import numpy as np # Import NumPy library for numerical operations

from scipy import stats # Import stats module from SciPy for statistical functions

# Data: durability of print-heads in million characters

durability = 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])

n = len(durability) # Number of observations

mean = np.mean(durability) # Sample mean of durability

s = np.std(durability, ddof=1) # Sample standard deviation (ddof=1 for unbiased estimate)

# 99% Confidence Interval using sample standard deviation (t-distribution)

alpha = 0.01 # Significance level for 99% confidence

df = n - 1 # Degrees of freedom

t\_crit = stats.t.ppf(1 - alpha/2, df) # Critical t-value

se = s / np.sqrt(n) # Standard error of the mean

margin\_t = t\_crit \* se # Margin of error

ci\_t = (mean - margin\_t, mean + margin\_t) # Confidence interval using t-distribution

# 99% Confidence Interval using known population standard deviation (z-distribution)

sigma = 0.2 # Known population standard deviation

z\_crit = stats.norm.ppf(1 - alpha/2) # Critical z-value

se\_known = sigma / np.sqrt(n) # Standard error using known population SD

margin\_z = z\_crit \* se\_known # Margin of error

ci\_z = (mean - margin\_z, mean + margin\_z) # Confidence interval using z-distribution

# Print results

print(f"Sample mean: {mean:.4f} million characters")

print(f"99% CI using sample SD (t): {ci\_t[0]:.3f} to {ci\_t[1]:.3f} million characters")

print(f"99% CI using known population SD (z): {ci\_z[0]:.3f} to {ci\_z[1]:.3f} million characters")

OUTPUT:

Sample mean: 1.2387 million characters

99% CI using sample SD (t): 1.090 to 1.387 million characters

99% CI using known population SD (z): 1.106 to 1.372 million characters