**E**stimation **A**nd **C**onfidence **I**ntervals

**Background**

In quality control processes, especially when dealing with high-value items, destructive sampling is a necessary but costly method to ensure product quality. The test to determine whether an item meets the quality standards destroys the item, leading to the requirement of small sample sizes due to cost constraints.

**Scenario**

A manufacturer of print-heads for personal computers is interested in estimating the mean durability of their print-heads in terms of the number of characters printed before failure. To assess this, the manufacturer conducts a study on a small sample of print-heads due to the destructive nature of the testing process.

**Data**

A total of 15 print-heads were randomly selected and tested until failure. The durability of each print-head (in millions of characters) was recorded as follows:

1.13, 1.55, 1.43, 0.92, 1.25, 1.36, 1.32, 0.85, 1.07, 1.48, 1.20, 1.33, 1.18, 1.22, 1.29

**Assignment Tasks**

**a. Build 99% Confidence Interval Using Sample Standard Deviation**

Assuming the sample is representative of the population, construct a 99% confidence interval for the mean number of characters printed before the print-head fails using the sample standard deviation. Explain the steps you take and the rationale behind using the t-distribution for this task.

**b. Build 99% Confidence Interval Using Known Population Standard Deviation**

If it were known that the population standard deviation is 0.2 million characters, construct a 99% confidence interval for the mean number of characters printed before failure.

Ans=

Step 1= import the libraries required like numpy, pandas and stats from scipy

Step 2=Converted the data into series from using pd.series() function

Step3= calculated mean and standard deviation of given data using numpy

Step4=then use inbuilt python function for interval calculation ie.stats.norm.interrval() after passing value according to a and b requirements we get required interval.

Code:

import pandas as pd

import numpy as np

from scipy import stats

data=pd.Series([1.13, 1.55, 1.43, 0.92, 1.25, 1.36, 1.32, 0.85, 1.07, 1.48, 1.20, 1.33, 1.18, 1.22, 1.29])

data

mean=np.mean(data)

stdev=np.std(data)

print(mean,stdev)

sample\_confidence\_int=stats.norm.interval(0.99,mean,stdev)

print("sample",sample\_confidence\_int)

pop\_confidence\_int=stats.norm.interval(0.99,mean,0.2)

print("population",pop\_confidence\_int)

Code screenshots:



