

---

Name : Devansh Parapalli  
Roll Number: 35  
Semester: 6 | Year: 3  
Branch : CSE

## Problem 01

A manufacturer wants to determine the inner diameter of a chalk. Ideally, the diameter should be 57 mm. The data is as follows: 72, 57, 53, 58, 56, 57, 56, 50. Calculate the measures of central value and dispersion, and comment on the quality of the chalk produced.

```
import statistics
from typing import List

def calculate_central_values(data: List[float]) -> tuple:
    mean = sum(data) / len(data)
    median = statistics.median(data)
    try:
        mode = statistics.mode(data)
    except statistics.StatisticsError:
        mode = "No mode found (data has no repeating values)"

    return mean, median, mode

def calculate_dispersion(data: List[float]) -> tuple:
    data_range = max(data) - min(data)
    variance = statistics.variance(data)
    std_dev = statistics.stdev(data)

    return data_range, variance, std_dev

def main():
    chalk_diameters = [72, 57, 53, 58, 56, 57, 56, 50]

    ideal_diameter = 57

    mean, median, mode = calculate_central_values(chalk_diameters)
    print(f"Mean diameter: {mean:.2f} mm")
    print(f"Median diameter: {median:.2f} mm")
    print(f"Mode diameter: {mode}")

    data_range, variance, std_dev = calculate_dispersion(chalk_diameters)
    print(f"Range: {data_range:.2f} mm")
    print(f"Variance: {variance:.2f}")
    print(f"Standard deviation: {std_dev:.2f} mm")

    if mean == ideal_diameter and std_dev == 0:
        print("The chalk produced is of excellent quality, with all diameters equal to the ideal diameter")
    elif abs(mean - ideal_diameter) <= std_dev:
        print("The chalk produced is of acceptable quality, with the mean diameter within one standard deviation")
    else:
        print("The chalk produced is of poor quality, with the mean diameter deviating significantly from the ideal")

if __name__ == "__main__":
    main()
```

## Output

```
PS C:\DevParapalli\Projects\RTMNU-SEM-6> & "C:/Program Files/Python310/python.exe" c:/DevParapalli/Project01/problem01.py
Mean diameter: 57.38 mm
Median diameter: 56.50 mm
Mode diameter: 57
Range: 22.00 mm
Variance: 41.70
Standard deviation: 6.46 mm
The chalk produced is of acceptable quality, with the mean diameter within one standard deviation of the ideal diameter
```

---

## Problem 02

About 1000 college students are graded according to their IQ level and economic condition. Use the chi-square test to find out whether there is any association between economic condition and the level of IQ.

|      | High | Medium | Low |
|------|------|--------|-----|
| Rich | 160  | 300    | 140 |
| Poor | 140  | 100    | 160 |

```
import numpy as np
from scipy.stats import chi2_contingency

# Observed data
observed = np.array([[160, 300, 140],
                     [140, 100, 160]])

# Perform the chi-square test
chi2_stat, p_val, dof, expected = chi2_contingency(observed)

# Print the results
print("Chi-square statistic: ", chi2_stat)
print("p-value: ", p_val)
print("Degrees of freedom: ", dof)

# Interpret the results
alpha = 0.05 # Significance level
if p_val < alpha:
    print("There is a significant association between economic condition and IQ level.")
else:
    print("There is no significant association between economic condition and IQ level.")
```

### Output

```
& "C:/Program Files/Python310/python.exe" c:/DevParapalli/Projects/RTMNU-SEM-6/DS/b.py
PS C:\DevParapalli\Projects\RTMNU-SEM-6> & "C:/Program Files/Python310/python.exe" c:/DevParapalli/Proje
Chi-square statistic: 65.27777777777777
p-value: 6.6851423617186904e-15
Degrees of freedom: 2
There is a significant association between economic condition and IQ level.
```

---

## Problem 03

An IQ test was administered to 5 persons before and after they were trained. IQ before training was 110, 120, 123, 132, 125 of 5 persons. After training IQ became 120, 118, 125, 136, 121. Test whether there is any change in IQ after training. Given t value at 1% level significance is 4.6

```
import scipy.stats as stats
import numpy as np

# IQ scores before training
iq_before = np.array([110, 120, 123, 132, 125])

# IQ scores after training
iq_after = np.array([120, 118, 125, 136, 121])

# Calculate the differences between paired observations
diff = iq_after - iq_before

# Calculate the test statistic and p-value
t_statistic, p_value = stats.ttest_rel(iq_after, iq_before)

# Print the results
print(f"t-statistic: {t_statistic:.4f}")
print(f"p-value: {p_value:.4f}")

# Check if the p-value is less than the significance level (0.01 for 1% level)
alpha = 0.01
if p_value < alpha:
    print("There is a significant change in IQ after training.")
else:
    print("There is no significant change in IQ after training.")
```

## Output

```
& "C:/Program Files/Python310/python.exe" c:/DevParapalli/Projects/RTMNU-SEM-6/DS/c.py
PS C:\DevParapalli\Projects\RTMNU-SEM-6> & "C:/Program Files/Python310/python.exe" c:/DevParapalli/Projects/RTMNU-SEM-6/DS/c.py
t-statistic: 0.8165
p-value: 0.4601
There is no significant change in IQ after training.
```