

A1:Load a dataset using Pandas and perform basic data exploration tasks such as displaying the first few rows, summary statistics, and data visualization using matplotlib.

```
In [3]: # Import necessary Libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [17]: from sklearn.datasets import load_iris
iris = load_iris()
df=pd.DataFrame(iris.data, columns= iris.feature_names)
df['species'] = pd.Categorical.from_codes(iris.target, iris.target_names)
df
```

Out[17]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

```
In [5]: # Display basic properties of the dataset
print("Dataset Head:")
print(df.head()) # Display the first 5 rows of the dataset
```

Dataset 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
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

In [6]: `df.head(15)`

Out[6]:

	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
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
5	5.4	3.9	1.7	0.4
6	4.6	3.4	1.4	0.3
7	5.0	3.4	1.5	0.2
8	4.4	2.9	1.4	0.2
9	4.9	3.1	1.5	0.1
10	5.4	3.7	1.5	0.2
11	4.8	3.4	1.6	0.2
12	4.8	3.0	1.4	0.1
13	4.3	3.0	1.1	0.1
14	5.8	4.0	1.2	0.2

In [7]: `df.tail(10)`

Out[7]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
140	6.7	3.1	5.6	2.4
141	6.9	3.1	5.1	2.3
142	5.8	2.7	5.1	1.9
143	6.8	3.2	5.9	2.3
144	6.7	3.3	5.7	2.5
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

In [8]:

```
print("\nDataset Info:")
print(df.info()) # Display summary of the dataset including column data types and non-null counts
```

Dataset Info:

```
<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
None
```

In [9]:

```
print("\nDataset Description:")
print(df.describe()) # Display statistical summary of numeric columns
```

Dataset Description:

	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
max	2.500000

```
In [10]: print("\nDataset Shape:")
print(df.shape) # Display the dimensions of the dataset (rows, columns)
```

Dataset Shape:
(150, 4)

```
In [11]: print("\nColumn Names:")
print(df.columns) # Display column names
```

Column Names:
Index(['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)',
 'petal width (cm)'],
 dtype='object')

```
In [12]: print("\nMissing Values:")
print(df.isnull().sum()) # Display the number of missing values per column
```

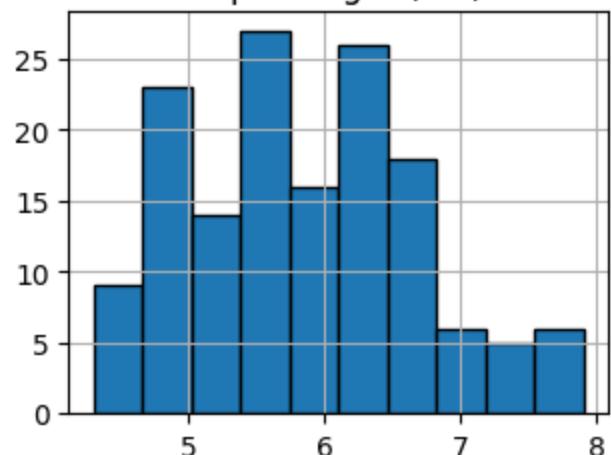
```
Missing Values:  
sepal length (cm)      0  
sepal width (cm)       0  
petal length (cm)      0  
petal width (cm)       0  
dtype: int64
```

```
In [15]: # 1. Histogram of Sepal Length
```

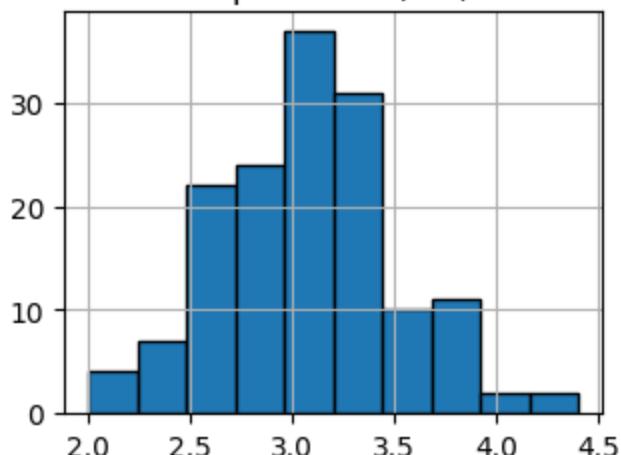
```
# 1. Histogram (feature distributions)  
df.hist(figsize=(8, 6), edgecolor='black')  
plt.suptitle("Feature Distributions", fontsize=14)  
plt.show()
```

Feature Distributions

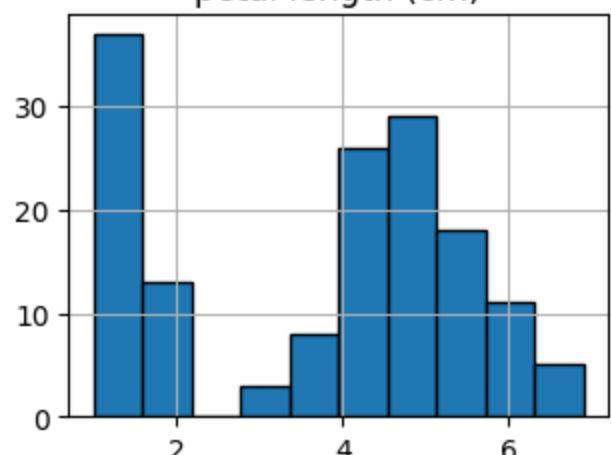
sepal length (cm)



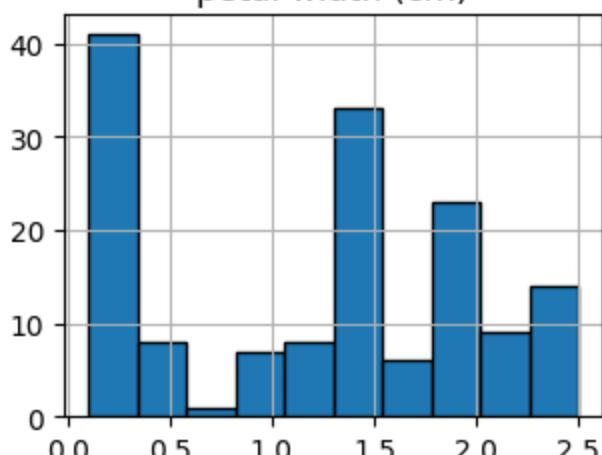
sepal width (cm)



petal length (cm)

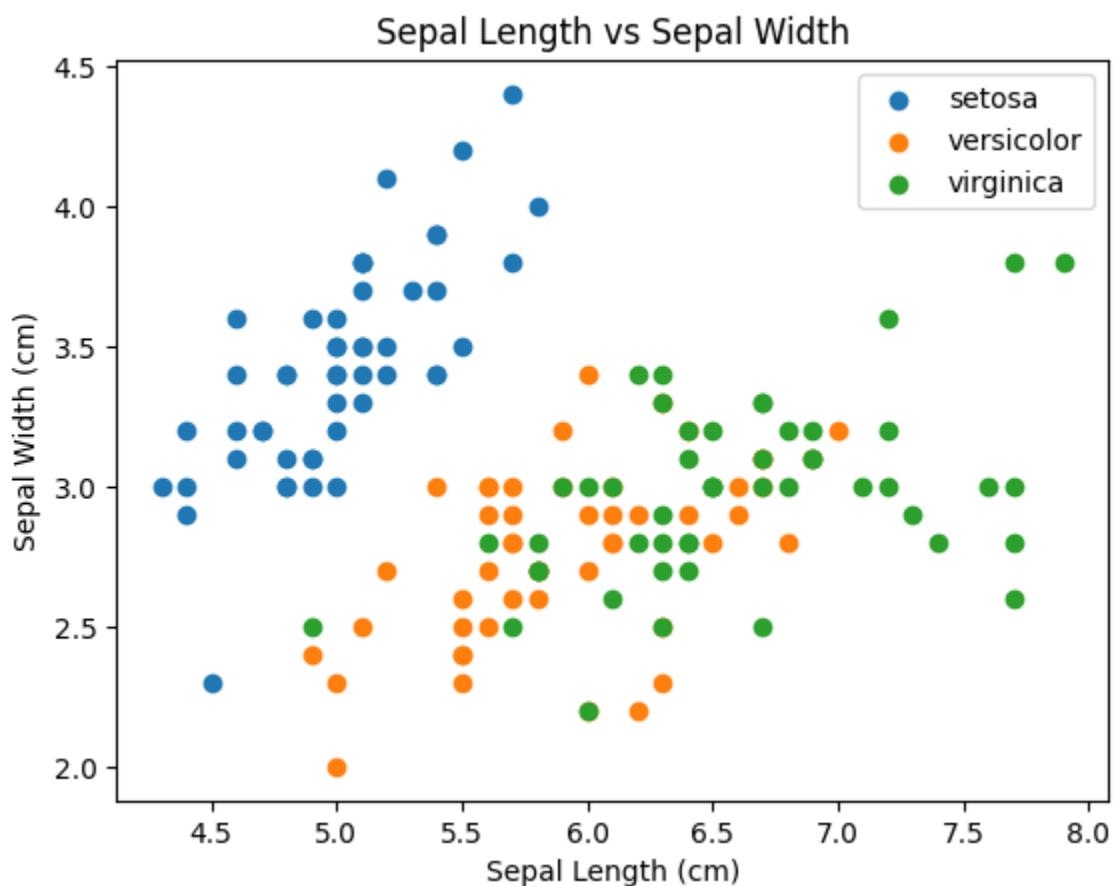


petal width (cm)



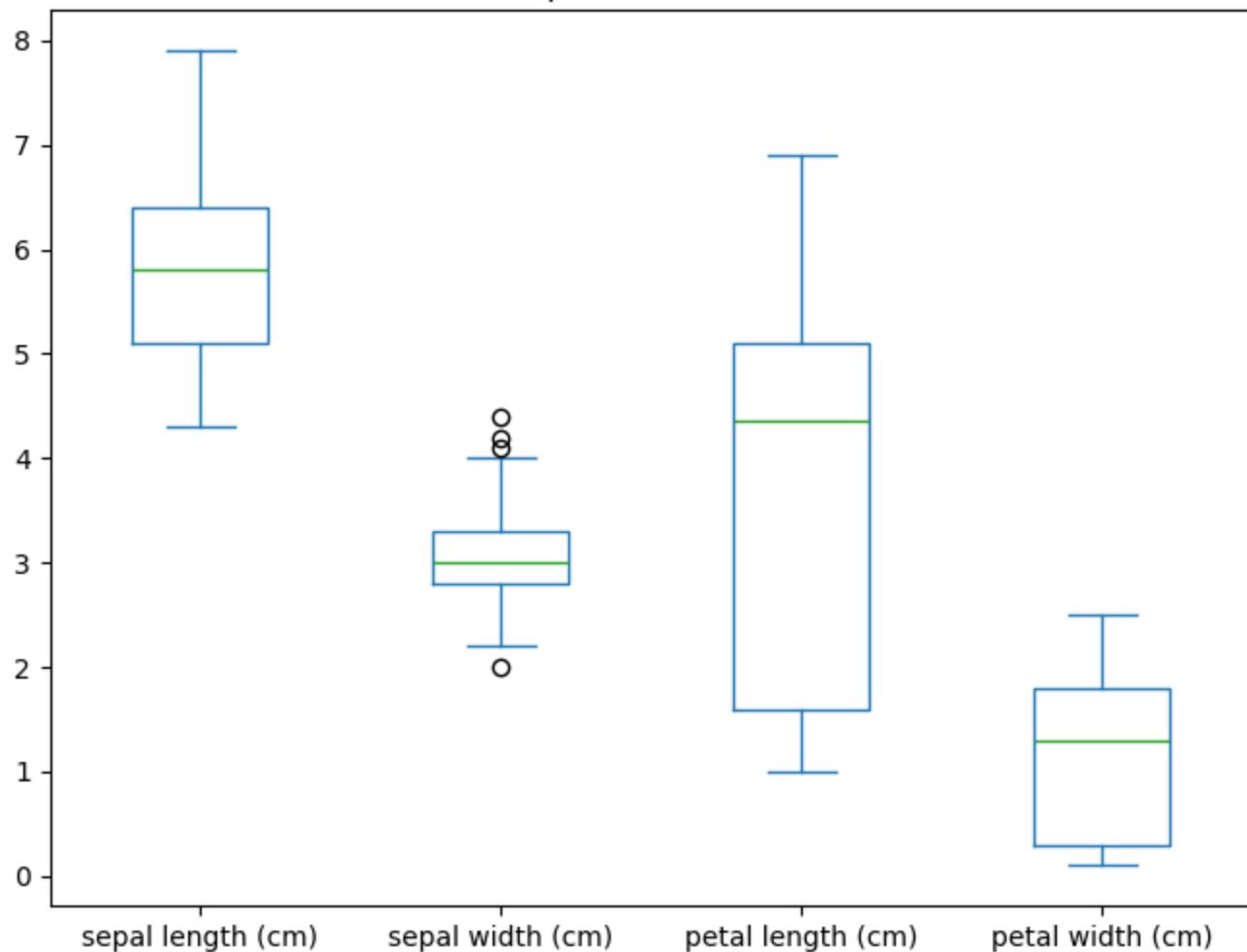
```
In [18]: # 2. Scatter Plot (Sepal Length vs Sepal Width)
for species in df['species'].unique():
    subset = df[df['species'] == species]
    plt.scatter(subset['sepal length (cm)'],
                subset['sepal width (cm)'],
                label=species)
plt.xlabel('Sepal Length (cm)')
plt.ylabel('Sepal Width (cm)')
```

```
plt.title('Sepal Length vs Sepal Width')
plt.legend()
plt.show()
```



```
In [19]: # 3. Boxplot (compare distributions per feature)
df.drop(columns='species').plot(kind='box', figsize=(8, 6))
plt.title("Boxplots of Features")
plt.show()
```

Boxplots of Features



In []:

```
# 6. Correlation heatmap (numeric columns only)
# Plot heatmap
# Compute correlation matrix (numeric columns only)
corr_matrix = df.drop(columns='species').corr()

plt.figure(figsize=(8, 6))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f")
```

```
plt.title("Feature Correlation Heatmap")
plt.show()
```

