Data Visualization III

Download the Iris flower dataset or any other dataset into a DataFrame. (e.g., 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. List down the features and their types (e.g., numeric, nominal) available in the dataset.
2. Create a histogram for each feature in the dataset to illustrate the feature distributions.
3. Create a box plot for each feature in the dataset.
4. Compare distributions and identify outliers.

In [1]:

```
import pandas as pd
import numpy as np
csv_url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'
col_names = ['Sepal_Length','Sepal_Width','Petal_Length','Petal_Width','Species']
```

In [2]:

```
iris = pd.read_csv(csv_url, names = col_names)
```

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

In [3]:

```
column = len(list(iris))
column
```

Out[3]:

```
5
```

Clearly, dataset has 5 column indicating 5 features about the data

In [4]:

```
iris.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
```

Hence the dataset contains 4 numerical columns and 1 object column

In [6]:

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

Out[6]:

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

Q2. Compute and display summary statistics for each feature available in the dataset.

In [13]:

```
iris.describe()
```

Out[13]:

	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

Q3. Data Visualization-Create a histogram for each feature in the dataset to illustrate the feature distributions. Plot each histogram.

In [8]:

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

In [10]:

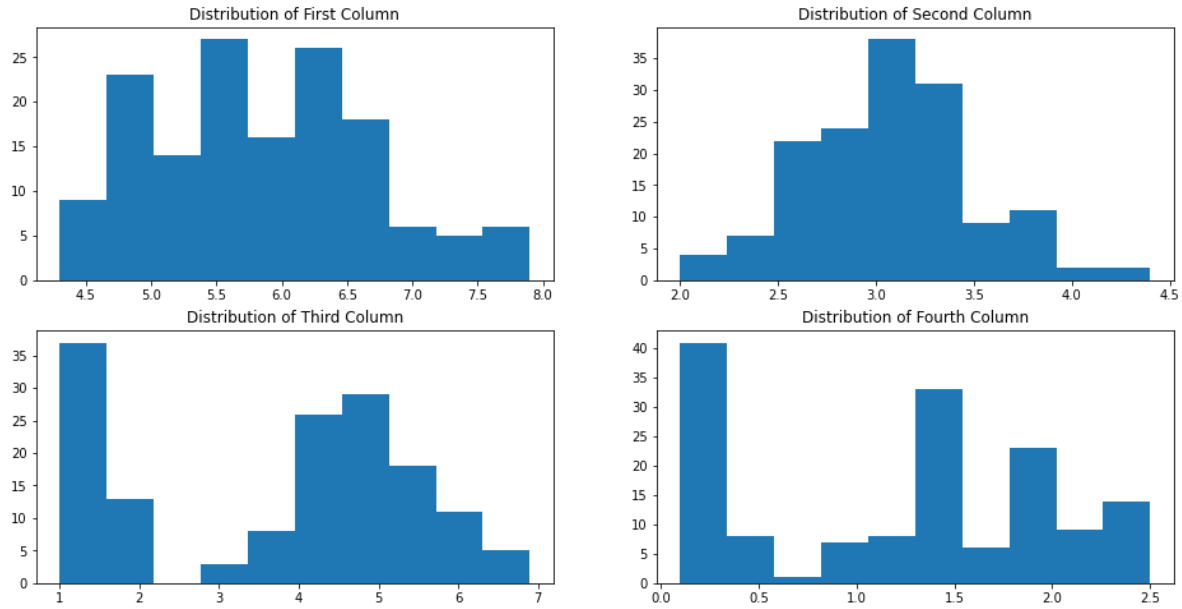
```
fig, axes = plt.subplots(2, 2, figsize=(16, 8))


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

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

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

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



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.

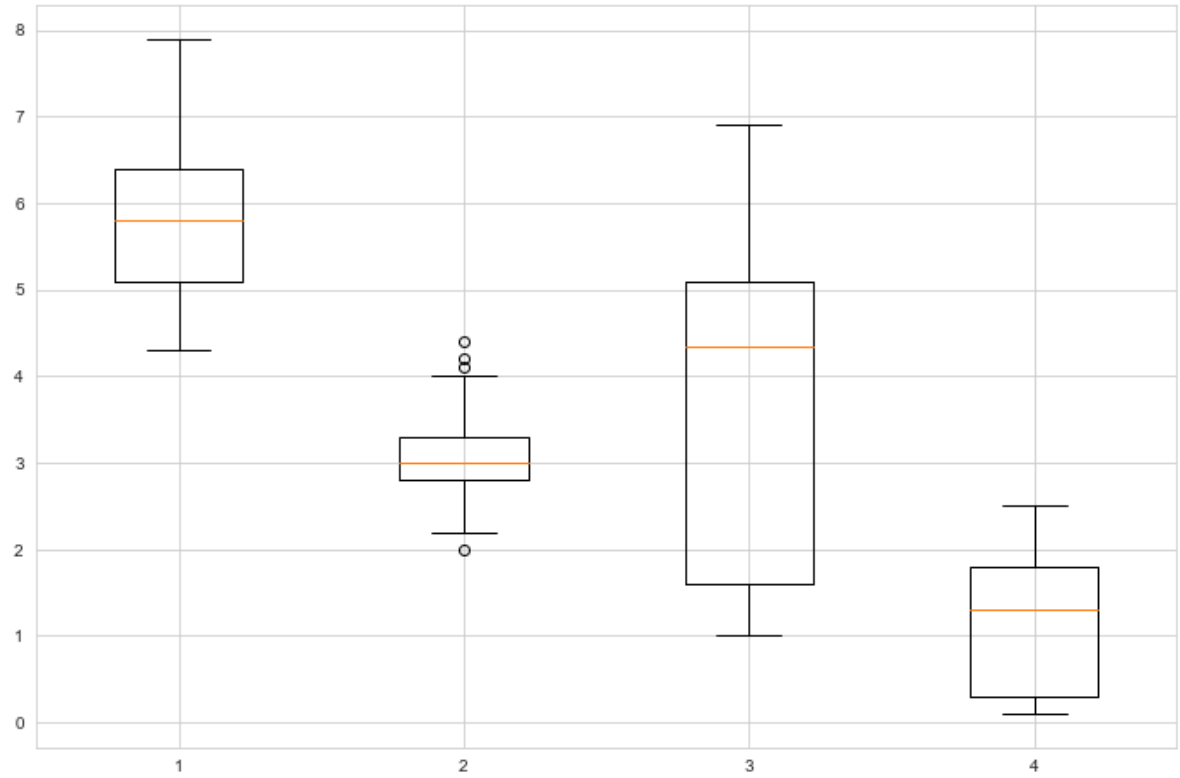
In [12]:

```
data_to_plot = [iris["Sepal_Length"],iris["Sepal_Width"],iris["Petal_Length"],iris["Petal_W

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);
```



If we observe closely. for the box 2, interquartile distance is roughly around 0.75 hence the values lying beyond this range of (third quartile + interquartile distance) i.e. roughly around 4.05 will be considered as outliers. Similarly outliers with other boxplots can be found.