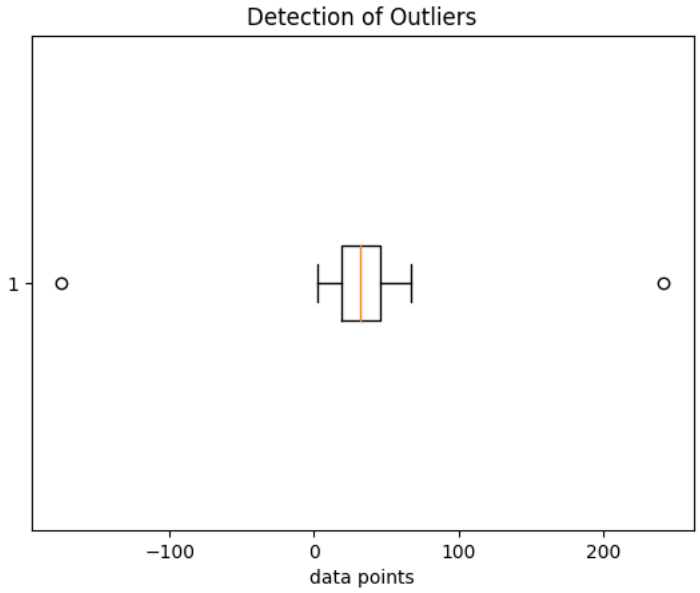
```
#using defined data
import matplotlib.pyplot as plt
import numpy as np

data = [34, 2, 3, 43, 54, 67, 34, -175, 25, 26, 30, 242]
print("data_points", data)

# Boxplot
plt.boxplot(data, vert=False)
plt.title("Detection of Outliers")
plt.xlabel("data points")
plt.show()
```

```
data_points [34, 2, 3, 43, 54, 67, 34, -175, 25, 26, 30, 242]
```




```
mean = np.mean(data)
standard_deviation = np.std(data)
print("Mean:", mean)
print("Standard Deviation:", standard_deviation)

# Display threshold for Z-score method
threshold = 2
print("Threshold (Z-score method):", threshold)

# Z-score method
outliers_zscore = [i for i in data if np.abs((i - mean) / standard_deviation) > threshold]
print("OUTLIERS (Z-score method):", outliers_zscore)
```

```
Mean: 32.083333333333336
Standard Deviation: 86.89884764611222
Threshold (Z-score method): 2
OUTLIERS (Z-score method): [-175, 242]
```

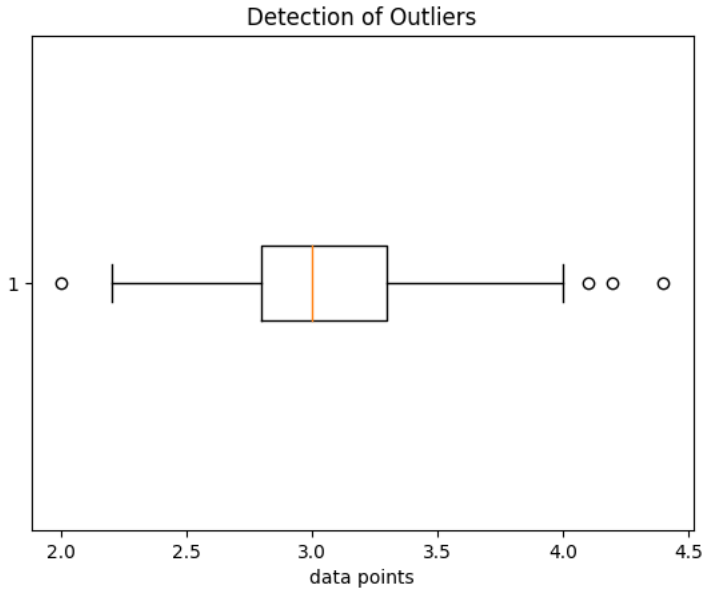
```
# IQR method
sample = np.sort(data)
Q1 = np.percentile(sample, 25)
Q3 = np.percentile(sample, 75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Display Q1, Q3, IQR, lower bound, and upper bound
print("Q1:", Q1)
print("Q3:", Q3)
print("IQR:", IQR)
print("Lower Bound:", lower_bound)
print("Upper Bound:", upper_bound)

# Display threshold for IQR method
threshold_iqr = 1.5
print("Threshold (IQR method):", threshold_iqr)

outliers_iqr = [i for i in sample if i < lower_bound or i > upper_bound]
print("OUTLIERS (IQR method):", outliers_iqr)
```

```
Q1: 19.5
Q3: 45.75
IQR: 26.25
Lower Bound: -19.875
Upper Bound: 85.125
Threshold (IQR method): 1.5
OUTLIERS (IQR method): [-175, 242]
```

USING IRIS DATASET

```
from sklearn.datasets import load_iris
import matplotlib.pyplot as plt
import numpy as np

# Load the Iris dataset
iris_data = load_iris()
data = iris_data.data[:, 1]
print("data_points", data)

# Boxplot
plt.boxplot(data, vert=False)
plt.title("Detection of Outliers")
plt.xlabel("data points")
plt.show()

# Display mean and standard deviation
mean = np.mean(data)
standard_deviation = np.std(data)
print("Mean:", mean)
print("Standard Deviation:", standard_deviation)

# Display threshold for Z-score method
threshold_zscore = 2
print("Threshold (Z-score method):", threshold_zscore)

# Z-score method
outliers_zscore = [i for i in data if np.abs((i - mean) / standard_deviation) > threshold_zscore]
print("OUTLIERS (Z-score method):", outliers_zscore)

# IQR method
Q1 = np.percentile(data, 25)
Q3 = np.percentile(data, 75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Display Q1, Q3, IQR, lower bound, and upper bound
print("Q1:", Q1)
print("Q3:", Q3)
print("IQR:", IQR)
print("Lower Bound:", lower_bound)
print("Upper Bound:", upper_bound)

# Display threshold for IQR method
threshold_iqr = 1.5
print("Threshold (IQR method):", threshold_iqr)

outliers_iqr = [i for i in data if i < lower_bound or i > upper_bound]
print("OUTLIERS (IQR method):", outliers_iqr)
```

```
data_points [3.5 3.  3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 3.7 3.4 3.  3.  4.  4.4 3.9 3.5
 3.8 3.8 3.4 3.7 3.6 3.3 3.4 3.  3.4 3.5 3.4 3.2 3.1 3.4 4.1 4.2 3.1 3.2
 3.5 3.6 3.  3.4 3.5 2.3 3.2 3.5 3.8 3.  3.8 3.2 3.7 3.3 3.2 3.2 3.1 2.3
 2.8 2.8 3.3 2.4 2.9 2.7 2.  3.  2.2 2.9 2.9 3.1 3.  2.7 2.2 2.5 3.2 2.8
 2.5 2.8 2.9 3.  2.8 3.  2.9 2.6 2.4 2.4 2.7 2.7 3.  3.4 3.1 2.3 3.  2.5
 2.6 3.  2.6 2.3 2.7 3.  2.9 2.9 2.5 2.8 3.3 2.7 3.  2.9 3.  3.  2.5 2.9
 2.5 3.6 3.2 2.7 3.  2.5 2.8 3.2 3.  3.8 2.6 2.2 3.2 2.8 2.8 2.7 3.3 3.2
 2.8 3.  2.8 3.  2.8 3.8 2.8 2.8 2.6 3.  3.4 3.1 3.  3.1 3.1 3.1 2.7 3.2
 3.3 3.  2.5 3.  3.4 3. ]
```




```
Mean: 3.0573333333333337
Standard Deviation: 0.4344109677354946
Threshold (Z-score method): 2
OUTLIERS (Z-score method): [4.0, 4.4, 4.1, 4.2, 2.0]
Q1: 2.8
Q3: 3.3
IQR: 0.5
Lower Bound: 2.05
Upper Bound: 4.05
Threshold (IQR method): 1.5
OUTLIERS (IQR method): [4.4, 4.1, 4.2, 2.0]
```