assignment5

May 2, 2024

[2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[3]:
```
df = pd.read_csv("Iris.csv")
```

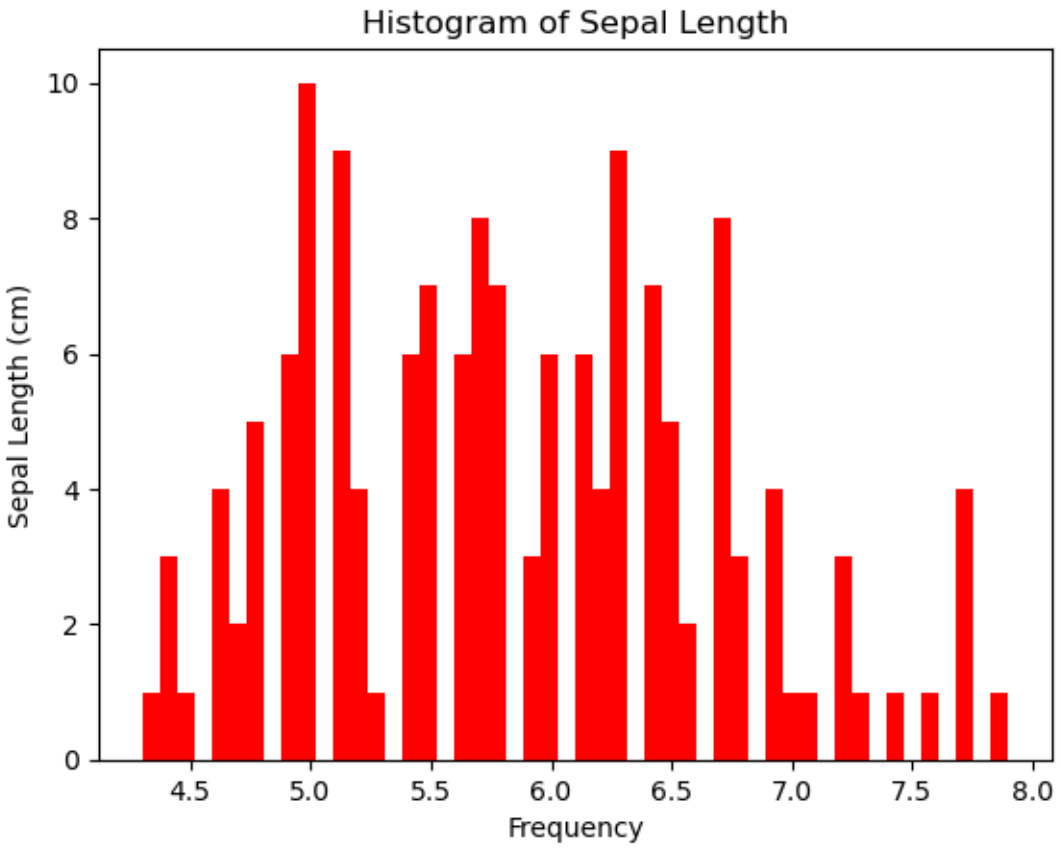
[4]:
```
df.info()
```

```
<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
```

[25]:
```
sepalLengthCm = df["SepalLengthCm"]
```

[26]:
```
plt.hist(sepalLengthCm, bins=50,  color="red")
plt.title("Histogram of Sepal Length")
plt.xlabel("Frequency")
plt.ylabel("Sepal Length (cm)")
```

[26]:
```
Text(0, 0.5, 'Sepal Length (cm)')
```



[7]:
```
df
```

[7]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	
..	…	…	…	…	…	
145	146	6.7	3.0	5.2	2.3	
146	147	6.3	2.5	5.0	1.9	
147	148	6.5	3.0	5.2	2.0	
148	149	6.2	3.4	5.4	2.3	
149	150	5.9	3.0	5.1	1.8	

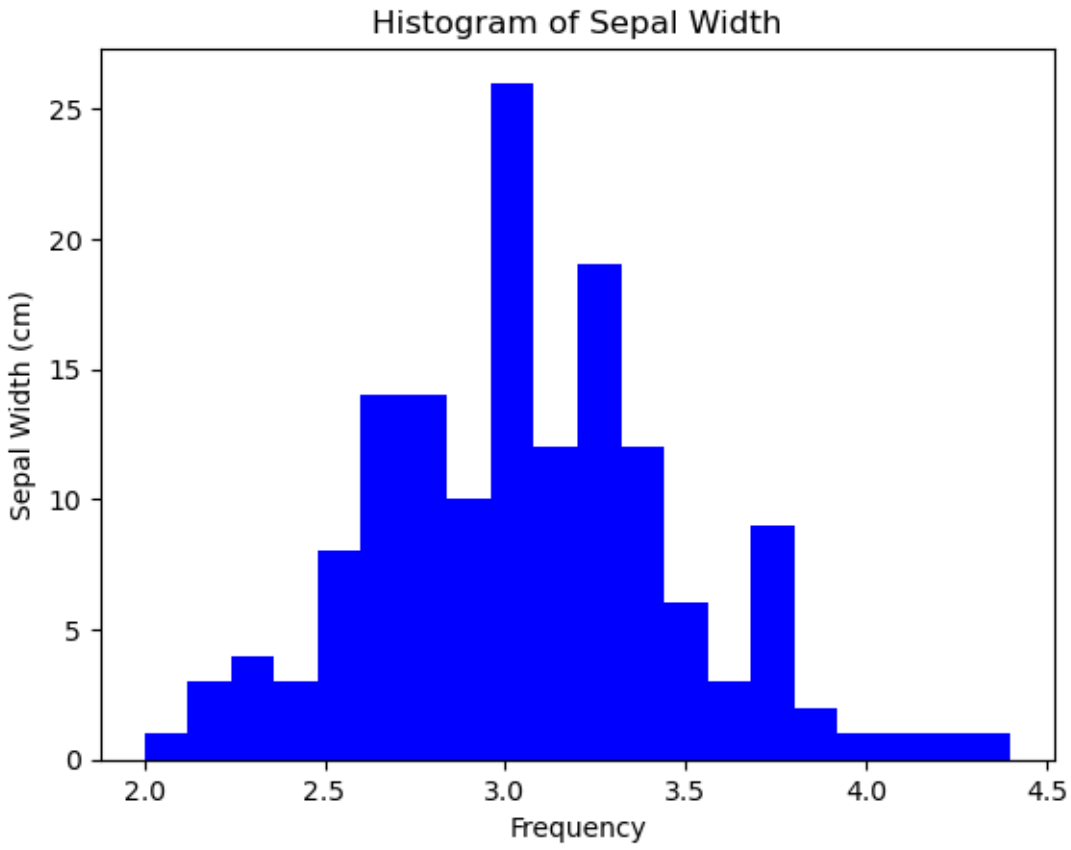
```
4       Iris-setosa
..              …
145  Iris-virginica
146  Iris-virginica
147  Iris-virginica
148  Iris-virginica
149  Iris-virginica

[150 rows x 6 columns]
```

[12]:
```
sepalWidthCm = df["SepalWidthCm"]
petalLengthCm = df["PetalLengthCm"]
petalWidthCm = df["PetalWidthCm"]
```

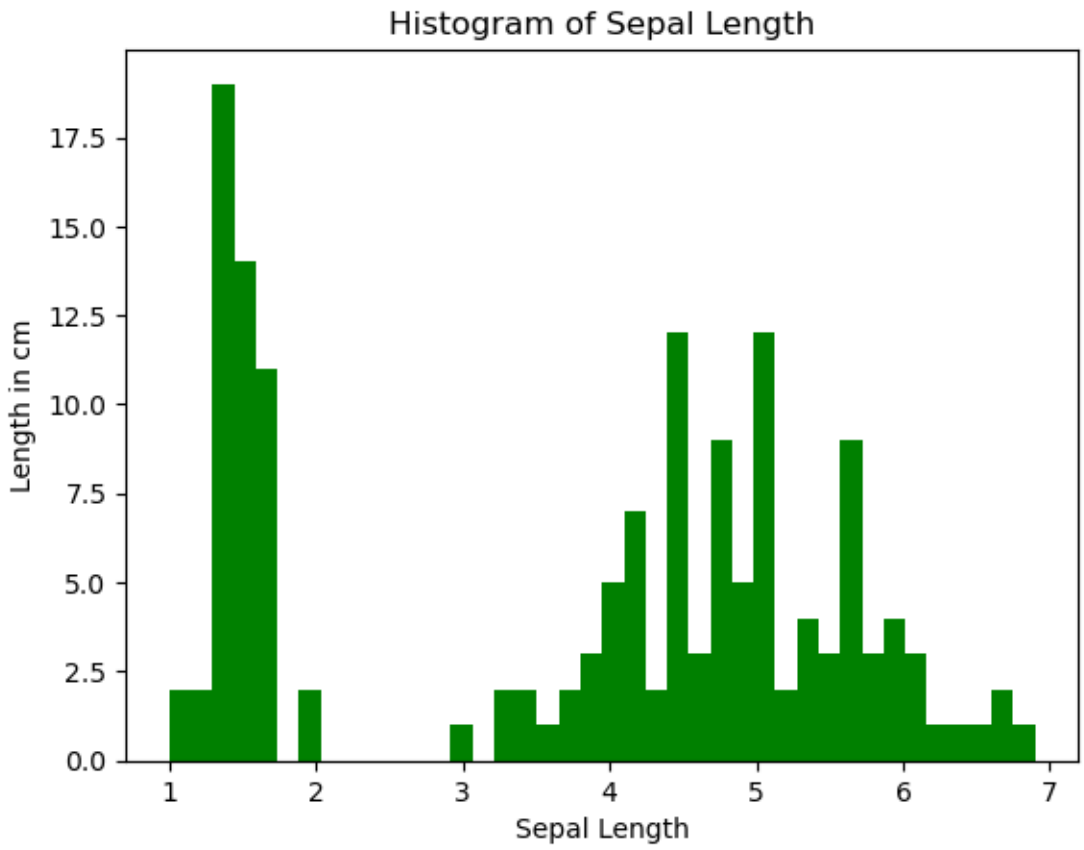
[21]:
```
plt.hist(sepalWidthCm, bins=20, color="blue")
plt.title("Histogram of Sepal Width")
plt.xlabel("Frequency")
plt.ylabel("Sepal Width (cm)")
```

[21]:
```
Text(0, 0.5, 'Sepal Width (cm)')
```



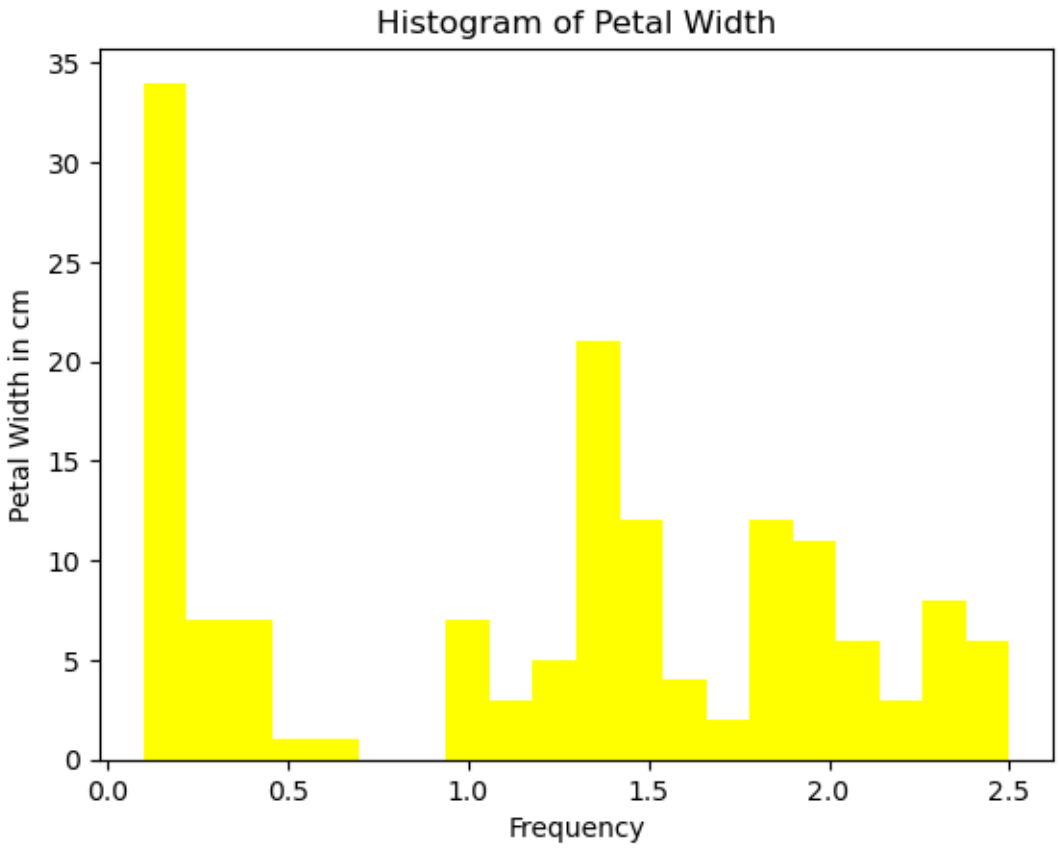
[23]:
```
plt.hist(petalLengthCm, bins=40, color="green")
plt.title("Histogram of Sepal Length")
plt.xlabel("Sepal Length")
plt.ylabel("Length in cm")
```

[23]: Text(0, 0.5, 'Length in cm')



[24]:
```
plt.hist(petalWidthCm, bins=20, color="yellow")
plt.title("Histogram of Petal Width")
plt.xlabel("Frequency")
plt.ylabel("Petal Width in cm")
```

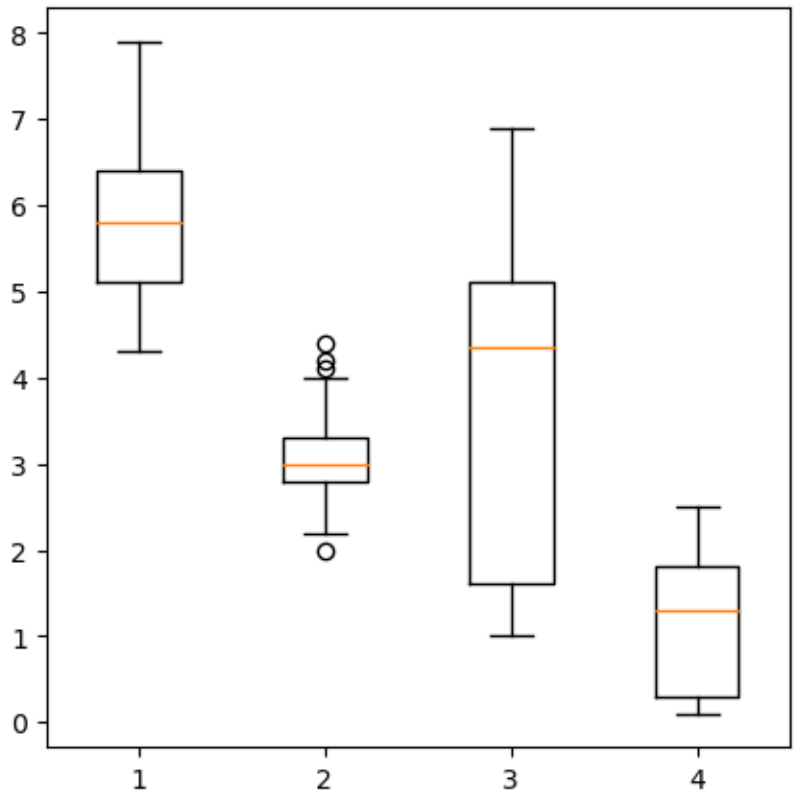
[24]: Text(0, 0.5, 'Petal Width in cm')



[31]:
```
fig = plt.figure(figsize=(5,5))
plt.boxplot([sepalLengthCm, sepalWidthCm, petalLengthCm, petalWidthCm])
```

[31]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x24598a9bac0>,
  <matplotlib.lines.Line2D at 0x24598a9bd90>,
  <matplotlib.lines.Line2D at 0x24598aaaeb0>,
  <matplotlib.lines.Line2D at 0x24598ab81c0>,
  <matplotlib.lines.Line2D at 0x24598ac32e0>,
  <matplotlib.lines.Line2D at 0x24598ac35b0>,
  <matplotlib.lines.Line2D at 0x24598ad1700>,
  <matplotlib.lines.Line2D at 0x24598ad19d0>],
 'caps': [<matplotlib.lines.Line2D at 0x24598aaa0a0>,
  <matplotlib.lines.Line2D at 0x24598aaa370>,
  <matplotlib.lines.Line2D at 0x24598ab8490>,
  <matplotlib.lines.Line2D at 0x24598ab8760>,
  <matplotlib.lines.Line2D at 0x24598ac3880>,
  <matplotlib.lines.Line2D at 0x24598ac3b50>,
  <matplotlib.lines.Line2D at 0x24598ad1ca0>,
  <matplotlib.lines.Line2D at 0x24598ad1f70>],
 'boxes': [<matplotlib.lines.Line2D at 0x24598a9b7f0>,
```

```
 <matplotlib.lines.Line2D at 0x24598aaabe0>,
 <matplotlib.lines.Line2D at 0x24598ab8fd0>,
 <matplotlib.lines.Line2D at 0x24598ad1430>],
'medians': [<matplotlib.lines.Line2D at 0x24598aaa640>,
 <matplotlib.lines.Line2D at 0x24598ab8a30>,
 <matplotlib.lines.Line2D at 0x24598ac3e50>,
 <matplotlib.lines.Line2D at 0x24598adf280>],
'fliers': [<matplotlib.lines.Line2D at 0x24598aaa910>,
 <matplotlib.lines.Line2D at 0x24598ab8d00>,
 <matplotlib.lines.Line2D at 0x24598ad1160>,
 <matplotlib.lines.Line2D at 0x24598adf550>],
'means': []}
```



OUTLIERS

1.) OUTLIERS IN SEPAL LENGTH (CM)

[50]:
```
Q1 = np.percentile(df["SepalLengthCm"], 25, interpolation="midpoint")
Q3 = np.percentile(df["SepalLengthCm"], 75, interpolation="midpoint")
IQR = Q3 - Q1
print("Quartile 1: ",Q1)
print("Quartile 3: ",Q3)
```

```
print("Inter Quartile Range: ",IQR)

low_limit = Q1 - 1.5 * IQR
up_limit = Q3 + 1.5 * IQR
print("Lower Limit is: ",low_limit)
print("Upper Limit is: ", up_limit)

outliers = []
for x in df["SepalLengthCm"]:
    if((x>up_limit) or (x<low_limit)):
        outliers.append(x)

print("Outliers in the dataset are: ", outliers)
```

```
Quartile 1:  5.1
Quartile 3:  6.4
Inter Quartile Range:  1.3000000000000007
Lower Limit is:  3.1499999999999986
Upper Limit is:  8.350000000000001
Outliers in the dataset are:  []
```

2.) OUTLIERs IN SEPAL WIDTH (CM)

[49]:
```
Q1_SepalWidthCm = np.percentile(df["SepalWidthCm"], 25,␣
 ↪interpolation="midpoint")
Q3_SepalWidthCm = np.percentile(df["SepalWidthCm"], 75,␣
 ↪interpolation="midpoint")
IQR = Q3 - Q1
print("Quartile 1: ",Q1)
print("Quartile 3: ",Q3)
print("Inter Quartile Range: ",IQR)

low_limit = Q1 - 1.5 * IQR
up_limit = Q3 + 1.5 * IQR
print("Lower Limit is: ",low_limit)
print("Upper Limit is: ", up_limit)

outliers = []
for x in df["SepalWidthCm"]:
    if((x>up_limit) or (x<low_limit)):
        outliers.append(x)

print("Outliers in the Sepal Width (Cm) are: ", outliers)
```

```
Quartile 1:  5.1
Quartile 3:  6.4
Inter Quartile Range:  1.3000000000000007
Lower Limit is:  3.1499999999999986
```

```
Upper Limit is:  8.350000000000001
Outliers in the Sepal Width (Cm) are:  [3.0, 3.1, 2.9, 3.1, 3.0, 3.0, 3.0, 3.1,
3.1, 3.1, 3.0, 2.3, 3.0, 3.1, 2.3, 2.8, 2.8, 2.4, 2.9, 2.7, 2.0, 3.0, 2.2, 2.9,
2.9, 3.1, 3.0, 2.7, 2.2, 2.5, 2.8, 2.5, 2.8, 2.9, 3.0, 2.8, 3.0, 2.9, 2.6, 2.4,
2.4, 2.7, 2.7, 3.0, 3.1, 2.3, 3.0, 2.5, 2.6, 3.0, 2.6, 2.3, 2.7, 3.0, 2.9, 2.9,
2.5, 2.8, 2.7, 3.0, 2.9, 3.0, 3.0, 2.5, 2.9, 2.5, 2.7, 3.0, 2.5, 2.8, 3.0, 2.6,
2.2, 2.8, 2.8, 2.7, 2.8, 3.0, 2.8, 3.0, 2.8, 2.8, 2.8, 2.6, 3.0, 3.1, 3.0, 3.1,
3.1, 3.1, 2.7, 3.0, 2.5, 3.0, 3.0]
```

3.) OUTLIERS IN PETAL LENGTH (CM)

[48]:
```
Q1_PetalLengthCm = np.percentile(df["PetalLengthCm"], 25,␣
 ↪interpolation="midpoint")
Q3_PetalLengthCm = np.percentile(df["PetalLengthCm"], 75,␣
 ↪interpolation="midpoint")
IQR = Q3 - Q1
print("Quartile 1: ",Q1)
print("Quartile 3: ",Q3)
print("Inter Quartile Range: ",IQR)

low_limit = Q1 - 1.5 * IQR
up_limit = Q3 + 1.5 * IQR
print("Lower Limit is: ",low_limit)
print("Upper Limit is: ", up_limit)

outliers = []
for x in df["PetalLengthCm"]:
    if((x>up_limit) or (x<low_limit)):
        outliers.append(x)

print("Outliers in the Petal Width (Cm) are: ", outliers)
```

```
Quartile 1:  5.1
Quartile 3:  6.4
Inter Quartile Range:  1.3000000000000007
Lower Limit is:  3.1499999999999986
Upper Limit is:  8.350000000000001
Outliers in the Petal Width (Cm) are:  [1.4, 1.4, 1.3, 1.5, 1.4, 1.7, 1.4, 1.5,
1.4, 1.5, 1.5, 1.6, 1.4, 1.1, 1.2, 1.5, 1.3, 1.4, 1.7, 1.5, 1.7, 1.5, 1.0, 1.7,
1.9, 1.6, 1.6, 1.5, 1.4, 1.6, 1.6, 1.5, 1.5, 1.4, 1.5, 1.2, 1.3, 1.5, 1.3, 1.5,
1.3, 1.3, 1.3, 1.6, 1.9, 1.4, 1.6, 1.4, 1.5, 1.4, 3.0]
```

4.) OUTLIERS IN PETAL WIDTH (CM)

[51]:
```
Q1_PetalWidthCm = np.percentile(df["PetalWidthCm"], 25,␣
 ↪interpolation="midpoint")
Q3_PetalWidthCm = np.percentile(df["PetalWidthCm"], 75,␣
 ↪interpolation="midpoint")
```

```
IQR = Q3 - Q1
print("Quartile 1: ",Q1)
print("Quartile 3: ",Q3)
print("Inter Quartile Range: ",IQR)

low_limit = Q1 - 1.5 * IQR
up_limit = Q3 + 1.5 * IQR
print("Lower Limit is: ",low_limit)
print("Upper Limit is: ", up_limit)

outliers = []
for x in df["PetalWidthCm"]:
    if((x>up_limit) or (x<low_limit)):
        outliers.append(x)

print("Outliers in the Petal Width (Cm) are: ", outliers)
```

```
Quartile 1:  5.1
Quartile 3:  6.4
Inter Quartile Range:  1.3000000000000007
Lower Limit is:  3.1499999999999986
Upper Limit is:  8.350000000000001
Outliers in the Petal Width (Cm) are:  [0.2, 0.2, 0.2, 0.2, 0.2, 0.4, 0.3, 0.2,
0.2, 0.1, 0.2, 0.2, 0.1, 0.1, 0.2, 0.4, 0.4, 0.3, 0.3, 0.3, 0.2, 0.4, 0.2, 0.5,
0.2, 0.2, 0.4, 0.2, 0.2, 0.2, 0.2, 0.4, 0.1, 0.2, 0.1, 0.2, 0.2, 0.1, 0.2, 0.2,
0.3, 0.3, 0.2, 0.6, 0.4, 0.3, 0.2, 0.2, 0.2, 0.2, 1.4, 1.5, 1.5, 1.3, 1.5, 1.3,
1.6, 1.0, 1.3, 1.4, 1.0, 1.5, 1.0, 1.4, 1.3, 1.4, 1.5, 1.0, 1.5, 1.1, 1.8, 1.3,
1.5, 1.2, 1.3, 1.4, 1.4, 1.7, 1.5, 1.0, 1.1, 1.0, 1.2, 1.6, 1.5, 1.6, 1.5, 1.3,
1.3, 1.3, 1.2, 1.4, 1.2, 1.0, 1.3, 1.2, 1.3, 1.3, 1.1, 1.3, 2.5, 1.9, 2.1, 1.8,
2.2, 2.1, 1.7, 1.8, 1.8, 2.5, 2.0, 1.9, 2.1, 2.0, 2.4, 2.3, 1.8, 2.2, 2.3, 1.5,
2.3, 2.0, 2.0, 1.8, 2.1, 1.8, 1.8, 1.8, 2.1, 1.6, 1.9, 2.0, 2.2, 1.5, 1.4, 2.3,
2.4, 1.8, 1.8, 2.1, 2.4, 2.3, 1.9, 2.3, 2.5, 2.3, 1.9, 2.0, 2.3, 1.8]
```

[ ]: