Find mean, median, mode of your own dataset.

In [ ]:
```python
# Name: Atul Rajput
# Roll No. : 2230158
import pandas as pd
import numpy as np
from scipy import stats

# Create a DataFrame with 'value' and 'weight' columns
dataframe = pd.DataFrame({
    'value': [45, 67, 89, 34, 56, 78, 90, 67, 45, 23],
    'weight': [1, 3, 2, 4, 5, 2, 1, 3, 2, 4]
})

# Calculate the mean of the 'value' column
x = np.mean(dataframe.value)

# Calculate the median of the 'value' column
y = np.median(dataframe.value)

# Calculate the mode of the 'value' column
z = stats.mode(dataframe.value, keepdims=True)

# Print the mean value
print("Mean: ", x)

# Print the median value
print("Median: ", y)

# Print the mode value (along with its count)
print("Mode: ", z.mode[0], "Count: ", z.count[0])

# Function to compute the weighted average
def weighted_average(dataframe, value, weight):
    value = dataframe[value]  # Extract the value column
    weight = dataframe[weight]  # Extract the weight column
    # Compute the weighted average
    return (value * weight).sum() / weight.sum()

# Calculate and print the weighted average
print("Weighted Average: ", weighted_average(dataframe, 'value', 'weight'))

# Function to compute the weighted median
def weighted_median(dataframe, value, weight):
    sorted_df = dataframe.sort_values(by=value)  # Sort by value
```

```
    cumulative_weight = sorted_df[weight].cumsum()  # Cumulative sum of weights
    cutoff = sorted_df[weight].sum() / 2.0  # Half of total weight
    # Find the value where cumulative weight is just greater than or equal to the cutoff
    return sorted_df[sorted_df[weight].cumsum() >= cutoff][value].iloc[0]

# Calculate and print the weighted median
print("Weighted Median: ", weighted_median(dataframe, 'value', 'weight'))
```

```
Mean:  59.4
Median:  61.5
Mode:  45 Count:  2
Weighted Average:  54.407407407407405
Weighted Median:  56
```