IRIS DATA ANALYSIS

Section1: Exploratory Data Analysis (EDA) with Python:

In [2]:

```
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:

```
df=pd.read_csv("IRIS.csv")
```

In [4]:

```
df
```

Out[4]:

	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
...	...	...	...	...	...
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

In [5]:

```
df.head(5)
```

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 [7]:

```
#to display stats about data
df.describe()
```

Out[7]:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [8]:

```
df.isnull().sum()
```

Out[8]:

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

In [10]:

```
#to display basic info of dataset
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 [11]:

```
#to display no. of samples on each class
df['species'].value_counts()
```

Out[11]:

```
species
Iris-setosa        50
Iris-versicolor    50
```

```
Iris-virginica     50
Name: count, dtype: int64
```

In [12]:

```
#check for null values
df.isnull().sum()
```

Out[12]:

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

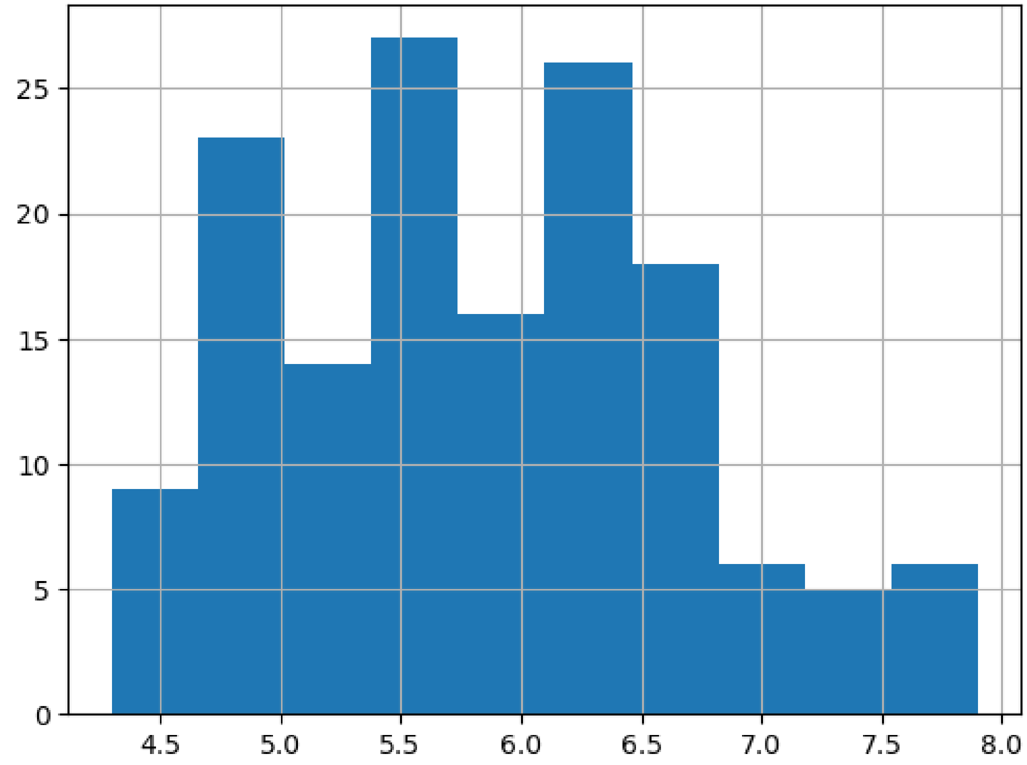
Exploratory Data Analysis

In [13]:

```
df['sepal_length'].hist()
```

Out[13]:

```
<Axes: >
```

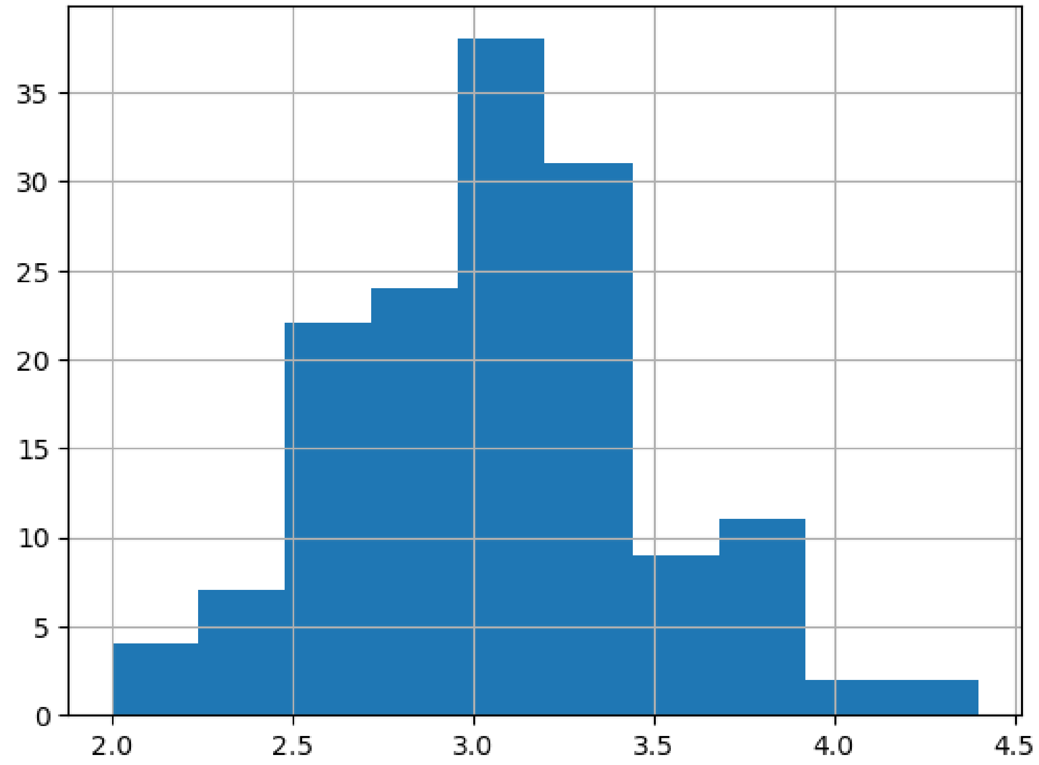


In [14]:

```
df['sepal_width'].hist()
```

Out[14]:

```
<Axes: >
```

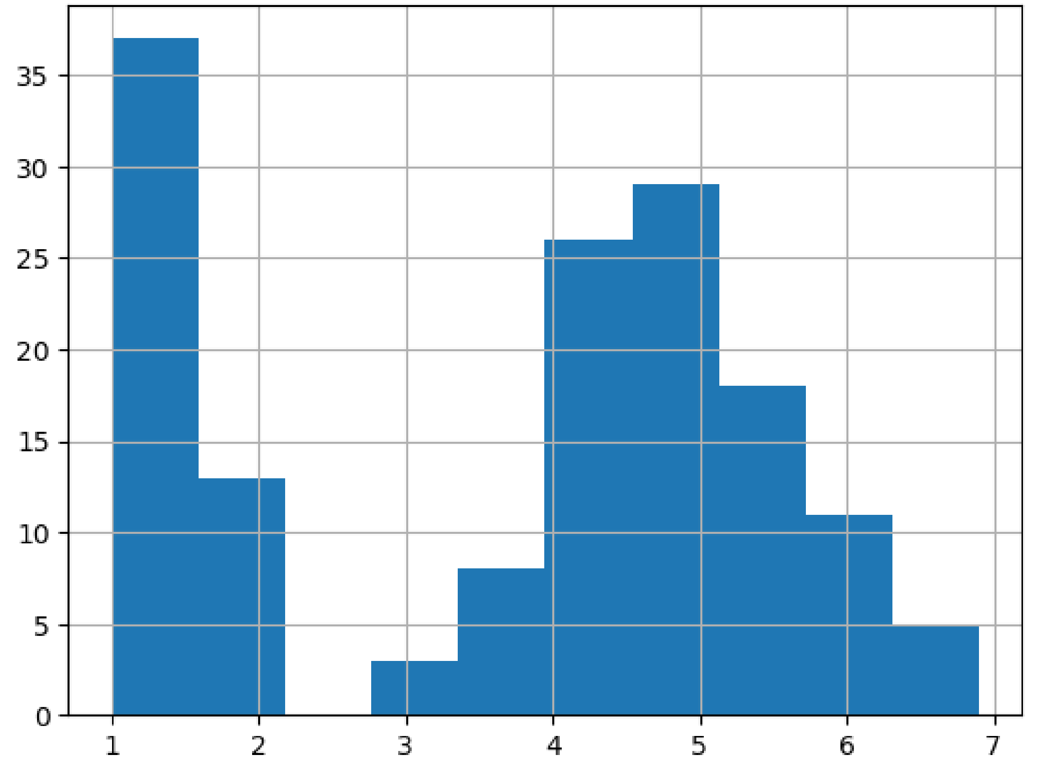


In [15]:

```
df['petal_length'].hist()
```

Out[15]:

```
<Axes: >
```

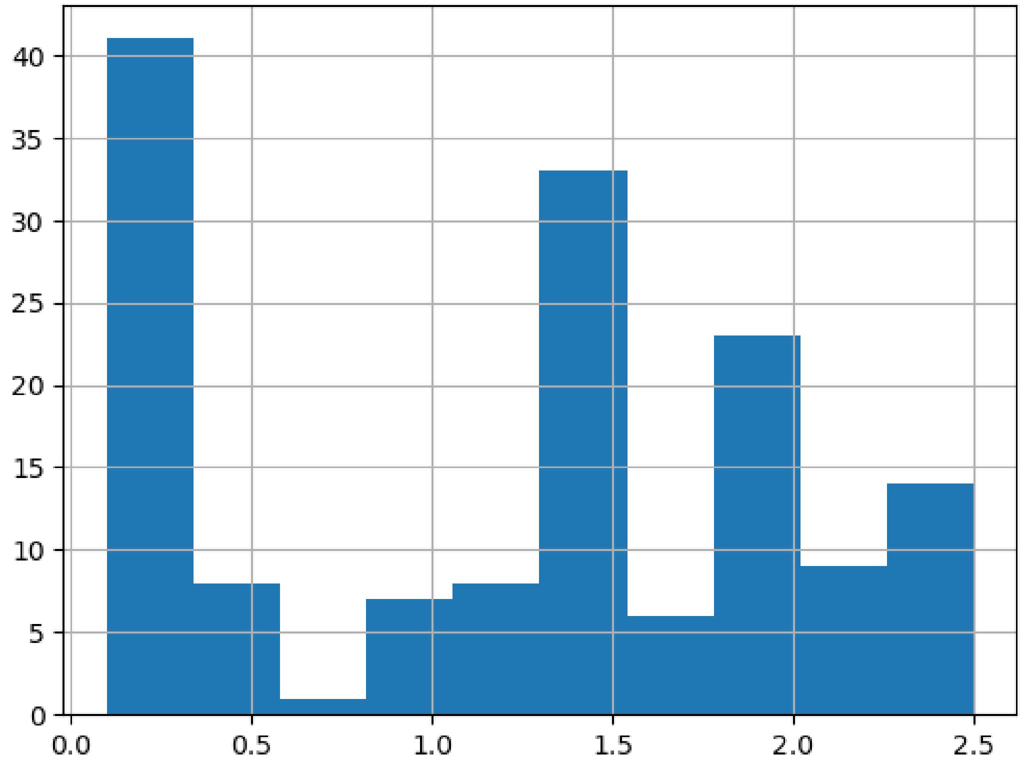


In [16]:

```
df['petal_width'].hist()
```

Out[16]:

```
<Axes: >
```

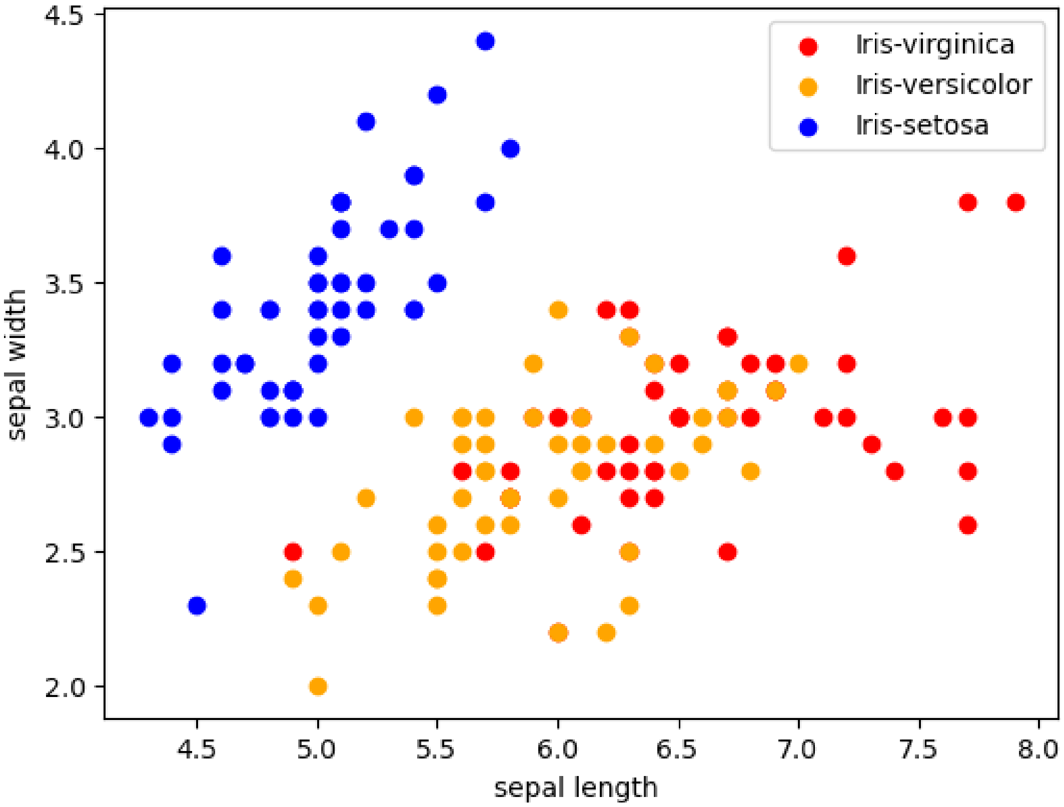


In [17]:

```
#scatterplot
colors=['red','orange','blue']
species=['Iris-virginica','Iris-versicolor','Iris-setosa']
```

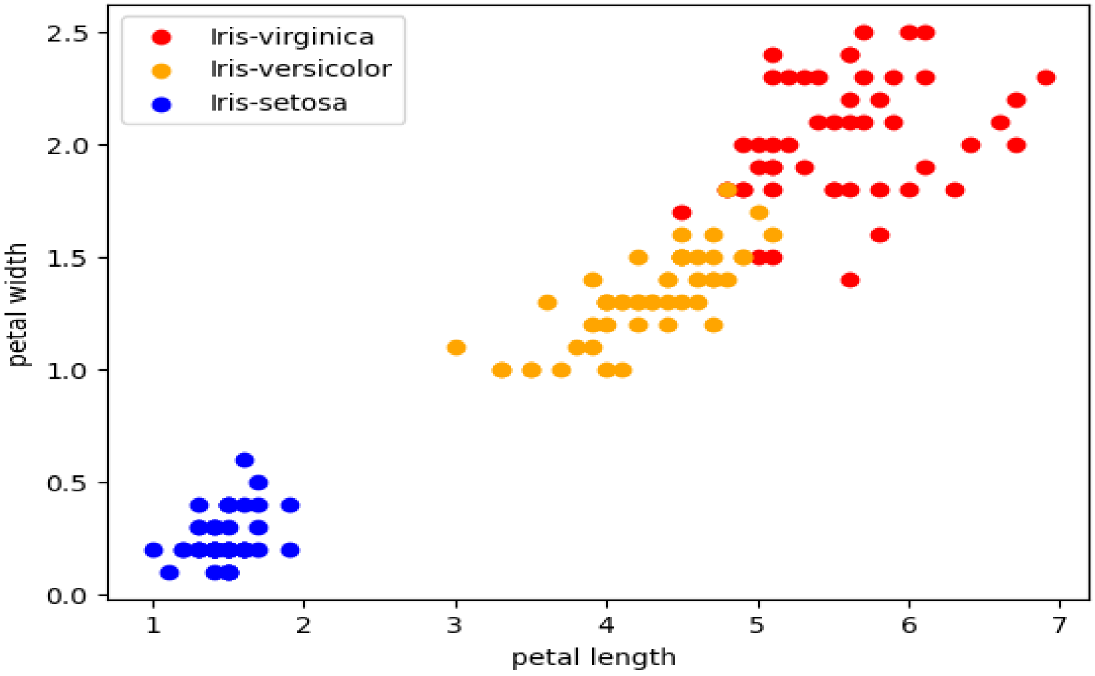
In [19]:

```
for i in range(3):
    x=df[df['species']==species[i]]
    plt.scatter(x['sepal_length'],x['sepal_width'],c=colors[i],label=species[i])
    plt.xlabel('sepal length ')
    plt.ylabel('sepal width')
    plt.legend()
```



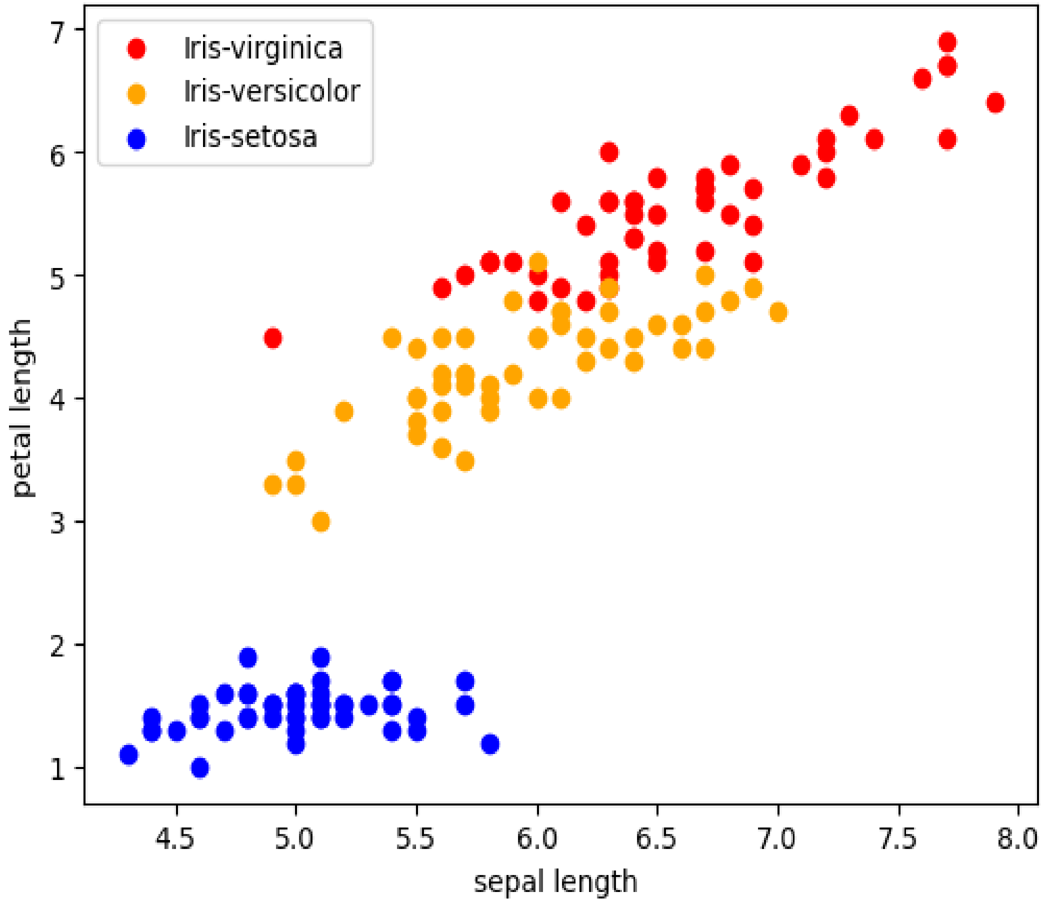
In [20]:

```
for i in range(3):
    x=df[df['species']==species[i]]
    plt.scatter(x['petal_length'],x['petal_width'],c=colors[i],label=species[i])
    plt.xlabel('petal length ')
    plt.ylabel('petal width')
    plt.legend()
```



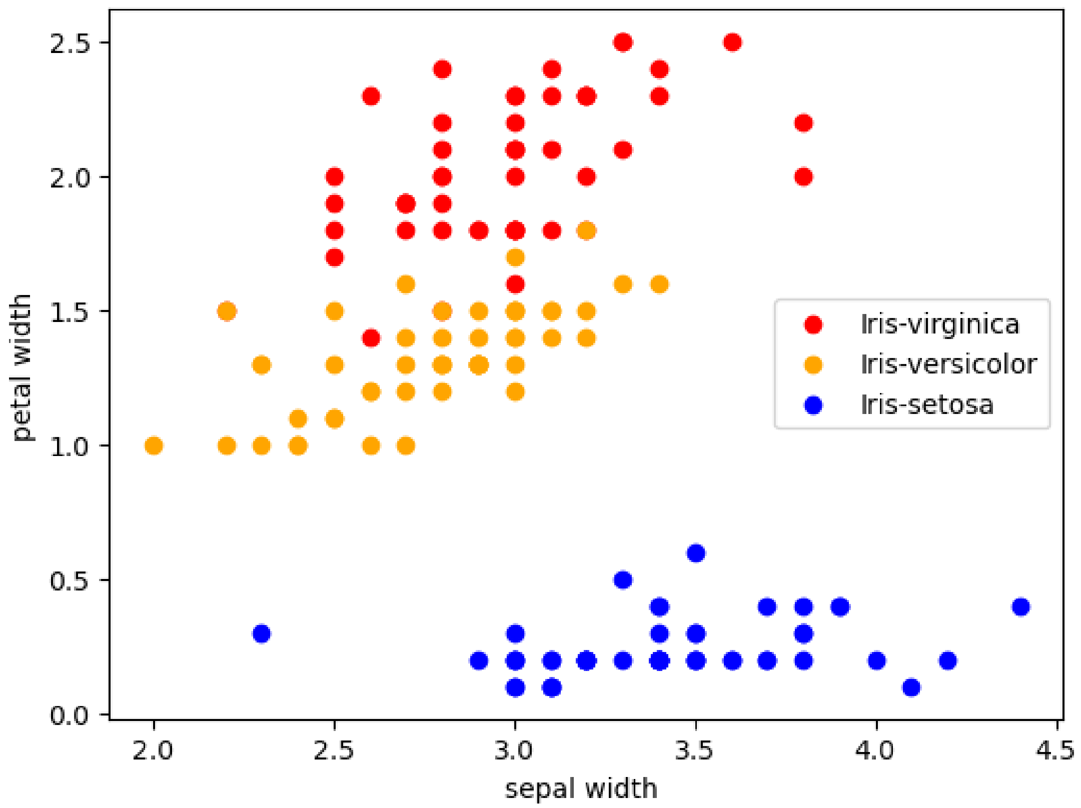
In [21]:

```
for i in range(3):
    x=df[df['species']==species[i]]
    plt.scatter(x['sepal_length'],x['petal_length'],c=colors[i],label=species[i])
    plt.xlabel('sepal length ')
    plt.ylabel('petal length')
    plt.legend()
```



In [22]:

```
for i in range(3):
    x=df[df['species']==species[i]]
    plt.scatter(x['sepal_width'],x['petal_width'],c=colors[i],label=species[i])
    plt.xlabel('sepal width ')
    plt.ylabel('petal width')
    plt.legend()
```



Correlation Matrix

In [32]:

```
df['species'] = pd.to_numeric(df['species'], errors='coerce')
```

In [33]:

```
df.dtypes
```

Out[33]:

```
sepal_length    float64
sepal_width     float64
petal_length    float64
petal_width     float64
species         float64
dtype: object
```

In [34]:

```
df.corr()
```

Out[34]:

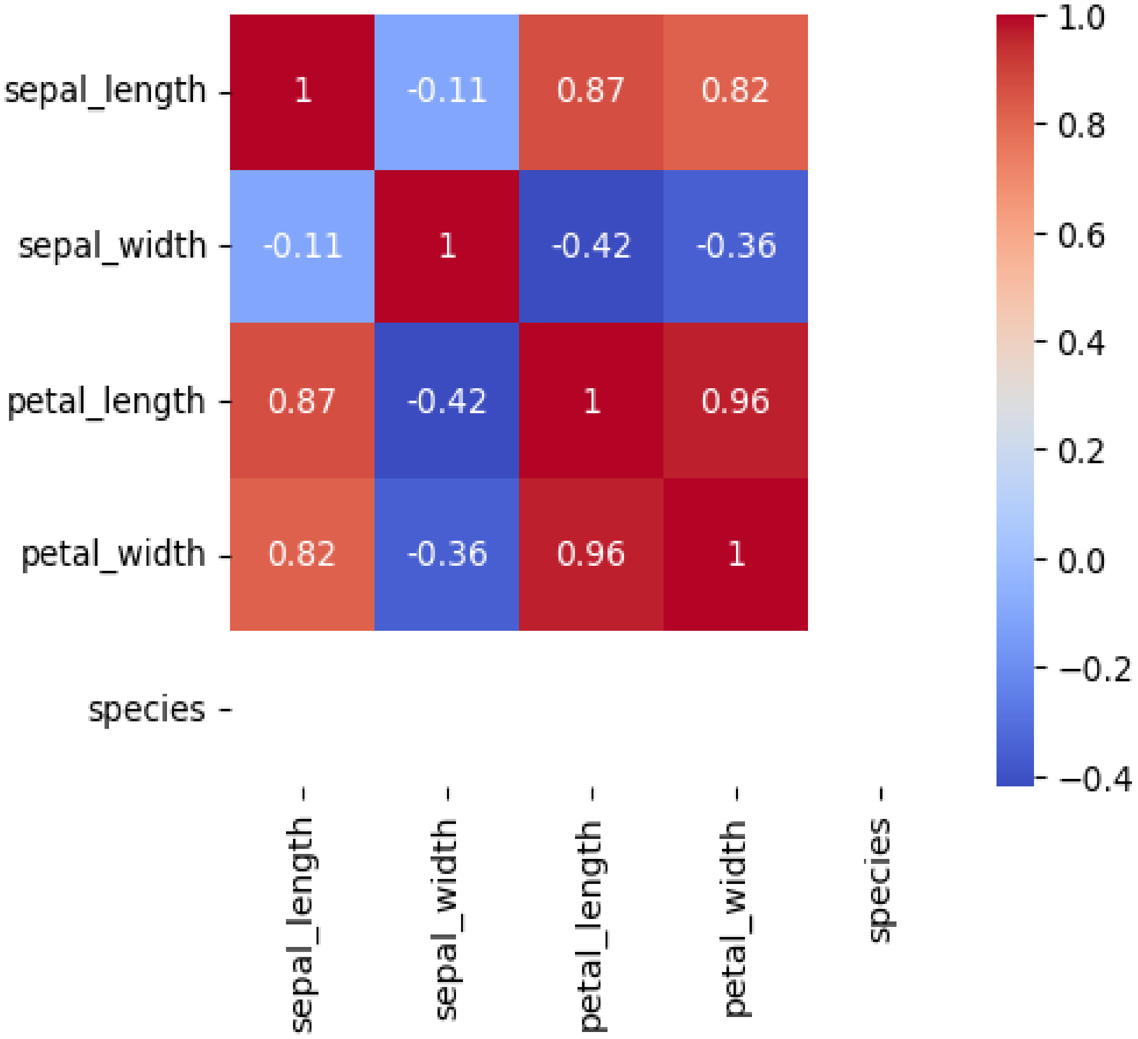
	sepal_length	sepal_width	petal_length	petal_width	species
sepal_length	1.000000	-0.109369	0.871754	0.817954	NaN
sepal_width	-0.109369	1.000000	-0.420516	-0.356544	NaN
petal_length	0.871754	-0.420516	1.000000	0.962757	NaN
petal_width	0.817954	-0.356544	0.962757	1.000000	NaN
species	NaN	NaN	NaN	NaN	NaN

In [36]:

```
corr=df.corr()
fig,ax=plt.subplots(figsize=(5,4))
sns.heatmap(corr,annot=True,ax=ax, cmap='coolwarm')
```

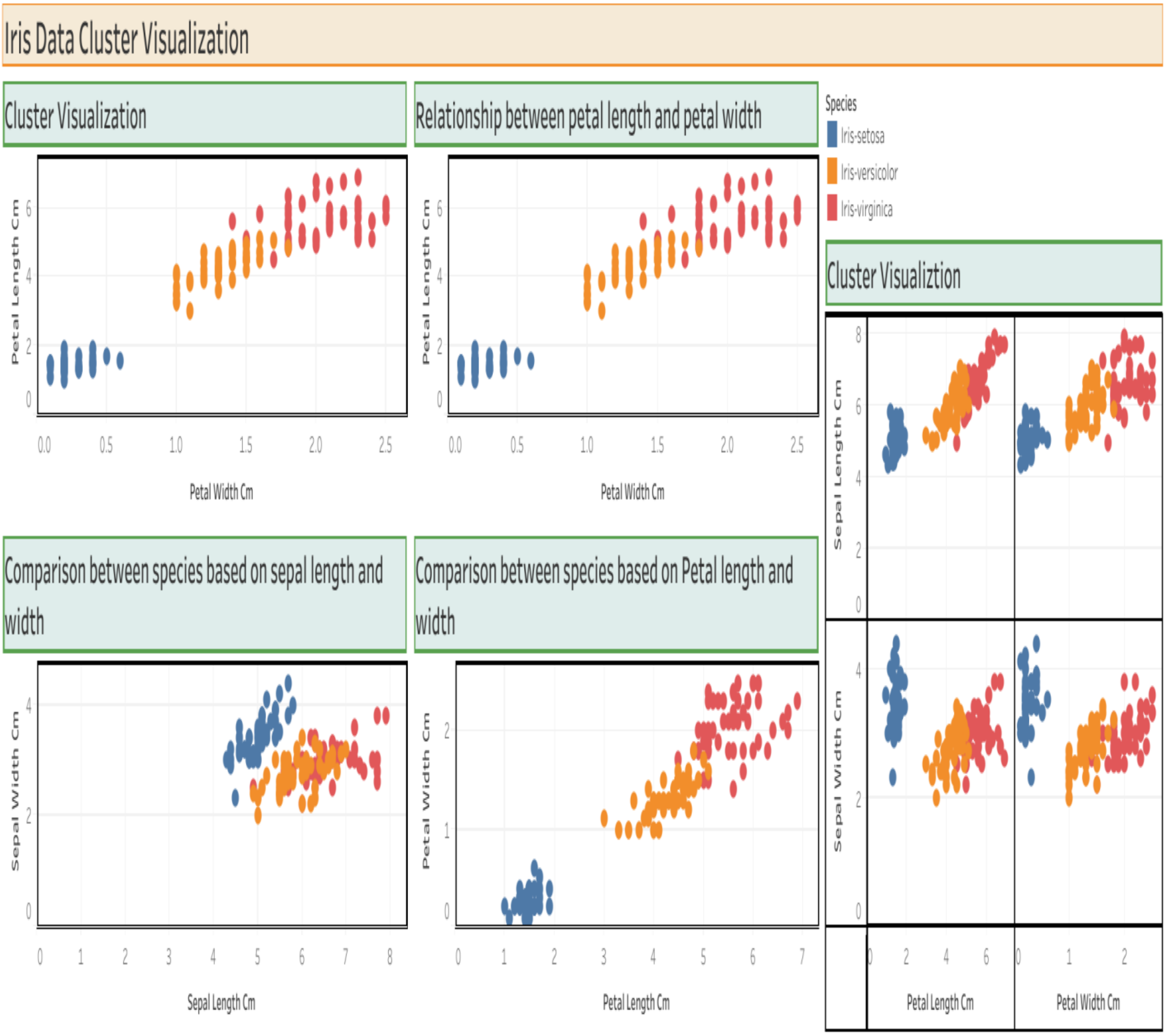
Out[36]:

```
<Axes: >
```



Section 2: Data Visualization with Power BI or Tableau

1.Iris Data Cluster Visualization



Iris Data Cluster Visualization

Cluster Visualization

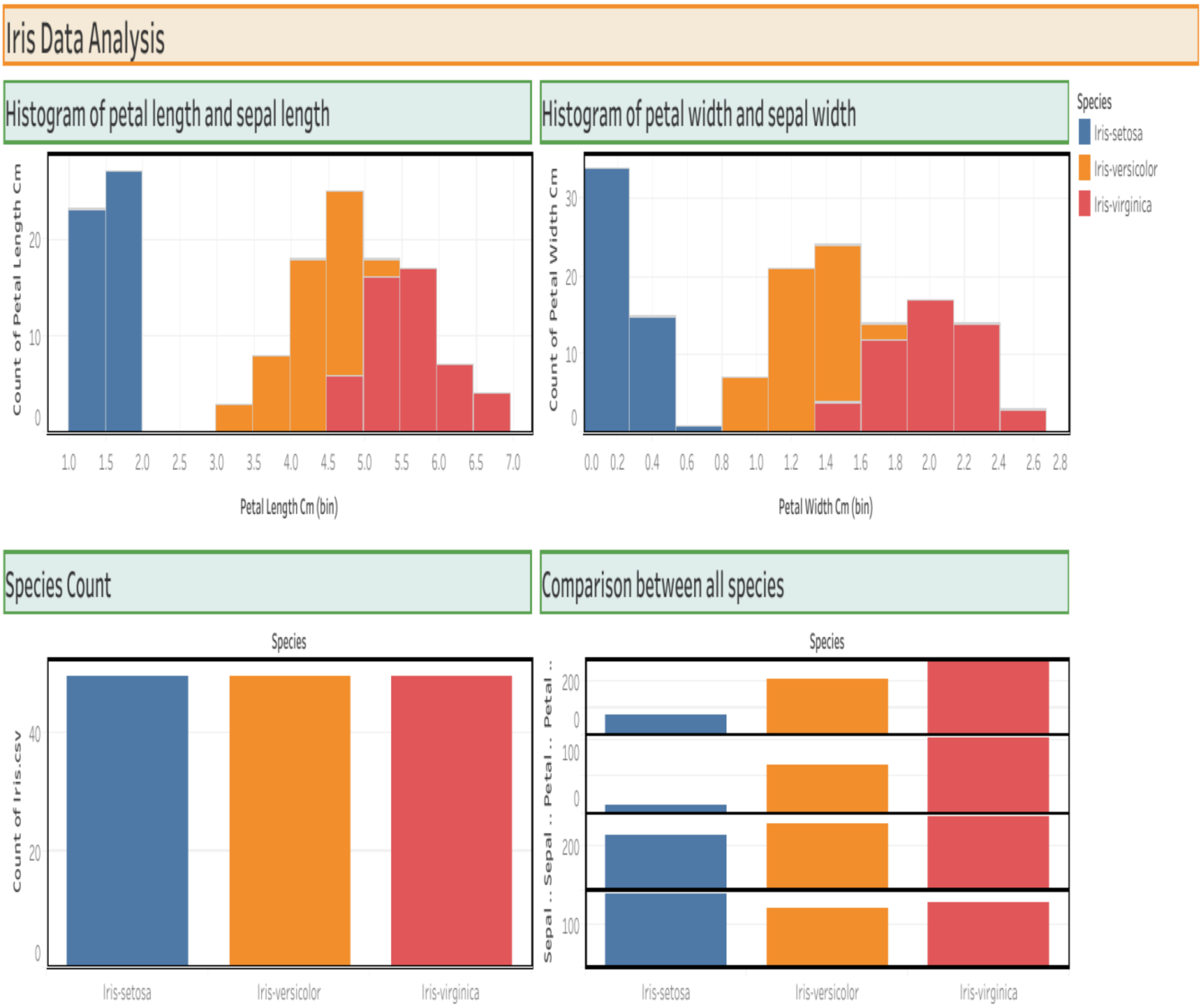
Relationship between petal length and petal width

Comparison between species based on sepal length and width

Comparison between species based on Petal length and width

Cluster Visualiztion

2. Iris Data Analysis



Conclusion:

In conclusion, the Iris data analysis has provided valuable insights into the intricate relationships within the dataset. Through meticulous exploration, visualization, and modeling, distinct patterns have emerged, showcasing the clear separation of the three iris species based on their sepal and petal characteristics. Feature importance analysis has highlighted key attributes influencing the classification, aiding in a nuanced understanding of the dataset. The chosen machine learning model, following careful evaluation and hyperparameter tuning, demonstrates robust performance in accurately classifying iris flowers. Moreover, considerations of feature correlations, outlier detection, and generalization have enhanced the reliability and generalizability of the model. This comprehensive analysis not only reaffirms the suitability of the Iris dataset for introductory purposes but also underscores the significance of employing a systematic approach in unraveling patterns and deriving meaningful insights from complex datasets.