1. Data Analysis:

test\_data\_analysis.py

from data\_analysis import calculate\_mean,calculate\_median,calculate\_standard\_deviation,calculate\_variance,calculate\_range

# test\_data = []

# test\_data = [2, 4, 6, 8, 10, 12]

test\_data = [3, 7, 11, 15, 19]

def main():

    print(f"The dataset is: {test\_data}")

    print(f"Mean: {calculate\_mean(test\_data)}")

    print(f"Median: {calculate\_median(test\_data)}")

    print(f"SD: {calculate\_standard\_deviation(test\_data)}")

    print(f"Variance: {calculate\_variance(test\_data)}")

    print(f"Variance: {calculate\_range(test\_data)}")

if \_\_name\_\_=="\_\_main\_\_":

    main()

data\_analysis.py

# a.       calculate\_mean(data): Accepts a list of numerical data and returns the mean (average) value.

# b.      calculate\_median(data): Accepts a list of numerical data and returns the median value.

# c.       calculate\_variance(data): Accepts a list of numerical data and returns the variance.

# d.      calculate\_standard\_deviation(data): Accepts a list of numerical data and returns the standard deviation.

from functools import reduce

# Mean:

def calculate\_mean(data):

    n = len(data)

    sum =  reduce(lambda x,y: x+y,data)

    mean = sum/n

    return mean

# Median:

def calculate\_median(data):

    n = len(data)

    d = list(data)

    d.sort()

    mid= n//2

    if n%2==0:

        median= (d[mid-1]+d[mid])/2

    else :

        median = d[mid]

    return median

# SD:

def calculate\_standard\_deviation(data):

    n=len(data)

    mean = calculate\_mean(data)

    squared\_deviation = reduce(lambda x,y: x + (y - mean)\*\*2,data,0)

    standard\_deviation = (squared\_deviation/n)\*\*0.5

    return standard\_deviation

# Variance:

def calculate\_variance(data):

    standard\_deviation = calculate\_standard\_deviation(data)

    variance = standard\_deviation \*\* 2

    return variance

# Range:

def calculate\_range(data):

    return max(data) - min(data)

Additional functionality: **Range**

Outputs:  


