basic-stats-2

November 20, 2024

[1]:
```
import numpy as np
from scipy.stats import t, norm

# Data
data = [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(data)
confidence_level = 0.99

# Part a: Confidence Interval Using Sample Standard Deviation
# Sample mean and standard deviation
sample_mean = np.mean(data)
sample_std = np.std(data, ddof=1)  # ddof=1 for sample standard deviation

# t-critical value
degrees_of_freedom = n - 1
t_critical = t.ppf((1 + confidence_level) / 2, df=degrees_of_freedom)

# Margin of Error
margin_of_error_a = t_critical * (sample_std / np.sqrt(n))

# Confidence Interval
ci_a = (sample_mean - margin_of_error_a, sample_mean + margin_of_error_a)

# Part b: Confidence Interval Using Known Population Standard Deviation
# Known population standard deviation
population_std = 0.2

# z-critical value
z_critical = norm.ppf((1 + confidence_level) / 2)

# Margin of Error
margin_of_error_b = z_critical * (population_std / np.sqrt(n))

# Confidence Interval
ci_b = (sample_mean - margin_of_error_b, sample_mean + margin_of_error_b)
```

```
# Results
print("Part a: 99% Confidence Interval using sample standard deviation:", ci_a)
print("Part b: 99% Confidence Interval using known population standard␣
  ↪deviation:", ci_b)
```

```
Part a: 99% Confidence Interval using sample standard deviation:
(1.0901973384384906, 1.3871359948948425)
Part b: 99% Confidence Interval using known population standard deviation:
(1.1056514133957607, 1.3716819199375725)
```

[ ]: