**ESTIMATION AND CONFIDENCE INTERVALS**

**(GIVEN QUESTIONS AND PROVIDED SOLUTIONS)**

**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.

***SOLUTION:***

***The below are the steps and the rationale behind using the t-distribution:***

##importing required libraries

##importing pandas

##importing scipy

##import numpy

import pandas as pd

import numpy as np

from scipy import stats

*###Passing the given data as Series*

*###question 1 using series*

Method 1:

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

###task a

###building 99% confidence interval using sample standard deviation

mean = data.mean()

std\_error = data.sem() # Standard deviation

print(“Standard deviation is :” , std\_error)

# Step 4: Determining degrees of freedom

degrees\_of\_freedom = len(data) - 1

# Step 5: confidence level

confidence\_level = 0.99

# Step 6: Finding the critical value (t-score)

t\_score = stats.t.ppf((1 + confidence\_level) / 2, degrees\_of\_freedom)

# Step 7: Calculating the value of margin of error

margin\_of\_error = t\_score \* std\_error

# Step 8: Computing the confidence interval

lower\_bound = mean - margin\_of\_error

upper\_bound = mean + margin\_of\_error

print("Confidence Interval:", (lower\_bound, upper\_bound))

***RESULTS OBTAINED:***

Standard deviation is : 0.04987476379384733

Confidence Interval: (1.090197338451367, 1.3871359948819662)

***Method 2:***

***####considering the data as an array***

***#####array#####***

data1 = np.array([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])

data1

import pandas as pd

import numpy as np

from scipy import stats

##question a solving using array

# Step 1: Calculating the values that is necessary statistics

mean1 = data1.mean()

std\_error1=stats.sem(data1) #standard deviation

# Step 2: Setting the confidence level

confidence\_level1 = 0.99

# Step 3: Value of degrees of freedom

degrees\_of\_freedom1 = len(data1) - 1

# Step 4: critical value (t-score)

t\_score1 =stats.t.ppf ((1 + confidence\_level1) / 2, degrees\_of\_freedom1)

# Step 5: Calculate margin of error

margin\_of\_error1 = t\_score1 \* std\_error1

# Step 6: confidence interval

lower\_bound1 = mean1 - margin\_of\_error1

upper\_bound1 = mean1 + margin\_of\_error1

print("99% Confidence Interval using known Sample Standard Deviation:")

print("Lower Bound:", lower\_bound1)

print("Upper Bound:", upper\_bound1)

***RESULTS:***

99% Confidence Interval using known Sample Standard Deviation:

Lower Bound: 1.090197338451367

Upper Bound: 1.3871359948819662

***QUESTION 2:***

***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.***

###task b

building 99% confidence interval using known population deviation

import numpy as np

# data given and considered

data2 = np.array([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])

# Given that the Known population standard deviation is 0.2 in (million characters)

population\_std\_dev2 = 0.2 # in million characters

# Step 1: Required statistics

mean2 = np.mean(data2)

# Step 2: confidence level

confidence\_level2 = 0.99

# Step 3: critical value: that is (z-score)

z\_score2 = 2.576 # For 99% confidence level

# Step 4: margin of error

margin\_of\_error2 = z\_score2 \* (population\_std\_dev2 / np.sqrt(len(data2)))

# Step 5: Final value that is confidence interval

lower\_bound2 = mean2 - margin\_of\_error2

upper\_bound2 = mean2 + margin\_of\_error2

print ("99% Confidence Interval using Known Population Standard Deviation")

print ("Lower Bound:", lower\_bound2)

print ("Upper Bound:", upper\_bound2)

***RESULTS:***

99% Confidence Interval using Known Population Standard Deviation

Lower Bound: 1.1056426

Upper Bound: 1.3716907

***CONCLUSION:***

Thus, the above shown are the steps followed to calculate the confidence intervals

For 2 cases:

1. T-score distribution using known sample standard deviation.
2. Z-score distribution using known population standard deviation.

A,Therefore,

Confidence Interval for the mean number of characters printed before for the print head-tails lies between (1.090,1.388). So that we can say that we are 99 % confident that the actual population mean lies in the obtained interval .

B,Therefore,

If it were known the population standard deviation then the confidence interval is (1.106,1.372). Hence we can say that 99% confidence interval for the mean number of characters printed before the failure is (1.106,1.372) with the known population standard deviation.