

Finding outlier for dataSet and boxplot

IQR

1. Sort the data
2. Calculate Q1(25%) and Q3(75%)
3. IQR (Q3 -Q1)
4. Find the Lower Fence (Q1 - 1.5(IQR))
5. Find the Uper Fence (Q3 - 1.5(IQR))

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

dataSet=
[11,10,12,14,12,15,14,13,15,102,12,17,19,107,10,13,12,14,12,108,12,13]

# sort the data #by default ascending
data = sorted(dataSet)

data

[10,
 10,
 11,
 12,
 12,
 12,
 12,
 12,
 12,
 12,
 13,
 13,
 13,
 13,
 14,
 14,
 14,
 14,
 15,
 15,
 17,
 19,
 102,
 107,
 108]

# Calculate Q1(25%) and Q3(75%)
# np lib
```

```

Q1,Q3 = np.percentile(data,[25,75])

Q1
np.float64(12.0)
Q3
np.float64(15.0)
#IQR (Q3 -Q1)
IQR = Q3 -Q1
print(IQR) # value return
3.0
# 4. Find the Lower Fence (Q1 - 1.5(IQR))

LF = Q1 -1.5*IQR
print(LF)
7.5
# 5. Find the Uper Fence (Q3 - 1.5(IQR))
UF = Q3 +1.5*IQR
print(UF)
19.5
# write a function in python that will filter the non outlier element
for data

def iqr_non_outliers(dataSet):
    Q1,Q3 = np.percentile(data,[25,75])
    IQR = Q3-Q1
    LF = Q1 -1.5*IQR
    UF = Q3 -1.5*IQR
    non_outliers_data = []
    for x in data:
        if x > LF and x <UF:
            non_outliers_data.append(x)
    return non_outliers_data,(Q1,Q3,IQR, LF,UF)

iqr_non_outliers(dataSet)

([10, 10],
 (np.float64(12.0),
  np.float64(15.0),
  np.float64(3.0),
  np.float64(7.5),
  np.float64(10.5)))

```

```
# box plot  
sns.boxplot(data)
```

<Axes: >

