~\OneDrive\Desktop\Statistics Study Materials\M.Sc. Statistics\Semester - II\Python Practicals\summary_basic.py

```python
# Practical 3 : Problem - 4(a) & 4(b)
# (a) Creating a Python module named : summary_basic.py
#(b) Move the mean(data) function into this module.

# This module contains basic statistical functions

def mean(data):
    """
    Calculates the arithmetic mean of a dataset.
    """
    total = 0
    count = 0

    for value in data:
        total += value
        count += 1

    return total / count


# Practical 3 : Problem - 5(a), 5(b), 5(c) & 5(d)

# (a) 'median(data)' – returns the median of the dataset.
def median(data):
    """
    Calculates the median of a dataset.
    """
    sorted_data = sorted(data)
    n = len(sorted_data)
    mid = n // 2

    # If even number of elements
    if n % 2 == 0:
        return (sorted_data[mid - 1] + sorted_data[mid]) / 2
    else:
        return sorted_data[mid]


# (b) 'mode(data)' – returns the mode of the dataset. If more than one mode exists, return any one of them.
def mode(data):
    """
    Calculates the mode of a dataset.
    """
    frequency = {}

    # Count frequency of each element
    for value in data:
        frequency[value] = frequency.get(value, 0) + 1

    # Find the value with maximum frequency
    max_frequency = max(frequency.values())

    for key in frequency:
        if frequency[key] == max_frequency:
            return key


# (c) 'std_dev(data)' – returns the sample standard deviation of the dataset.
def std_dev(data):
    """
    Calculates the sample standard deviation of a dataset.
    """
    m = mean(data)
    total = 0
    n = len(data)

    for value in data:
        total += (value - m) ** 2

    variance = total / (n - 1)   # Sample variance
    return variance ** 0.5

# (d) Each function should :
#      * Work only on cleaned numeric data
#      * Not rely on external statistical libraries
```