```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

```
df = pd.read_csv('iris.csv')
df.head()
```

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

How many features are there and what are their types (e.g., numeric, nominal)?

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             150 non-null    int64
 1   SepalLengthCm  150 non-null    float64
 2   SepalWidthCm   150 non-null    float64
 3   PetalLengthCm  150 non-null    float64
 4   PetalWidthCm   150 non-null    float64
 5   Species        150 non-null    object
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

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

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

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

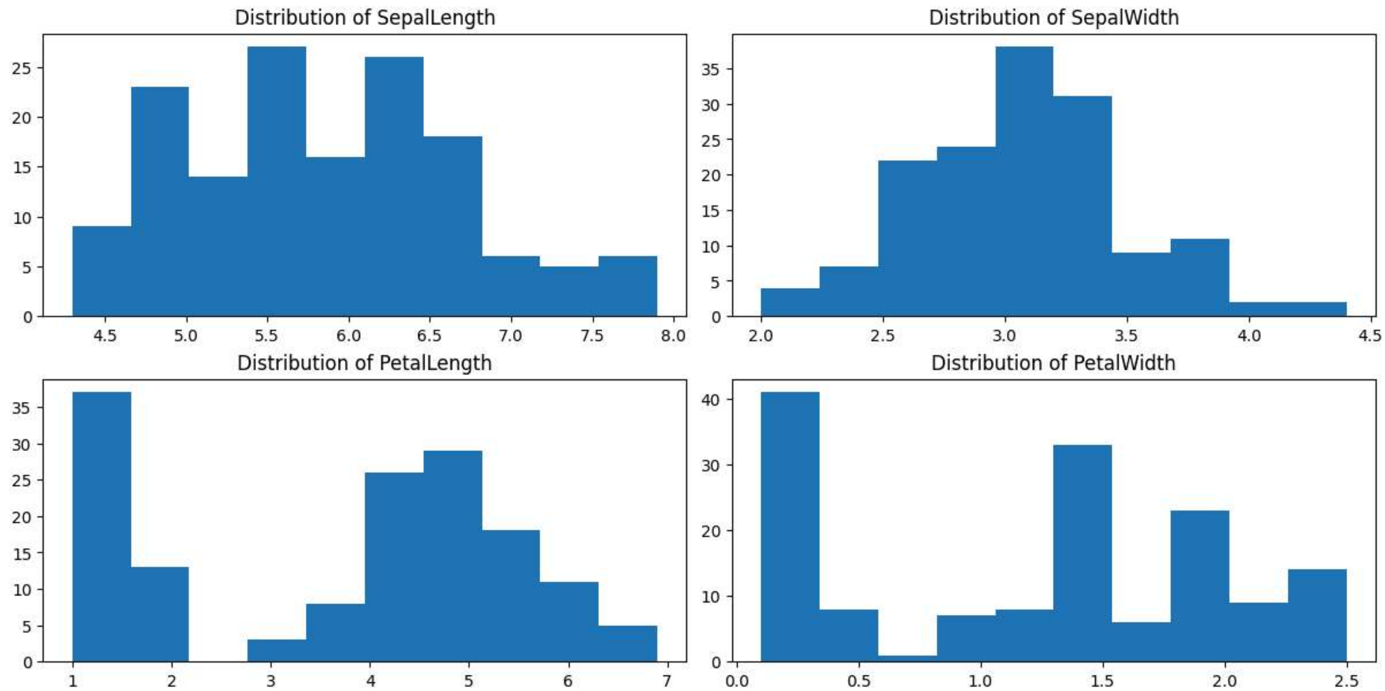
minimum value, maximum value, mean, range, standard deviation, variance and percentiles

```
df.describe()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

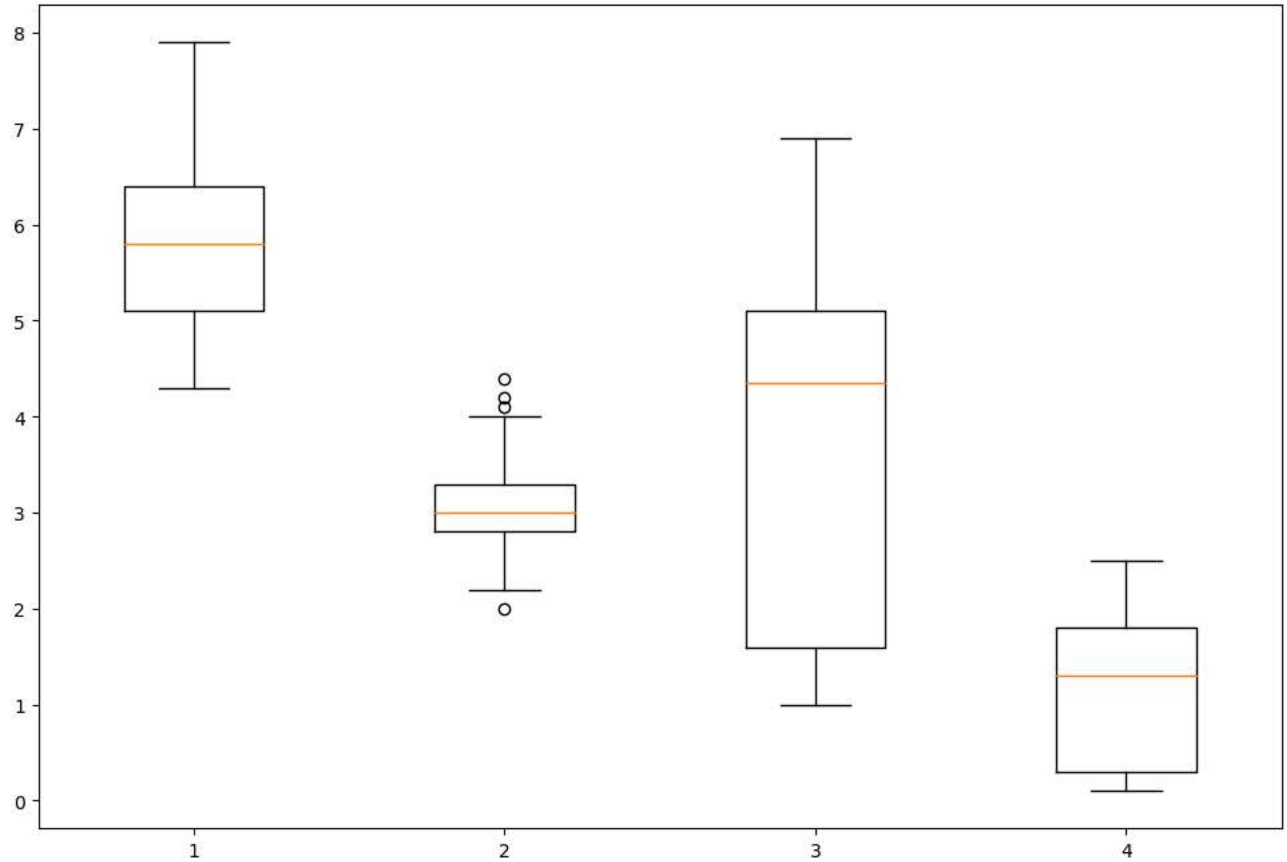
Data Visualization-Create a histogram for each feature in the dataset to illustrate the feature distributions

```
import seaborn as sns
import matplotlib
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 + 1]])
    _ = axes[x, y].set_title(f"Distribution of {df.columns[i + 1][:-2]}")
```



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.

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

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