In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sb

cols = ['Sepal Length','Sepal Width','Petal Length','Petal Width','Species']
df = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/iris/ir
```

In [2]:

```
df.head()
```

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

In [3]:

```
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 [4]:

```
np.unique(df["Species"])
```

Out[4]:

```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

In [5]:

```
df.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

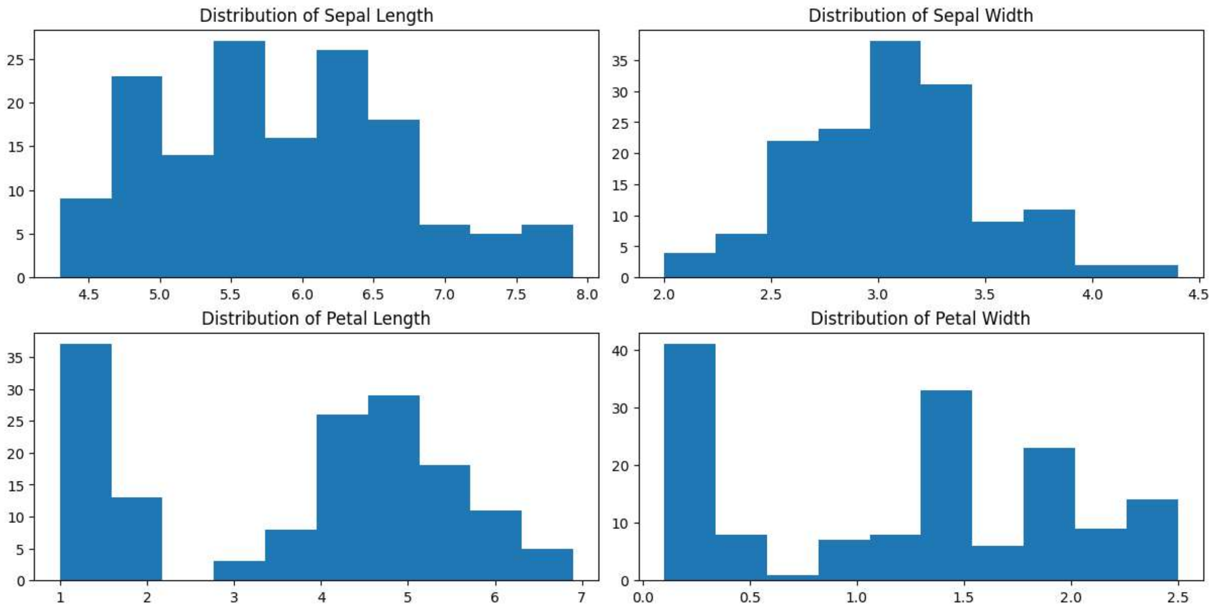
In [6]:

```
print("Creating a Histogram for each feature in the Iris dataset\n\n")

import matplotlib.pyplot as plt
fig,axes = plt.subplots(2,2,figsize=(12,6), constrained_layout = True)

for i in range(4):
    x, y = i//2, i%2
    axes[x,y].hist(df[df.columns[i]])
    axes[x,y].set_title(f"Distribution of {df.columns[i]}")
```

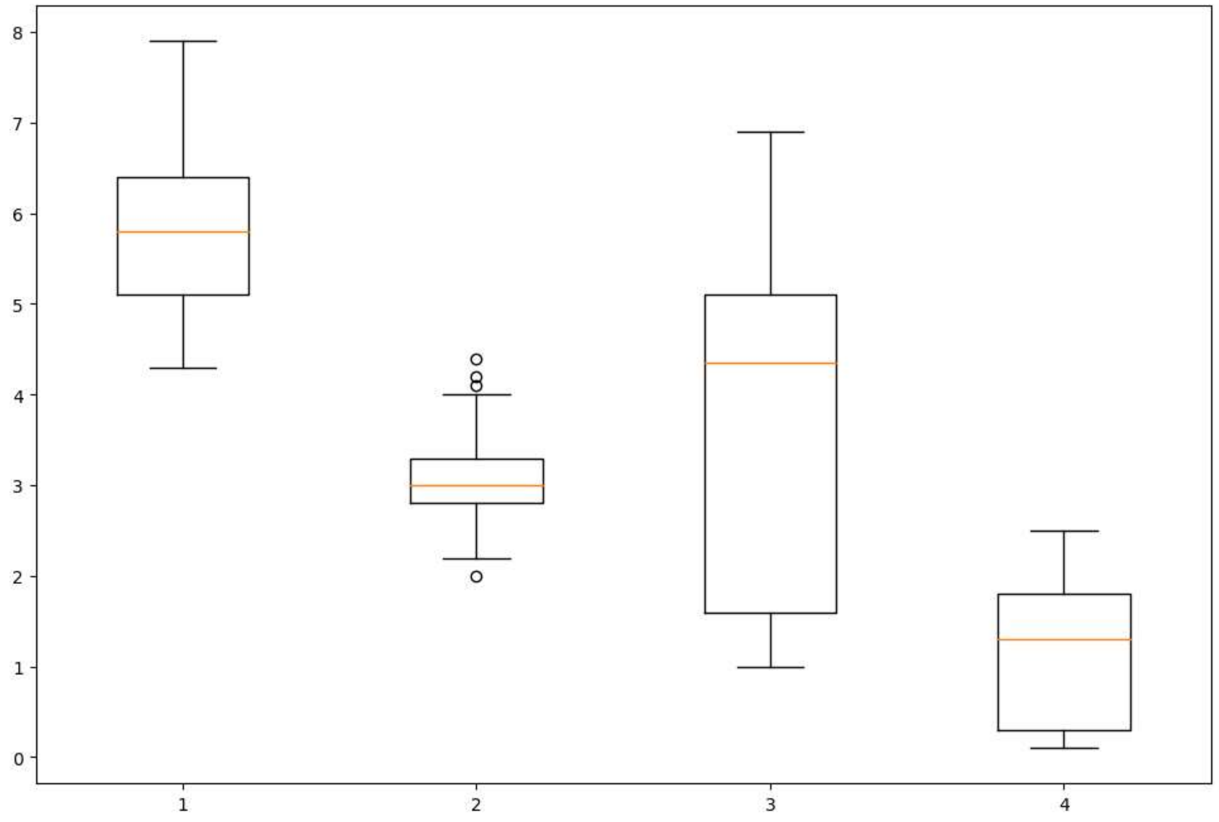
```
Creating a Histogram for each feature in the Iris dataset
```



In [7]:

```
data_to_plot = df[df.columns[0:-1]]

fig, axes = plt.subplots(1, figsize=(12,8))
bp = axes.boxplot(data_to_plot)
```



```
In [ ]:
```