In [14]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os
```

In [16]:
```
# Reading dataset of globalsuperstore
df = pd.read_csv('Iris.csv')
df.head()
```

Out[16]:

| | Id | SepalLengthCm | SepalWidthCm | PetalLengthCm | PetalWidthCm | Species |
|---|---|---|---|---|---|---|
| **0** | 1 | 5.1 | 3.5 | 1.4 | 0.2 | Iris-setosa |
| **1** | 2 | 4.9 | 3.0 | 1.4 | 0.2 | Iris-setosa |
| **2** | 3 | 4.7 | 3.2 | 1.3 | 0.2 | Iris-setosa |
| **3** | 4 | 4.6 | 3.1 | 1.5 | 0.2 | Iris-setosa |
| **4** | 5 | 5.0 | 3.6 | 1.4 | 0.2 | Iris-setosa |

In [17]:
```
# to delete the column ie Id
df = df.drop(columns=['Id'])
df.head()
```

Out[17]:

| | SepalLengthCm | SepalWidthCm | PetalLengthCm | PetalWidthCm | 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 [18]:
```
# to display statistics about data
df.describe()
```

Out[18]:

| | SepalLengthCm | SepalWidthCm | PetalLengthCm | PetalWidthCm |
|---|---|---|---|---|
| **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 [19]:
```
# to display basic info about datatype
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   SepalLengthCm  150 non-null    float64
 1   SepalWidthCm   150 non-null    float64
 2   PetalLengthCm  150 non-null    float64
 3   PetalWidthCm   150 non-null    float64
 4   Species        150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

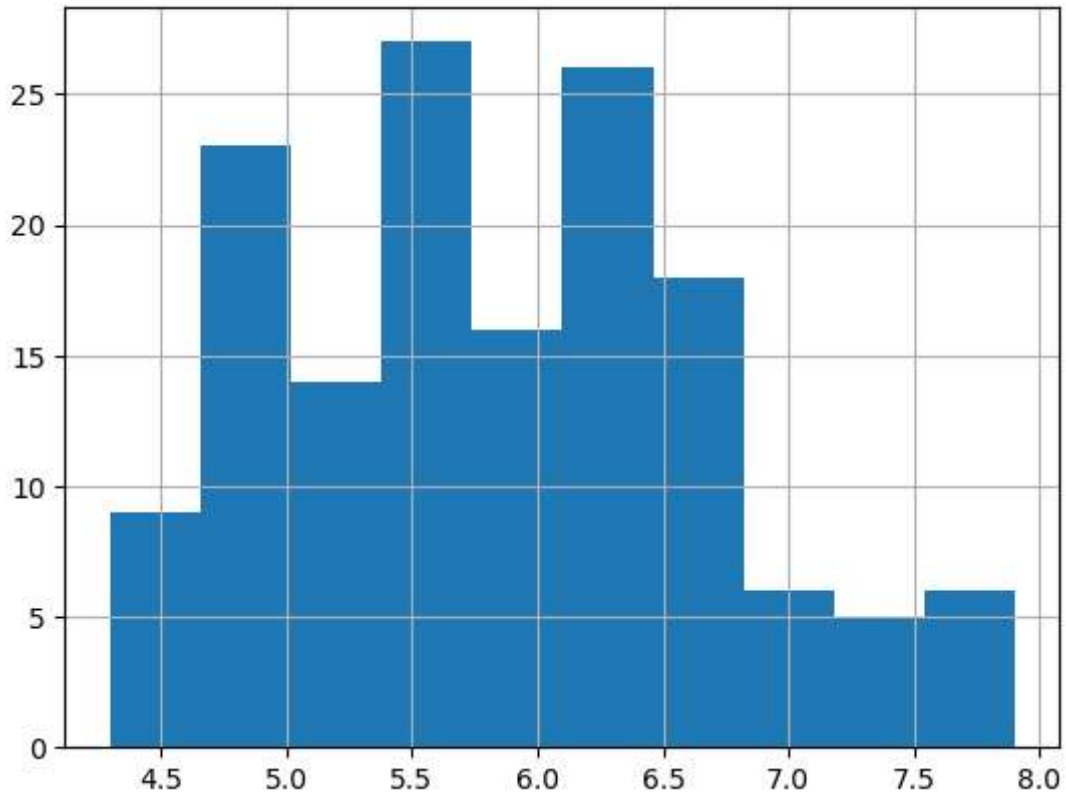
In [23]:
```
# to display no of sample on each class
df['Species'].value_counts()
```

Out[23]:
```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: Species, dtype: int64
```

In [25]:
```
#chech for null values
df.isnull().sum()
```

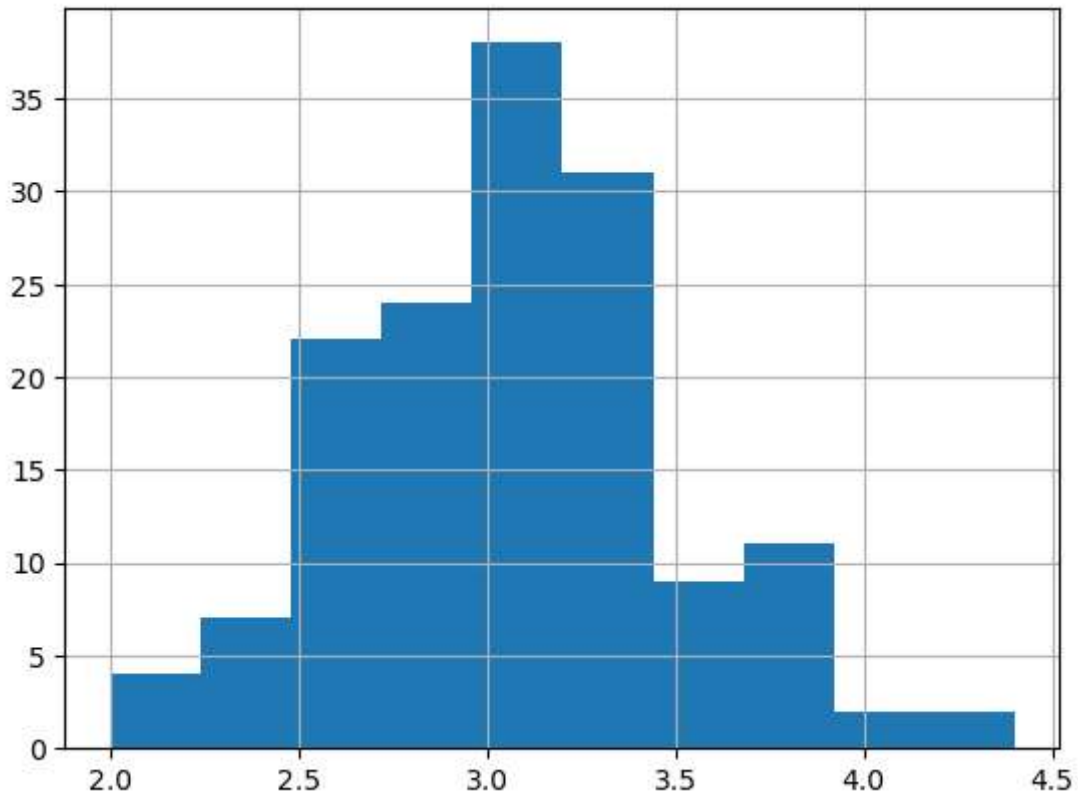
Out[25]:
```
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

In [56]:
```
#Histograms
df['SepalLengthCm'].hist()
plt.savefig('histogram')
plt.show()
```




In [29]:
```
df['SepalWidthCm'].hist()
```

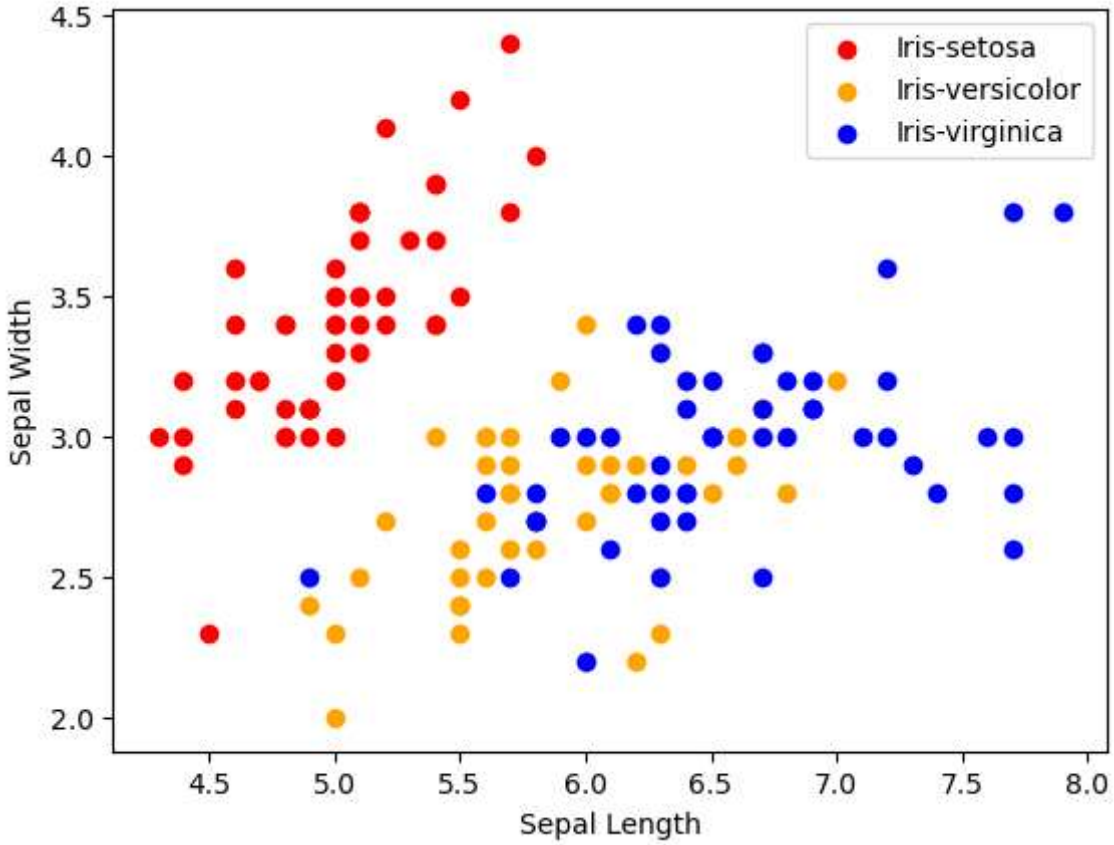
Out[29]:
```
<AxesSubplot:>
```



In [31]:
```
#Scatterplot
colors = ['red','orange','blue']
Species = ['Iris-setosa', 'Iris-versicolor','Iris-virginica']
```

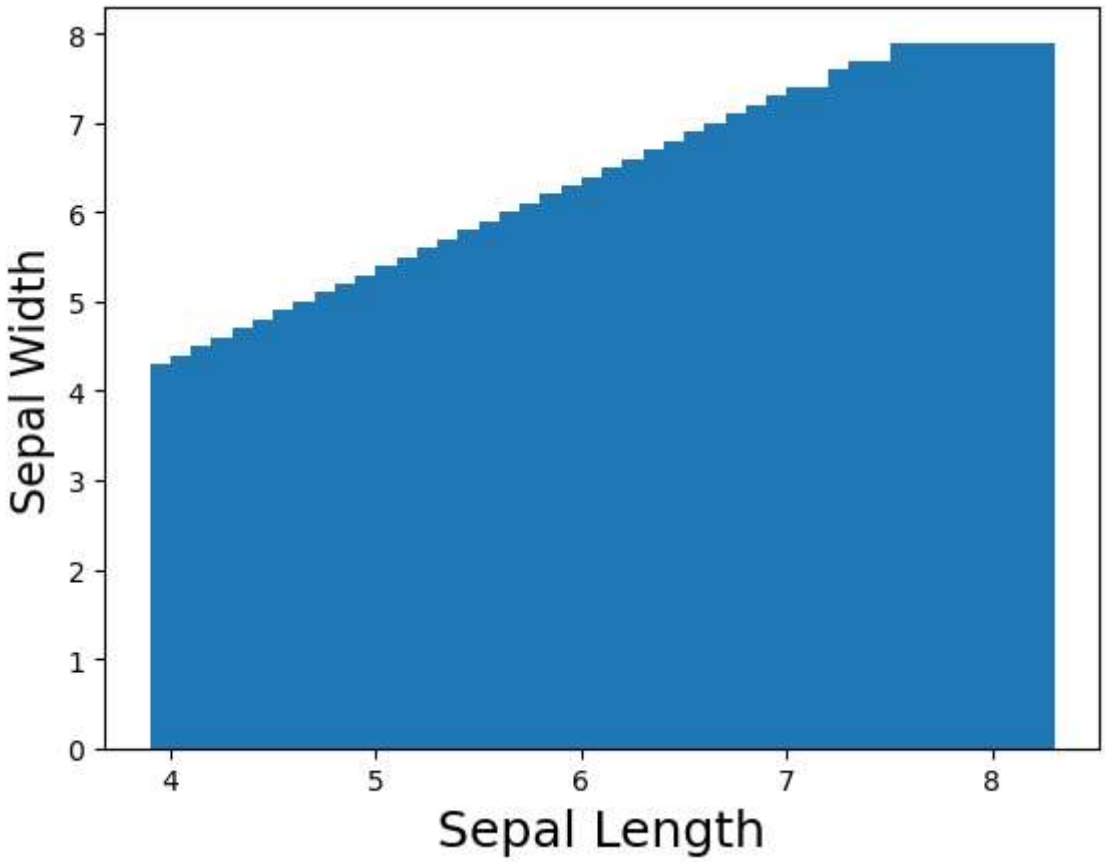
In [50]:
```
for i in range(3):
    x = df[df['Species']== Species[i]]
    plt.scatter(x['SepalLengthCm'], x['SepalWidthCm'], c = colors[i], label = Species[
plt.xlabel("Sepal Length")
plt.ylabel("Sepal Width")
plt.legend()
```

Out[50]:
```
<matplotlib.legend.Legend at 0x252021108e0>
```



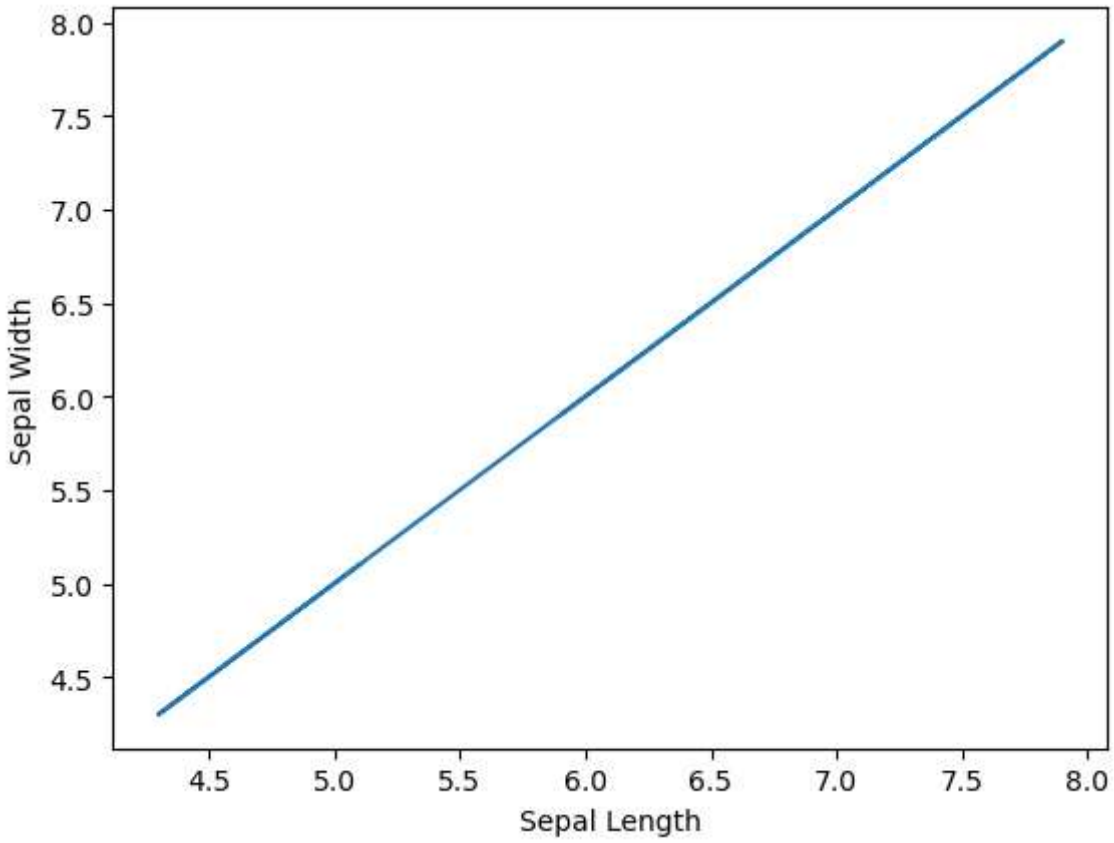

In [44]:
```
x = df['SepalLengthCm']
y = df['SepalLengthCm']

#Bar Graph
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.bar(x,y)
```

Out[44]:
```
<BarContainer object of 150 artists>
```




In [42]:
```
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.plot(x,y)
```

Out[42]:
```
[<matplotlib.lines.Line2D at 0x25200d11220>]
```


8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
Sepal Width
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
Sepal Length

In [ ]: