Assignment 10

Data Visualization III Download the Iris flower dataset or any other dataset into a DataFrame. (eg https://archive.ics.uci.edu/ml/datasets/Iris (https://archive.ics.uci.edu/ml/datasets/Iris)). Scan the dataset and give the inference as:

1. How many features are there and what are their types (e.g., numeric, nominal)?
2. Create a histogram for each feature in the dataset to illustrate the feature distributions.
3. Create a boxplot for each feature in the dataset. Compare distributions and identify outliers.

In [4]:

```
import numpy as np
import pandas as pd
```

In [5]:

```
csv_url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'
df = pd.read_csv(csv_url, header = None)
col_names = ['Sepal_Length','Sepal_Width','Petal_Length','Petal_Width','Species']
df = pd.read_csv(csv_url, names = col_names)
```

In [6]:

```
df.head()
```

Out[6]:

	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

Q1. How many features are there and what are their types?

In [7]:

```
# to determine the length of lists in a pandas dataframe column
column = len(list(df))
```

In [8]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
Sepal_Length    150 non-null float64
Sepal_Width     150 non-null float64
Petal_Length    150 non-null float64
Petal_Width     150 non-null float64
Species         150 non-null object
dtypes: float64(4), object(1)
memory usage: 5.3+ KB
```

Hence the dataset contains 4 numerical columns and 1 object column

In [9]:

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

Out[9]:

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

Q2. Data Visualization-Create a histogram for each feature in the dataset to illustrate the feature distributions. Plot each histogram.

The Seaborn library is built on top of Matplotlib and offers many advanced data visualization capabilities.

Though, the Seaborn library can be used to draw a variety of charts such as matrix plots, grid plots, regression plots etc.,

In [ ]:

```
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
```

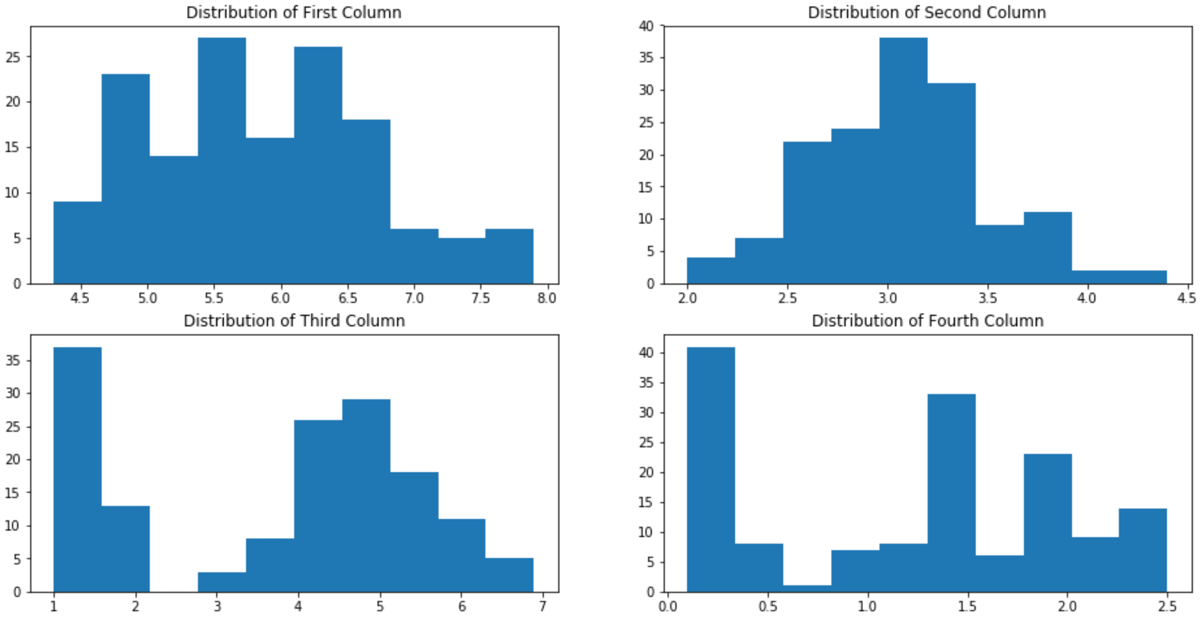
In [12]:
```
fig, axes = plt.subplots(2, 2, figsize=(16, 8))


axes[0,0].set_title("Distribution of First Column")
axes[0,0].hist(df["Sepal_Length"]);

axes[0,1].set_title("Distribution of Second Column")
axes[0,1].hist(df["Sepal_Width"]);

axes[1,0].set_title("Distribution of Third Column")
axes[1,0].hist(df["Petal_Length"]);

axes[1,1].set_title("Distribution of Fourth Column")
axes[1,1].hist(df["Petal_Width"]);
```



Distribution of First Column

Distribution of Second Column

Distribution of Third Column

Distribution of Fourth Column

Q4. Create a boxplot for each feature in the dataset. All of the boxplots should be combined into a single plot. Compare distributions and identify outliers.

seaborn.set_style(style=None, rc=None)

Parameters style: dict, or one of {darkgrid, whitegrid, dark, white, ticks} A dictionary of parameters or the name of a preconfigured style.

rc: dict, optional Parameter mappings to override the values in the preset seaborn style dictionaries. This only updates parameters that are considered part of the style definition.

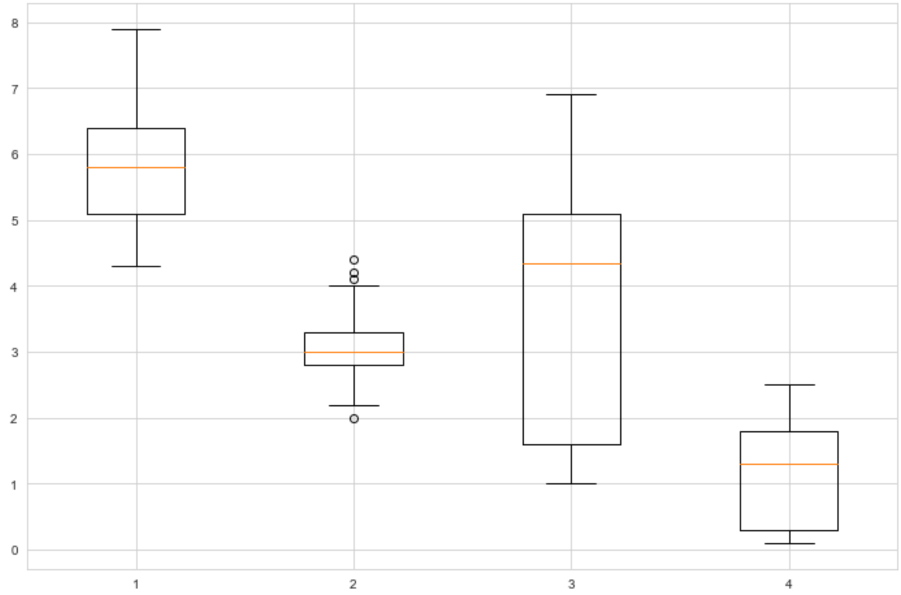
In [13]:

```
data_to_plot = [df["Sepal_Length"],df["Sepal_Width"],df["Petal_Length"],df["Petal_Width"]]

sns.set_style("whitegrid")
# Creating a figure instance
fig = plt.figure(1, figsize=(12,8))

# Creating an axes instance
ax = fig.add_subplot(111)

# Creating the boxplot
bp = ax.boxplot(data_to_plot);
```



In [ ]:

```
```