75%            1.800000
```

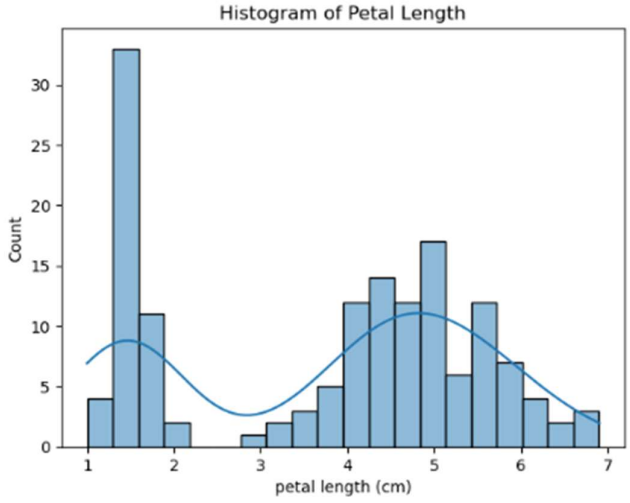
ii. Pre-process the dataset and treat them (like missing values, ‘na’?). Justify the treatment.

df = df.fillna(df.mean())

iii. Plot the histogram for continuous variables (at least two) to analyse the data.

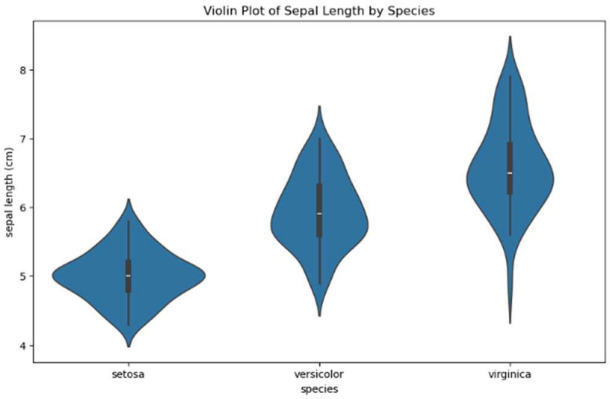
```
import seaborn as sns
import matplotlib.pyplot as plt

# Plot histogram for Sepal Length
sns.histplot(df['sepal length (cm)'], kde=True, bins=20)
plt.title('Histogram of Sepal Length')
plt.show()

# Plot histogram for Petal Length
sns.histplot(df['petal length (cm)'], kde=True, bins=20)
plt.title('Histogram of Petal Length')
plt.show()
```





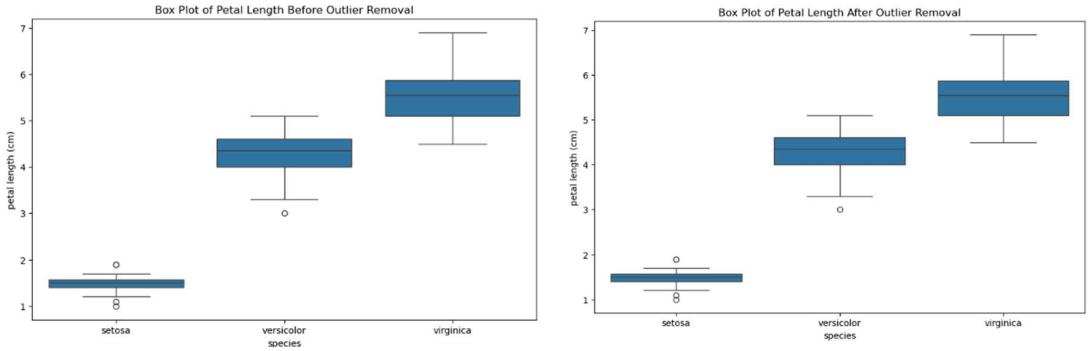
iv. Draw a violin plot do describe the distribution of a numerical variable to analyse the data.

```
plt.figure(figsize=(10, 6))
sns.violinplot(x='species', y='sepal length (cm)', data=df)
plt.title('Violin Plot of Sepal Length by Species')
plt.show()
```



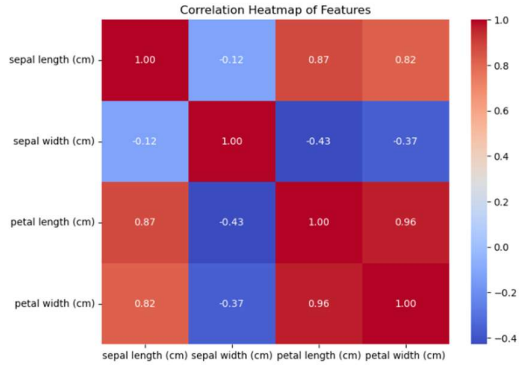
v. Recognize the outliers using box plot (Display the box plot before and after outlier treatment).

```
Q1 = df['petal length (cm)'].quantile(0.25)
Q3 = df['petal length (cm)'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

plt.figure(figsize=(10, 6))
sns.boxplot(x='species', y='petal length (cm)', data=df)
plt.title('Box Plot of Petal Length Before Outlier Removal')
plt.show()

df_filtered = df[(df['petal length (cm)'] >= lower_bound) & (df['petal length (cm)'] <= upper_bound)]

plt.figure(figsize=(10, 6))
sns.boxplot(x='species', y='petal length (cm)', data=df_filtered)
plt.title('Box Plot of Petal Length After Outlier Removal')
plt.show()
```



vi. Display a heat map to display the relationship among the attributes.

```
correlation_matrix = df.iloc[:, :-1].corr()

plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap of Features')
plt.show()
```



vii. Standardize the continuous variable (if any).

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

df_scaled = df.copy()
df_scaled.iloc[:, :-1] = scaler.fit_transform(df.iloc[:, :-1])

print(df_scaled.head())
```

```
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0          -0.900681          1.019004          -1.340227         -1.315444
1          -1.143017         -0.131979          -1.340227         -1.315444
2          -1.385353          0.328414          -1.397064         -1.315444
3          -1.506521          0.098217          -1.283389         -1.315444
4          -1.021849          1.249201          -1.340227         -1.315444

  species
0  setosa
1  setosa
2  setosa
3  setosa
4  setosa
```