In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [11]:
```
import statistics

# Creating a sample of data
arr= [115.3, 195.5, 120.5,110.2, 90.4, 105.6, 110.9, 116.3, 122.3, 125.4]

# Prints variance of the sample set

# Function will automatically calculate
# it's mean and set it as xbar
print("Variance of sample set is % s"
      %(statistics.variance(arr)))
```

```
Variance of sample set is 779.747111111111
```

In [12]:
```
import statistics

# list of positive integer numbers
arr= [115.3, 195.5, 120.5,110.2, 90.4, 105.6, 110.9, 116.3, 122.3, 125.4]

x = statistics.mean(arr)

# Printing the mean
print("Mean is :", x)
```

```
Mean is : 121.24
```

In [14]:
```
import statistics

# unsorted list of random integers
arr= [115.3, 195.5, 120.5,110.2, 90.4, 105.6, 110.9, 116.3, 122.3, 125.4]

# Printing median of the
# random data-set
print("Median of data-set is : % s "
       % (statistics.median(arr)))
```

```
Median of data-set is : 115.8
```

In [15]:
```
import statistics

# declaring a simple data-set consisting of real valued
# positive integers.
arr= [115.3, 195.5, 120.5,110.2, 90.4, 105.6, 110.9, 116.3, 122.3, 125.4]

print("Mode of given data set is % s" % (statistics.mode(arr)))
```

```
Mode of given data set is 115.3
```

In [18]:
```
import numpy as np

#initialize array
A = np.array([115.3, 195.5, 120.5,110.2, 90.4, 105.6, 110.9, 116.3, 122.3, 125.4])

#compute standard deviation
output = np.std(A)

print(output)
```

```
26.490987146574962
```

In [2]:
```
from sklearn import preprocessing
import numpy as np
numpy_array = np.array([115.3, 195.5, 120.5, 110.2, 90.4, 105.6, 110.9, 116.3, 122.3, 125.4]
normalized_array = preprocessing.normalize([numpy_array])
print("normalized data",normalized_array)
```

```
normalized data [[0.29380293 0.49816541 0.30705336 0.28080731 0.23035372 0.26908577
  0.28259102 0.29635109 0.31164005 0.31953935]]
```

In [4]:
```
import numpy as np

# Original data
original_data = [115.3, 195.5, 120.5, 110.2, 90.4, 105.6, 110.9, 116.3, 122.3, 125.4]

# Convert the data to a NumPy array
data_array = np.array(original_data)

# Calculate the mean and standard deviation of the data
mean = np.mean(data_array)
std_dev = np.std(data_array)

# Standardize the data
standardized_data = (data_array - mean) / std_dev

print("Standardized data:", standardized_data)
```

```
Standardized data: [-0.2242272   2.80321755 -0.02793403 -0.41674551 -1.16416953 -0.59038948
 -0.39032143 -0.18647852  0.04001361  0.15703454]
```

In [ ]:
```
```