In [2]:
```python
# Importing the Dependencies
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
#Data Collection
data=pd.read_csv("E:\Data Science\Irisdataset.csv")
data
```

Out[2]:

	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 [3]:
```python
# Data processing
data.head()
```

Out[3]:

	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 [5]:
```
data.describe()
```

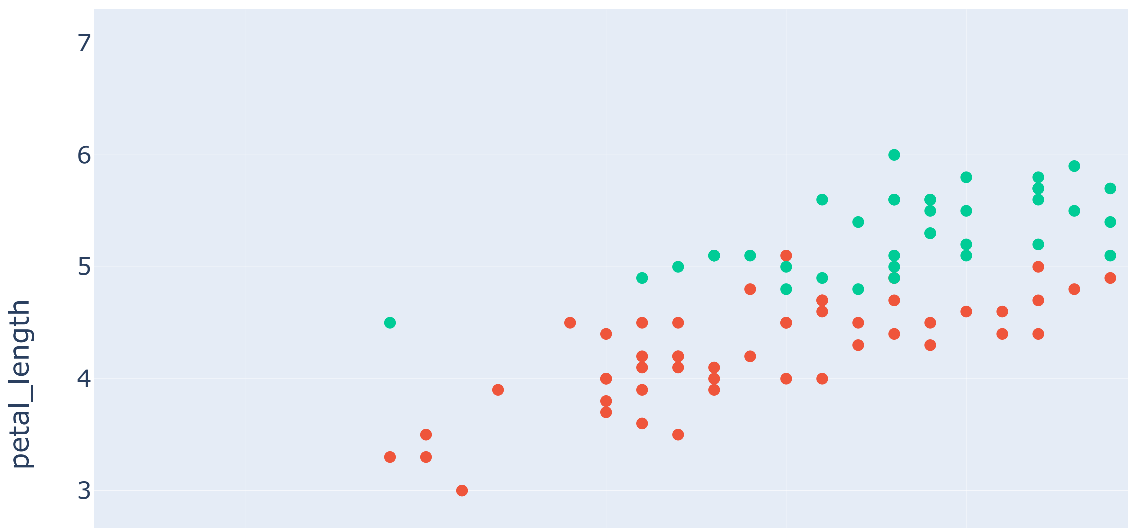
Out[5]:

	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 [ ]:
```
data.isnull().sum()
```

In [ ]:
```
import seaborn as sns
sns.countplot(data['species'])
```

In [16]:
```
# Distributions of Data
import plotly.express as px
fig=px.scatter(data,x="sepal_length",y="petal_length",color="species")
fig.show()
```



In [6]:

```python
# Training the data by using different models
from sklearn.model_selection import train_test_split
X=data.drop('species',axis=1)
Y=data['species']
print(X)
print(Y)
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=42
```

```
     sepal_length  sepal_width  petal_length  petal_width
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
..            ...          ...           ...          ...
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

[150 rows x 4 columns]
0         Iris-setosa
1         Iris-setosa
2         Iris-setosa
3         Iris-setosa
4         Iris-setosa
            ...
145    Iris-virginica
146    Iris-virginica
147    Iris-virginica
148    Iris-virginica
149    Iris-virginica
Name: species, Length: 150, dtype: object
```

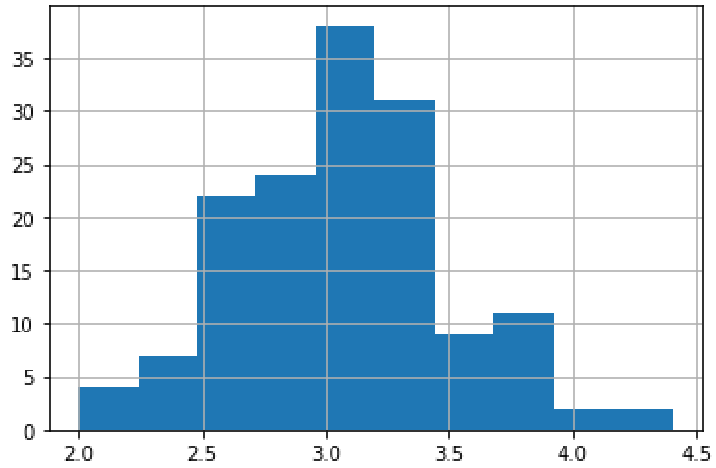
In [18]:

```python
data.tail()
```

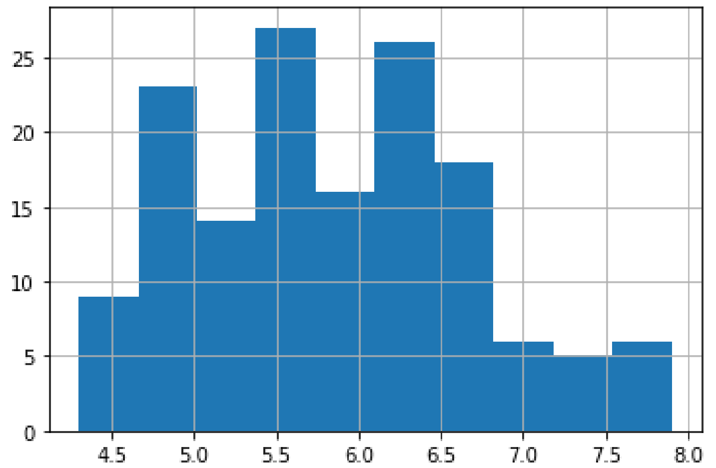
Out[18]:

	sepal_length	sepal_width	petal_length	petal_width	species
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

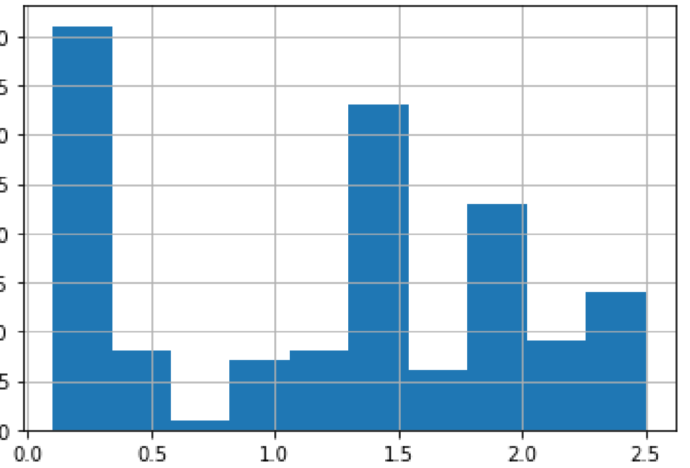
In [19]:
```
data['sepal_width'].hist()
```
Out[19]: <AxesSubplot:>



In [9]:
```
data['sepal_length'].hist()
```
Out[9]: <AxesSubplot:>

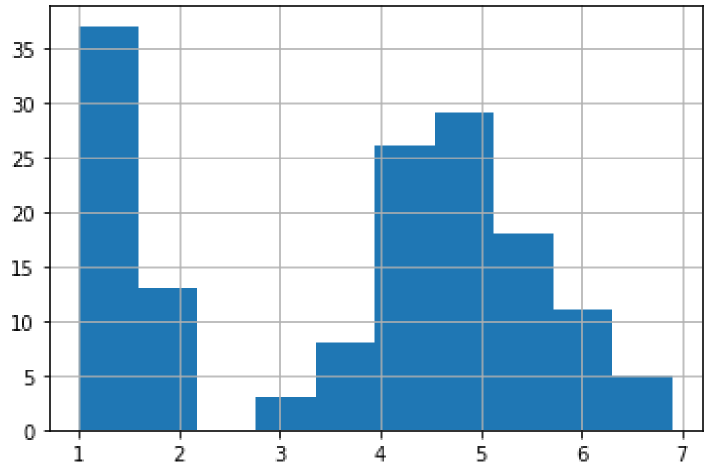


In [10]:
```
data['petal_width'].hist()
```
Out[10]: <AxesSubplot:>



In [11]:
```python
data['petal_length'].hist()
```

Out[11]: <AxesSubplot:>



In [12]:
```python
# Corelation Matrix
data.corr()
```

Out[12]:

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

In [13]:
```python
#DecisionTreeClassifier
from sklearn.model_selection import train_test_split
X=data.drop('species',axis=1)
Y=data['species']
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=42
```

In [14]:
```python
from sklearn.tree import DecisionTreeClassifier
f1=DecisionTreeClassifier()
f1.fit(X_train,Y_train)
l1=f1.score(X_test,Y_test)*100
print("Accuracy of Iris Flower Classifer using DecisionTreeClassifier is",l1)
```

```
Accuracy of Iris Flower Classifer using DecisionTreeClassifier is 100.0
```

In [15]:
```python
#KNeighborsClassifier
from sklearn.neighbors import KNeighborsClassifier
f11=KNeighborsClassifier()
f11.fit(X_train,Y_train)
l11=f11.score(X_test,Y_test)*100
print("Accuracy of Iris Flower Classifer  using KNeighborsClassifier  is",l11)
```

```
Accuracy of Iris Flower Classifer  using KNeighborsClassifier  is 100.0
```

In [21]:
```python
# Classification Algorithm
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=1)
knn.fit(X_train,Y_train)
x=np.array([[4.6,3.1,1.5,0.2]])
predict=knn.predict(x)
print("Prediction:{}".format(predict))
```

```
Prediction:['Iris-setosa']

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning:

X does not have valid feature names, but KNeighborsClassifier was fitted with f
eature names
```

In [ ]:
```python

```