22.Scenario:

Imagine you are an analyst for a popular online shopping website. Your task is to analyze customer

reviews and provide insights on the average rating and customer satisfaction level for a specific

product category.

Question:

You will use the pandas library to calculate confidence intervals to estimate the true population

mean rating.

You have been provided with a CSV file named "customer\_reviews.csv," which contains customer

ratings for products in the chosen category.

Code :

#22

import pandas as pd

import numpy as np

import scipy.stats as stats

# Function to calculate confidence interval for the mean rating

def calculate\_confidence\_interval(data, confidence\_level=0.95):

# Calculate sample mean and sample standard deviation

sample\_mean = np.mean(data)

sample\_std = np.std(data, ddof=1) # Sample standard deviation

# Sample size

sample\_size = len(data)

# Calculate the standard error

standard\_error = sample\_std / np.sqrt(sample\_size)

# Calculate the Z-score or t-score (t-score for small sample sizes, z-score for large)

alpha = 1 - confidence\_level

t\_score = stats.t.ppf(1 - alpha/2, df=sample\_size-1) # t-score for given confidence level

# Calculate margin of error

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

# Calculate confidence interval

lower\_bound = sample\_mean - margin\_of\_error

upper\_bound = sample\_mean + margin\_of\_error

return sample\_mean, lower\_bound, upper\_bound

# Main function

def main():

# Input the Excel file

file\_path = r"C:\Users\hares\Downloads\q22\_05.xlsx" # Change this to your actual Excel file path

# Load data from Excel

df = pd.read\_excel(file\_path)

# Assuming the column containing ratings is named 'Rating'

if 'Rating' not in df.columns:

print("Error: 'Rating' column not found in the dataset.")

return

# Get ratings data (drop NaN values if any)

ratings = df['Rating'].dropna()

# Calculate confidence interval

confidence\_level = 0.95 # Default 95% confidence level

sample\_mean, lower\_bound, upper\_bound = calculate\_confidence\_interval(ratings, confidence\_level)

# Output results

print(f"Sample Mean Rating: {sample\_mean:.4f}")

print(f"{confidence\_level\*100}% Confidence Interval: ({lower\_bound:.4f}, {upper\_bound:.4f})")

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

main()

output :

Sample Mean Rating: 4.2100

95.0% Confidence Interval: (3.9230, 4.4970)

Dataset :

|  |  |  |
| --- | --- | --- |
| **Product\_ID** | **Customer\_ID** | **Rating** |
| 101 | 1 | 4.5 |
| 101 | 2 | 3.9 |
| 101 | 3 | 4.2 |
| 101 | 4 | 5 |
| 101 | 5 | 4.3 |
| 102 | 6 | 3.8 |
| 102 | 7 | 4.1 |
| 102 | 8 | 4.6 |
| 102 | 9 | 4 |
| 102 | 10 | 3.7 |