# ML Lab 7: t-Test

#### Submitted by

Debatreya Das Roll No. 12212070 CS A4 ML Lab 7

### Import necessary Libraries

import pandas as pd  
import numpy as np

### Load Dataset

data = pd.read\_csv('./sample.csv')  
  
data

Participant Group Score  
0 1 A 56  
1 2 A 67  
2 3 A 45  
3 4 A 59  
4 5 A 63  
5 6 A 70  
6 7 A 54  
7 8 A 60  
8 9 A 58  
9 10 A 62  
10 11 A 65  
11 12 A 61  
12 13 A 64  
13 14 A 57  
14 15 A 68  
15 16 A 55  
16 17 A 69  
17 18 A 59  
18 19 A 66  
19 20 A 71  
20 21 B 68  
21 22 B 72  
22 23 B 61  
23 24 B 69  
24 25 B 75  
25 26 B 80  
26 27 B 64  
27 28 B 70  
28 29 B 66  
29 30 B 73  
30 31 B 77  
31 32 B 62  
32 33 B 74  
33 34 B 78  
34 35 B 65  
35 36 B 67  
36 37 B 71  
37 38 B 76  
38 39 B 63  
39 40 B 79

### Separate data into Group A and Group B

group\_a = data[data['Group'] == 'A']['Score'].values  
group\_b = data[data['Group'] == 'B']['Score'].values  
  
print("Group A", group\_a)  
print("Group B", group\_b)

Group A [56 67 45 59 63 70 54 60 58 62 65 61 64 57 68 55 69 59 66 71]  
Group B [68 72 61 69 75 80 64 70 66 73 77 62 74 78 65 67 71 76 63 79]

## t-Test Function Implementation

def calculate\_t\_statistic(group\_a, group\_b):  
 # Number of samples in each group  
 n1 = len(group\_a)  
 n2 = len(group\_b)  
   
 # Mean of each group  
 mean\_a = np.mean(group\_a)  
 mean\_b = np.mean(group\_b)  
   
 # Variance of each group  
 var\_a = np.var(group\_a, ddof=1) # Sample variance (ddof=1)  
 var\_b = np.var(group\_b, ddof=1)  
   
 # T-statistic formula  
 t\_stat = (mean\_a - mean\_b) / np.sqrt((var\_a / n1) + (var\_b / n2))  
   
 return t\_stat

### Calculating t-statistic for the given data

t\_statistic = calculate\_t\_statistic(group\_a, group\_b)  
print(f"T-statistic: {t\_statistic}")

T-statistic: -4.639420527316076

## Implementing the function to calculate the degrees of freedom

def calculate\_degrees\_of\_freedom(group\_a, group\_b):  
 n1 = len(group\_a)  
 n2 = len(group\_b)  
   
 var\_a = np.var(group\_a, ddof=1)  
 var\_b = np.var(group\_b, ddof=1)  
   
 numerator = ((var\_a / n1) + (var\_b / n2)) \*\* 2  
 denominator = ((var\_a \*\* 2) / ((n1 \*\* 2) \* (n1 - 1))) + ((var\_b \*\* 2) / ((n2 \*\* 2) \* (n2 - 1)))  
   
 df = numerator / denominator  
 return df

### Calculating the degress of freedom for the data

# Calculate the degrees of freedom  
df = calculate\_degrees\_of\_freedom(group\_a, group\_b)  
print(f"Degrees of freedom: {df}")

Degrees of freedom: 37.757207747519985

## Look Up Critical Value for T-distribution (5% level of significance)

from scipy.stats import t  
  
# Set significance level (alpha) for two-tailed test  
alpha = 0.05  
  
# Calculate the critical t-value  
critical\_value = t.ppf(1 - alpha / 2, df)  
print(f"Critical t-value: {critical\_value}")

Critical t-value: 2.0248217426546193

## Interpret the result

# Compare t-statistic with critical value  
if abs(t\_statistic) > critical\_value:  
 print("Reject the null hypothesis - there is a significant difference between the two groups.")  
else:  
 print("Accept the null hypothesis - there is no significant difference between the two groups.")

Reject the null hypothesis - there is a significant difference between the two groups.