1. Compute Central Tendency Measures: Mean, Median, Mode Measure of Dispersion: Variance, Standard Deviation.

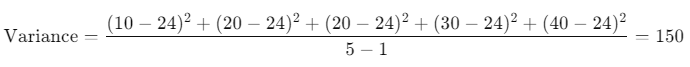
Description:

 **Mean**: The arithmetic average of all numbers.



 **Median**: The middle value in the sorted dataset.  
 Sorted dataset: [10, 20, 20, 30, 40]. The median is 20.

 **Mode**: The value that occurs most frequently.  
 Mode = 20.

 **Variance**: Measures how far the data points are spread from the mean. 

 **Standard Deviation**: The square root of the variance.



**Python Code:**

data = [10, 20, 20, 30, 40]

def calculate\_mean(data):

return sum(data) / len(data)

def calculate\_median(data):

sorted\_data = sorted(data)

n = len(sorted\_data)

mid = n // 2

if n % 2 == 0:

return (sorted\_data[mid - 1] + sorted\_data[mid]) / 2

else:

return sorted\_data[mid]

def calculate\_mode(data):

frequency = {}

for num in data:

frequency[num] = frequency.get(num, 0) + 1

max\_freq = max(frequency.values())

modes = [key for key, val in frequency.items() if val == max\_freq]

return modes if len(modes) > 1 else modes[0]

def calculate\_variance(data, mean):

squared\_diff = [(x - mean) \*\* 2 for x in data]

return sum(squared\_diff) / (len(data) - 1) # Sample variance (N-1)

def calculate\_standard\_deviation(variance):

return variance \*\* 0.5

mean = calculate\_mean(data)

median = calculate\_median(data)

mode = calculate\_mode(data)

variance = calculate\_variance(data, mean)

std\_dev = calculate\_standard\_deviation(variance)

print(f"Mean: {mean}")

print(f"Median: {median}")

print(f"Mode: {mode}")

print(f"Variance: {variance}")

print(f"Standard Deviation: {std\_dev}")

**Output:**

For the dataset [10, 20, 20, 30, 40], the output will be:

Mean: 24.0

Median: 20

Mode: 20

Variance: 150.0

Standard Deviation: 12.24744871391589