**Problem - 4**

**Student Name: Milan Sharma UID: 23MAI10003**

**Branch: ME – CSE - AIML Section/Group: MAI – 1 (A)**

**Semester: 1st Date of Performance: 23 Oct 2023**

**Subject Name: Artificial Intelligence Subject Code: 23 CSH 621**

1. **Aim of the Experiment :**

You are given an array of N integers separated by spaces, all in one line.

Display the following:

Mean (m): The average of all the integers.

Median of this array: In case, the number of integers is odd, the middle element; else, the average of the middle two elements.

Mode: The element(s) which occurs most frequently. If multiple elements satisfy this criteria, display the numerically smallest one.

Standard Deviation (SD).

SD = (((x1-m)2+(x2-m)2+(x3-m)2+(x4-m)2+...(xN-m)2))/N)0.5

where xi is the ith element of the array

Lower and Upper Boundary of the 95% Confidence Interval for the mean, separated by a space. This might be a new term to some. However, it is an important concept with a simple, formulaic solution. Look it up!

1. **Objective of the Experiment :**

The modal values (which should all be integers) the answers should be in decimal form till one place of decimal (0.0 format). An error margin of +/- 0.1 will be tolerated for the standard deviation and the confidence interval boundaries. The mean, mode and median values should match the expected answers exactly.

Assume that these numbers were sampled from a normal distribution; the sample is a reasonable representation of the distribution; a user can approximate that the population standard deviation =~ standard deviation computed for the given points- with the understanding that assumptions of normality are convenient approximations.

1. **Code :**

import sys

from collections import defaultdict

for line in sys.stdin:

    nums = line.split(" ")

mean = 0

mode = 0

median = 0

deviation = 0

lowerconfidence = 0

upperconfidence = 0

d = defaultdict(int)

for n in nums:

    mean+=int(n)

    d[n] += 1

n = len(nums)

mean = mean / n

nums = [int(x) for x in nums]

median = (sorted(nums)[n//2]) if n % 2 else (sorted(nums)[n//2] + sorted(nums)[n//2 - 1])/ 2

modeCandidates = []

freq = max(d.values())

for k in d:

    if freq == d[k]:

        modeCandidates.append(int(k))

if len(k) == 1:

    mode = k

else:

    mode = sorted(modeCandidates)[0]

deviation = [(x - mean) \*\* 2  for x in nums]

deviation = (sum(deviation) / n)  \*\* .5

lowerconfidence = mean - 1.96 \* (deviation) / (len(nums) \*\*.5)

upperconfidence = mean + 1.96 \* (deviation) / (len(nums) \*\*.5)

print('{:.1f}'.format(mean))

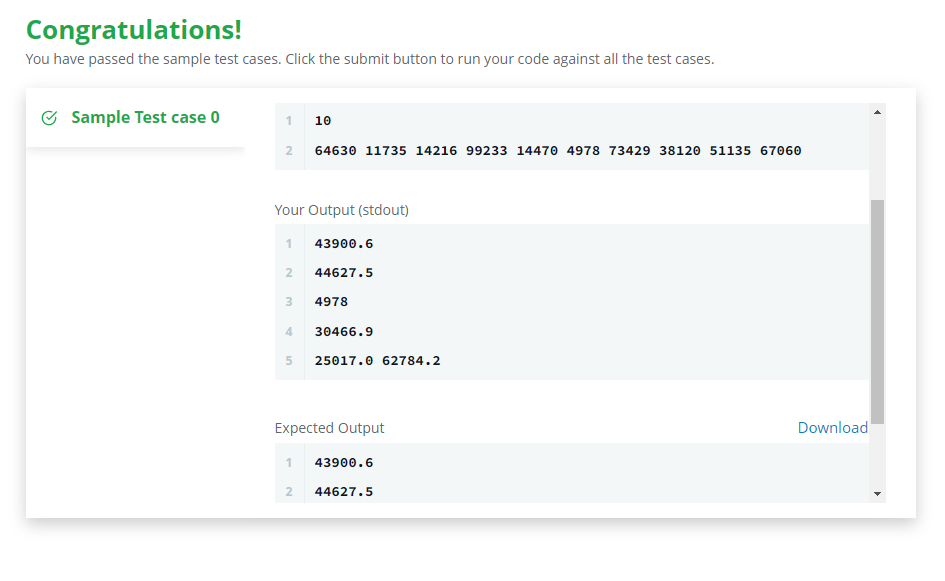
print('{:.1f}'.format(median))

print('{:.0f}'.format(mode))

print('{:.1f}'.format(deviation))

print(('{lowerconfidence:.1f} {upperconfidence:.1f}').format(lowerconfidence = lowerconfidence, upperconfidence = upperconfidence))

1. **Output:**

****

**Learning outcomes (What I have learnt):**

1. **Learned about numpy and pandas libraries.**
2. **Learned to use time constraints in python.**
3. **Learned how to use mean mode median in python using libraries.**