In [2]:

```
# IRIS FLOWER CLASSIFICATION TASK(3)
```

In [3]:

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

In [4]:

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

In [5]:

```
df
```

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
...	...	...	...	...	...
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 [6]:

```
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 [7]:

```
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 [9]:

```
df.shape
```

Out[9]:

```
(150, 5)
```

In [10]:

```
df.transpose()
print(df)
```

```
     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 x 5 columns]
```

In [11]:

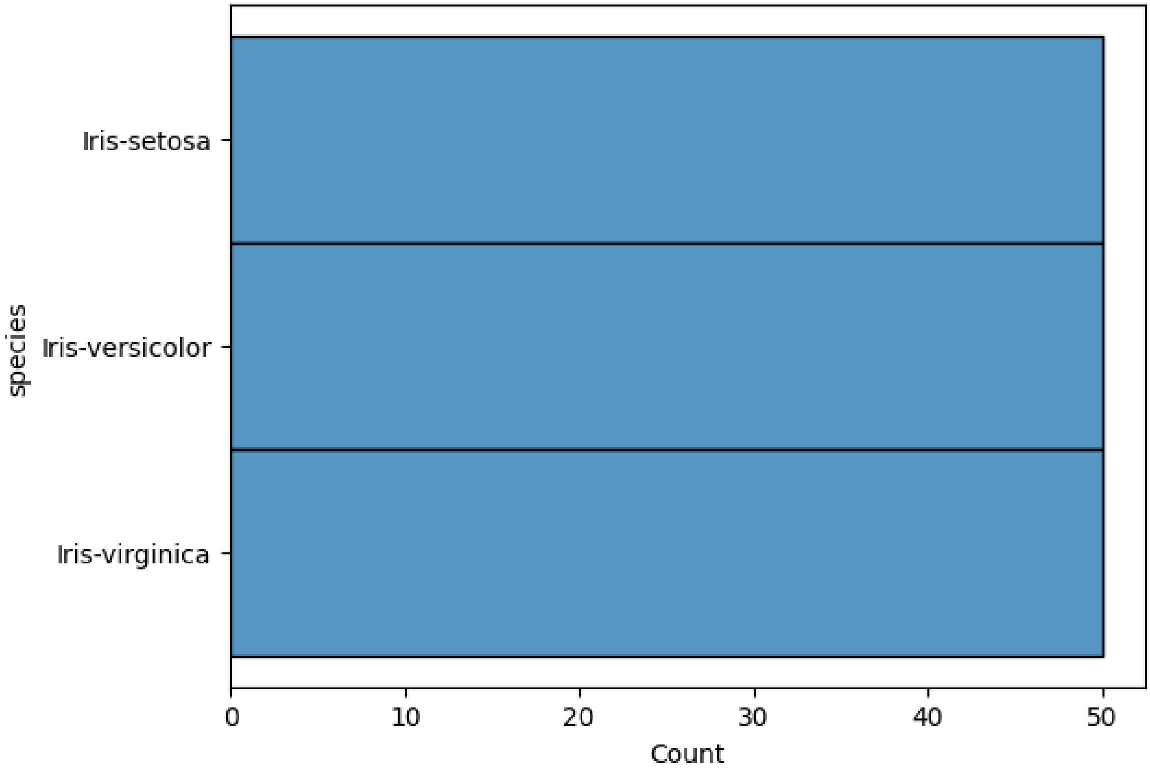
```
df["species"].value_counts()
```

Out[11]:

```
species
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: count, dtype: int64
```

In [12]:

```
sns.histplot(y="species",data=df)
plt.show()
```

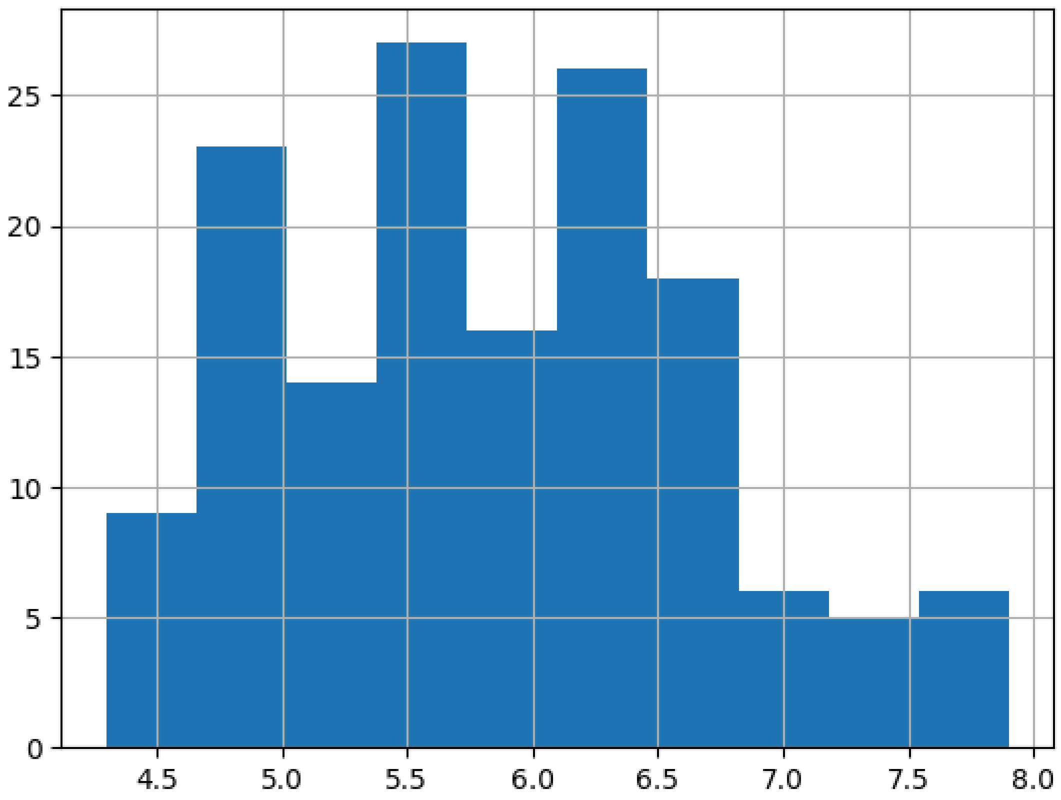


```
In [19]: df.isna().sum()
```

```
Out[19]: sepal_length    0
         sepal_width     0
         petal_length    0
         petal_width     0
         species         0
         dtype: int64
```

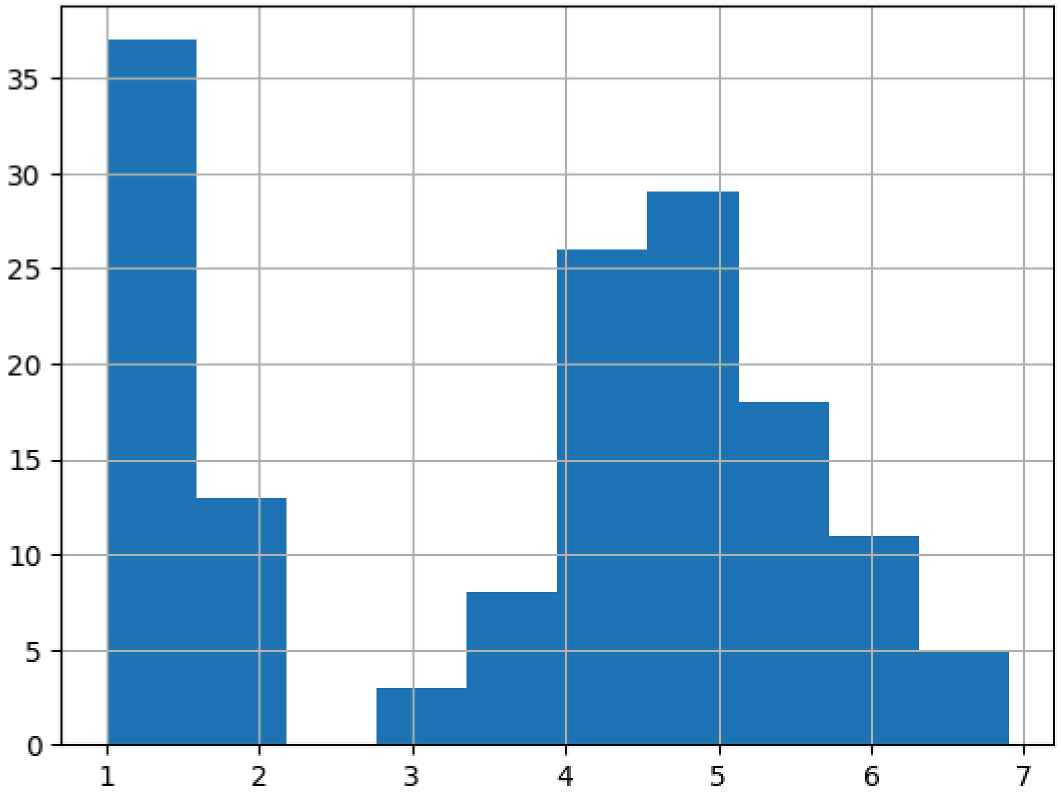
```
In [21]: df['sepal_length'].hist()
```

```
Out[21]: <Axes: >
```



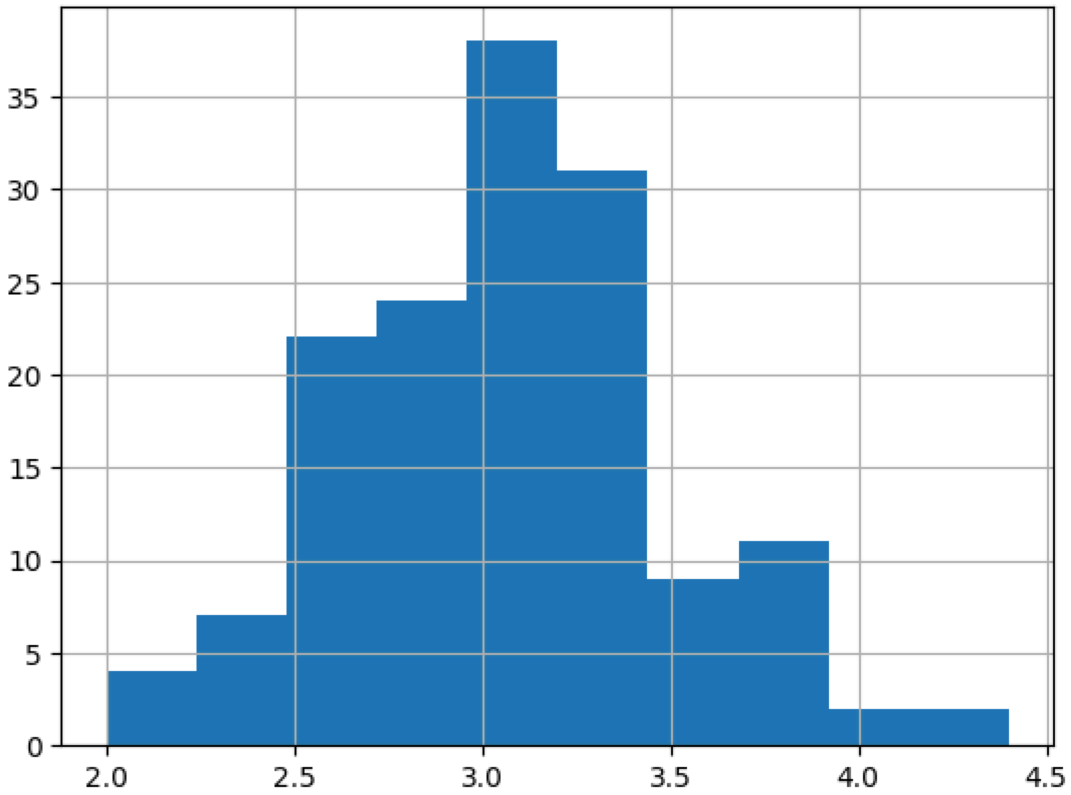
```
In [22]: df['petal_length'].hist()
```

```
Out[22]: <Axes: >
```



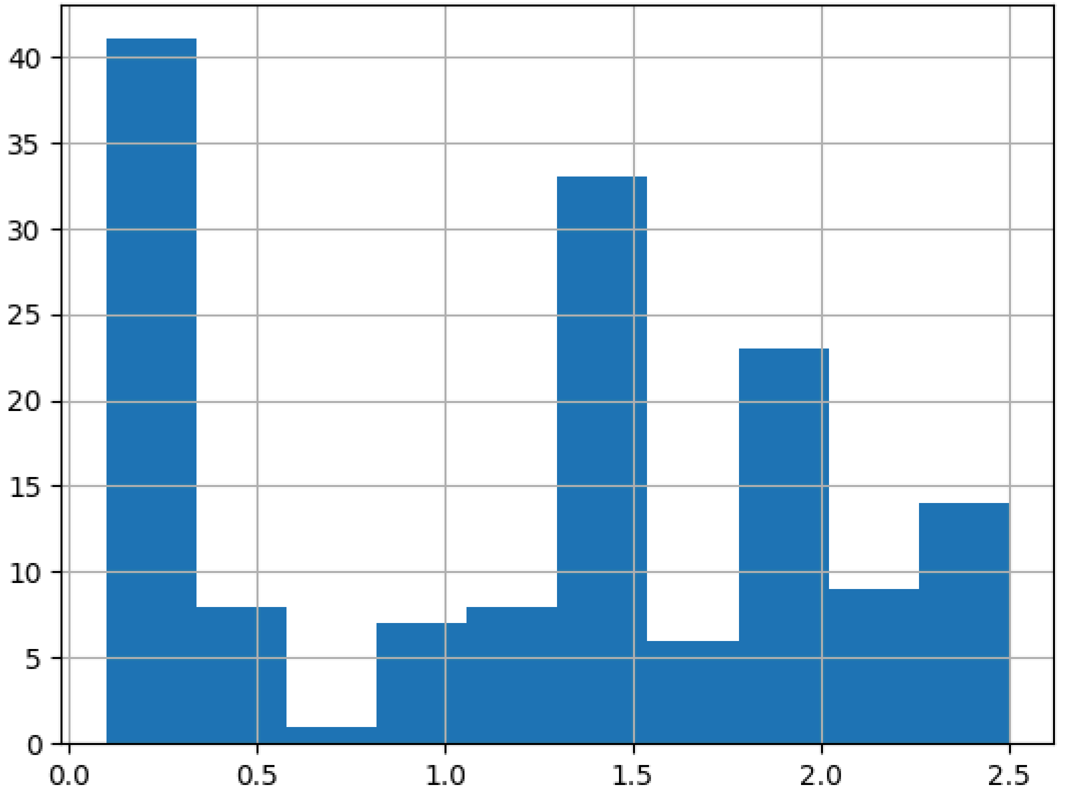
```
In [23]: df['sepal_width'].hist()
```

```
Out[23]: <Axes: >
```



```
In [24]: df['petal_width'].hist()
```

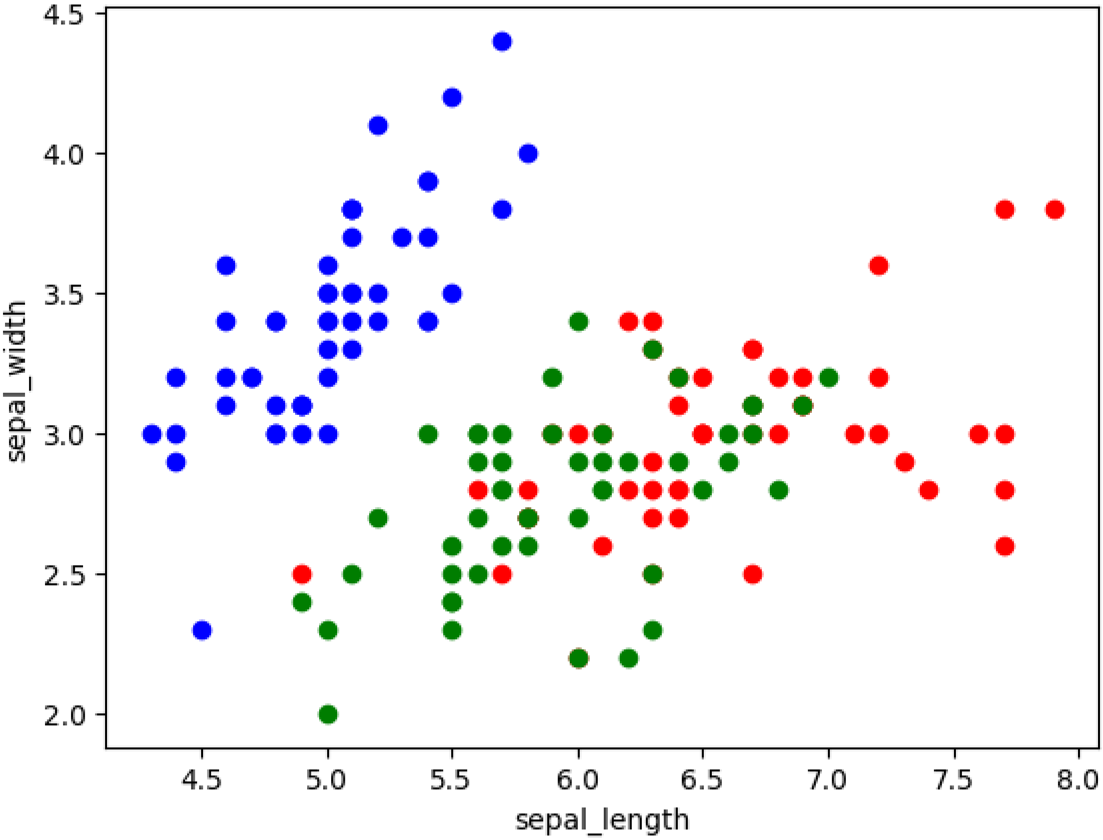
```
Out[24]: <Axes: >
```



```
In [28]: colors=['red','Green','blue']
         species =['Iris-virginica','Iris-versicolor','Iris-setosa']
```

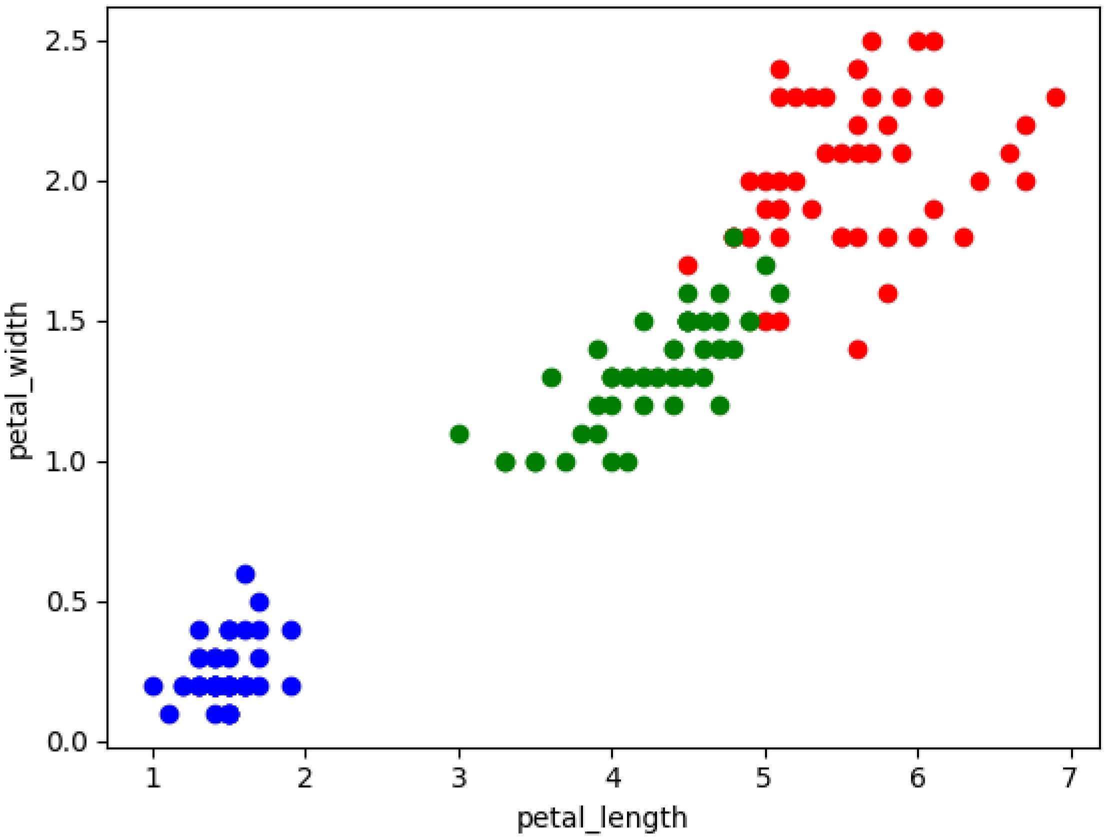
```
In [29]: for i in range(3):
             x = df[df['species'] == species[i]]
```

```
    plt.scatter(x['sepal_length'],x['sepal_width'],c= colors[i], label = species
plt.xlabel("sepal_length")
plt.ylabel("sepal_width")
plt.show()
```



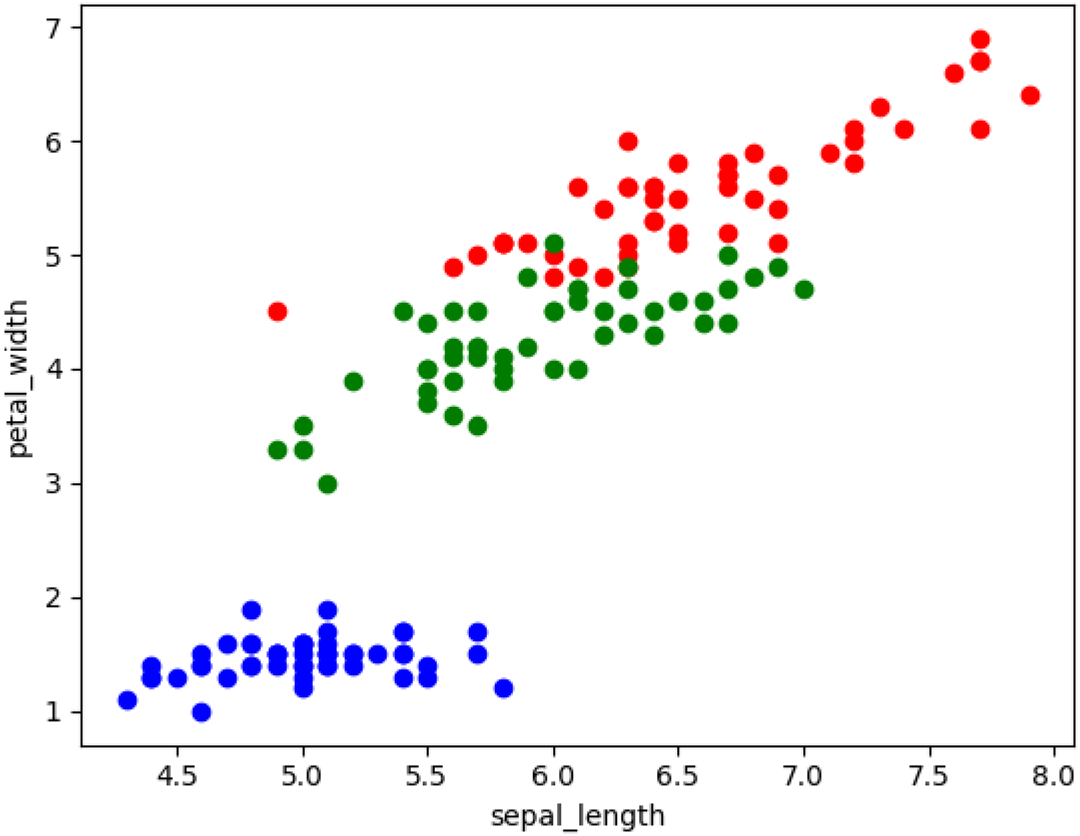
In [30]:

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



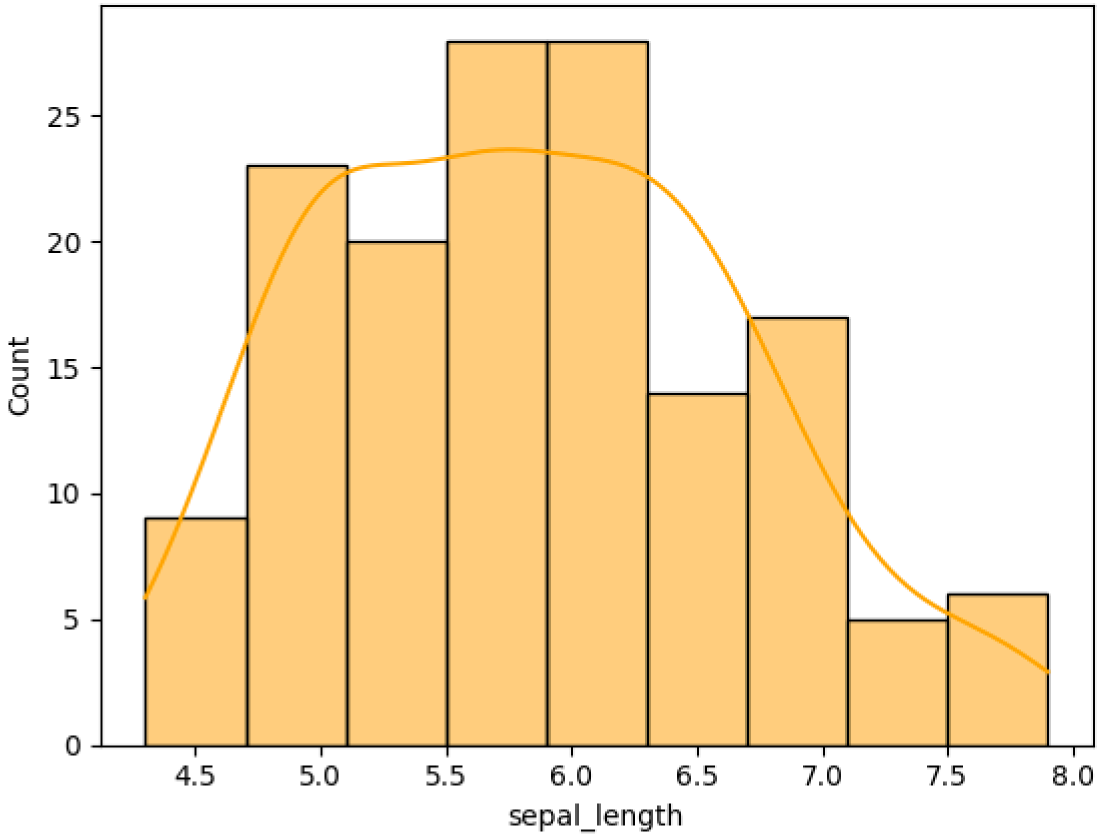
In [31]:

```
for i in range(3):
    x = df[df['species'] == species[i]]
    plt.scatter(x['sepal_length'],x['petal_length'],c= colors[i], label = specie
plt.xlabel("sepal_length")
plt.ylabel("petal_width")
plt.show()
```



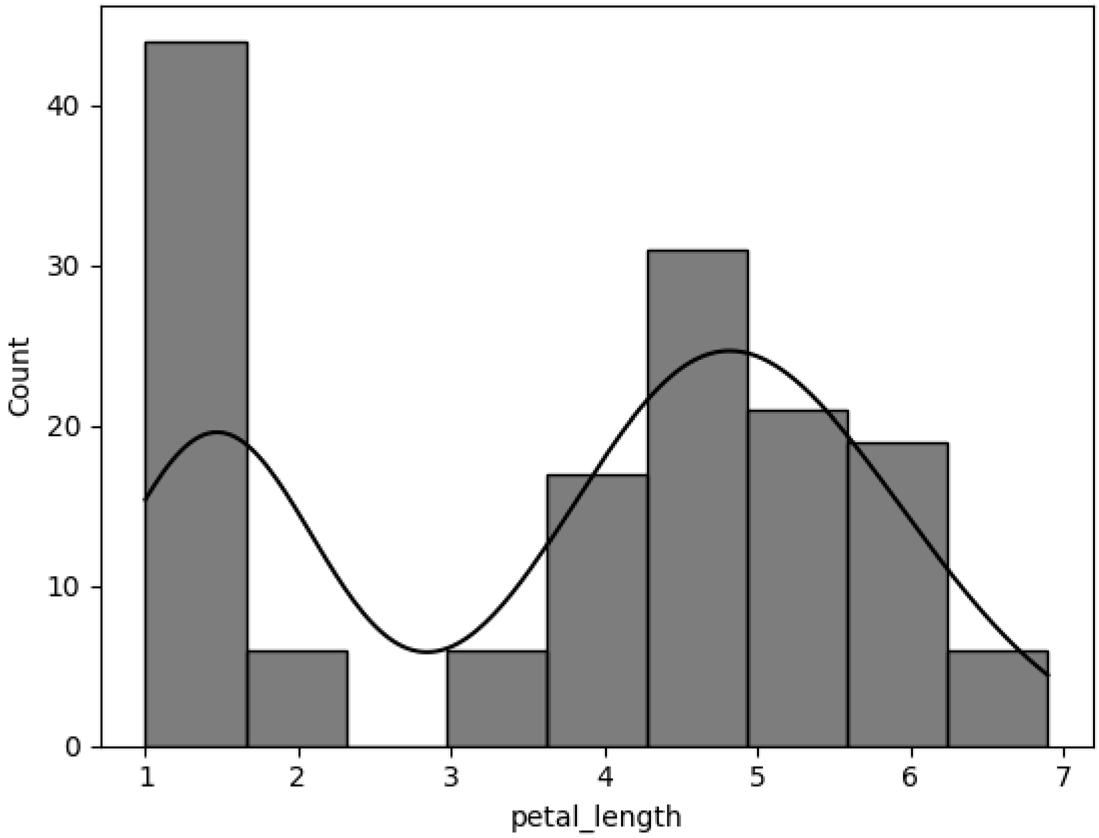
In [32]:

```
sns.histplot(data=df, x="sepal_length",color="orange",kde=True)
plt.show()
```



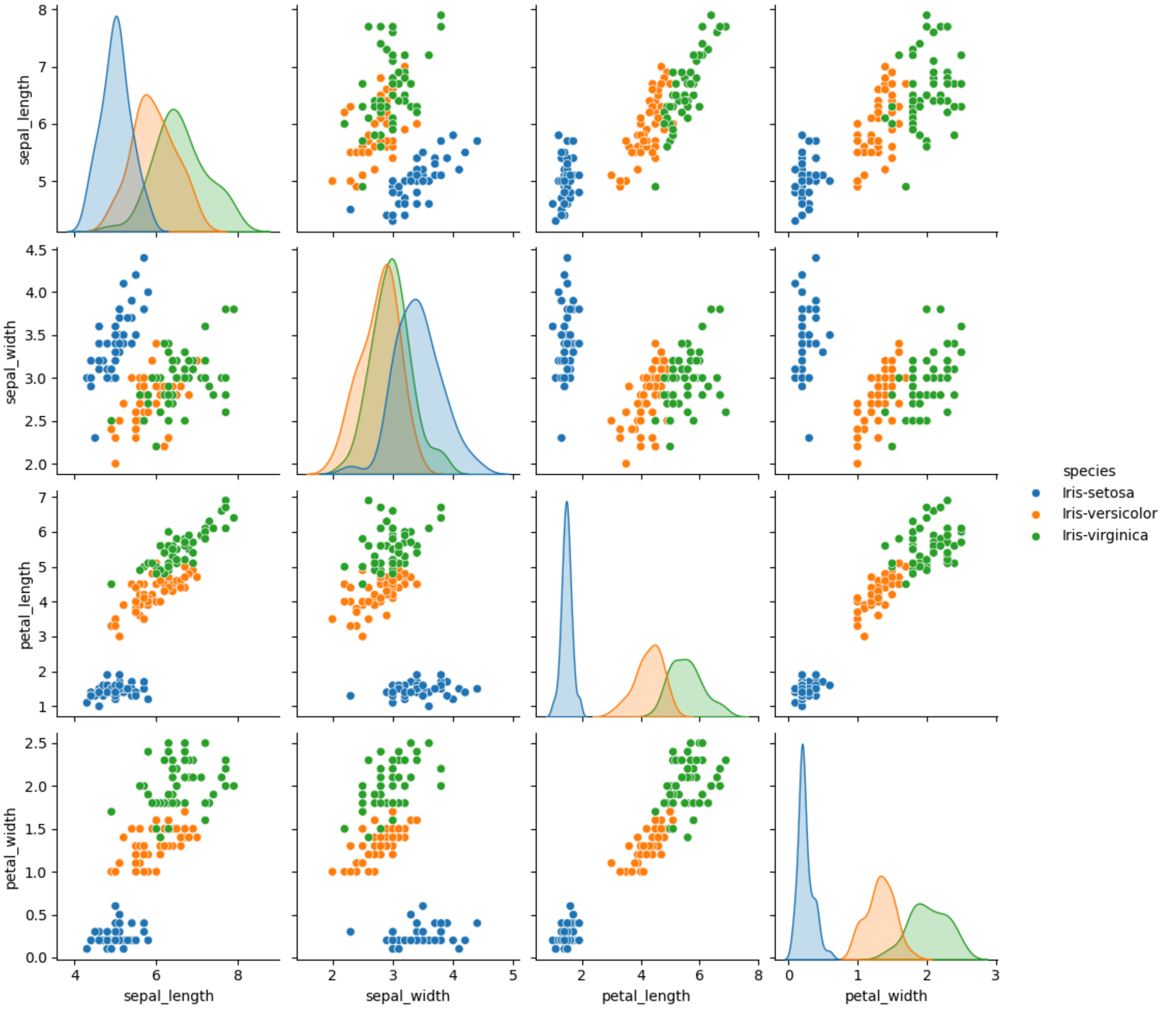
In [33]:

```
sns.histplot(data=df, x="petal_length",color="black",kde=True)
plt.show()
```



In [38]:

```
sns.pairplot(df, hue="species")
plt.show()
```



In [39]:

```
X=df.drop(columns="species",axis=1)
y=df["species"]
```

In [40]:

```
X.shape
```

Out[40]:

```
(150, 4)
```

In [49]:

```
from sklearn.preprocessing import LabelEncoder
import pandas as pd
le = LabelEncoder()
label_encoder = LabelEncoder()
```

In [50]:

```
df['species']= le.fit_transform(df['species'])
df.head()
```

Out[50]:

	species
0	0
1	1
2	2
3	0
4	2

In [52]:

```
df['species_encoded'] = label_encoder.fit_transform(df['species'])
print(df)
```

```
      species  species_encoded
0      setosa                0
1  versicolor                1
2   virginica                2
3      setosa                0
4   virginica                2
```

In [53]:

```
df.head()
```

Out[53]:

	species	species_encoded
0	setosa	0
1	versicolor	1
2	virginica	2
3	setosa	0
4	virginica	2

In [61]:

```
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import LogisticRegression
```

In [62]:

```
X= df.drop(columns=['species'])
Y= df['species']
x_train,x_test ,y_train , y_test = train_test_split(X,Y,test_size=0.30)
```

In [64]:

```
model=LogisticRegression()
model.fit(x_train,y_train)
```

Out[64]:

```
▾ LogisticRegression
LogisticRegression()
```

In [65]:

```
print("Accuracy: ",model.score(x_test,y_test)*100)
```

```
Accuracy:  100.0
```

In [66]:

```
model.fit(x_train,y_train)
```

```
Out[66]: ▾ LogisticRegression  i ?

         LogisticRegression()
```