EX.NO:5	Conducting hypothesis testing using t-test and chi-square test
DATE:	

AIM:
To perform a
- two-sample T-Test (Student's or Welch's) using either raw or summary data
- Chi-Square Test (for categorical data)

ALGORITHM:
T-Test:
Step 1: Prompt user to choose between raw or summary data.
Step 2: Ask if equal variances should be assumed.
Step 3: Collect sample sizes and data (either raw or summary).
Step 4: Check if both sample sizes are ≤ 30. If not, stop the program.
Step 5: Calculate t-statistic, p-value, and degrees of freedom (df).
Step 6: Compare p-value with significance level ($\alpha = 0.05$).
Step 7: Output whether the null hypothesis (no mean difference) is rejected or not
Chi-Square Test:
Step 1: Input observed data in matrix form.
Step 2: Use scipy.stats.chi2_contingency().
Step 3: Calculate chi-square statistic, p-value, degrees of freedom.
Step 4: Interpret results using significance level.

PROGRAM:
- T-Test Using Python:

```
from scipy import stats
import math

def t_test_from_summary(mean1, std1, n1, mean2, std2, n2, equal_var):
    if equal_var:
        var = (((n1 - 1) * std1**2 + (n2 - 1) * std2**2) / (n1 + n2 - 2))
        se = math.sqrt(var * (1/n1 + 1/n2))
        df = n1 + n2 - 2
    else:
        se = math.sqrt((std1**2 / n1) + (std2**2 / n2))
        df_n = (std1**2 / n1 + std2**2 / n2)**2
        df_d = ((std1**2 / n1)**2 / (n1 - 1)) + ((std2**2 / n2)**2 / (n2 - 1))
        df = df_n/ df_d

    t_stat = (mean1 - mean2) / se
    p_value = 2 * (1 - stats.t.cdf(abs(t_stat), df))
    return t_stat, p_value, df
```

```
def main():
    print("T-Test: Compare Means of Two Independent Groups (n ≤ 30)")
    print("Choose input type:")
    print("1. Raw data")
    print("2. Summary data")
    choice = input("Enter 1 or 2: ").strip()

    equal_var_input = input("\nAssume equal variances? (yes/no): ").strip().lower()
    equal_var = True if equal_var_input == "yes" else False

    if choice == '1':
        n1 = int(input("\nEnter number of elements in Group 1 (≤ 30): "))
        if n1 > 30:
            print("Sample size for Group 1 exceeds 30. Program terminates.")
            return
        print("Enter values for Group 1:")
        group1 = [float(input(f"  Value {i+1}: ")) for i in range(n1)]

        n2 = int(input("\nEnter number of elements in Group 2 (≤ 30): "))
        if n2 > 30:
            print(" Sample size for Group 2 exceeds 30. Program terminates.")
            return
        print("Enter values for Group 2:")
        group2 = [float(input(f"  Value {i+1}: ")) for i in range(n2)]

        t_stat, p_value = stats.ttest_ind(group1, group2, equal_var=equal_var)
        print("\n--- T-Test Results (Raw Data) ---")
        print("T-statistic:", round(t_stat, 4))
        print("P-value:", round(p_value, 4))

    elif choice == '2':
        print("\nEnter summary data for Group 1:")
        n1 = int(input("Sample size (n1 ≤ 30): "))
        if n1 > 30:
            print("Sample size for Group 1 exceeds 30. Program terminates.")
            return
        mean1 = float(input("Mean (x̄1): "))
        std1 = float(input("Standard deviation (s1): "))

        print("\nEnter summary data for Group 2:")
        n2 = int(input("Sample size (n2 ≤ 30): "))
        if n2 > 30:
            print("Sample size for Group 2 exceeds 30. Program terminates.")
            return
        mean2 = float(input("Mean (x̄2): "))
        std2 = float(input("Standard deviation (s2): "))
```

```
        t_stat, p_value, df = t_test_from_summary(mean1, std1, n1, mean2, std2,
n2, equal_var)
        print("\n--- T-Test Results (Summary Data) ---")
        print("T-statistic:", round(t_stat, 4))
        print("P-value:", round(p_value, 4))
        print("Degrees of freedom:", round(df, 2))

    else:
        print("Invalid choice. Please enter 1 or 2.")
        return

    # Final decision based on p-value
    alpha = 0.05
    print("\n--- Hypothesis Test Decision ---")
    if p_value < alpha:
        print("Reject H0 : Significant difference between group means at 5% level of
significance.")
    else:
        print("Accept Ho : No significant difference between group means at 5%
level of significance..")

# Run the main function
if __name__ == "__main__":
    main()
```

- Chi-Square Test Using Python:

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

rows = int(input("Enter number of rows: "))
cols = int(input("Enter number of columns: "))
observed = []

for i in range(rows):
    row = [int(input(f"Value at ({i+1},{j+1}): ")) for j in range(cols)]
    observed.append(row)

observed_array = np.array(observed)
chi2, p, dof, expected = chi2_contingency(observed_array)

print("Chi-Square Value:", chi2)
print("P-value:", p)
print("Degrees of Freedom:", dof)
```

```
print("Expected Frequencies:\n", expected)

if p < 0.05:
    print("Reject H0: Variables are dependent.")
else:
    print("Accept H0: Variables are independent.")
```

RESULT:

The hypothesis testing was successfully conducted using Python for:

- two-sample T-Test (Student's or Welch's) using either raw or summary data
- Chi-Square Test (for categorical data)

EXERCISE:

1. Generate a Python program to determine whether the mean of a small sample (with size ≤ 30) differs significantly from a known population mean (μ) using a one-sample t-test, given either raw data or summary statistics. Check the scenarios below:
   Scenario 1: A nutritionist wants to test if a new diet plan changes the average daily calorie intake of adults compared to the known average of 2500 calories. She collects data from 20 adults following the new diet and records their daily calorie intakes as follows (in calories): 2480, 2525, 2490, 2455, 2510, 2470, 2505, 2485, 2495, 2515, 2460, 2520, 2488, 2475, 2498, 2502, 2465, 2518, 2483, 2490. Using a significance level of 0.05, test if the diet changes the average daily calorie intake.
   Scenario 2: A manufacturer claims that the average weight of their protein bars is 60 grams. An independent lab takes a random sample of 25 bars and finds the average weight to be 58.7 grams with a standard deviation of 1.8 grams. Using a significance level of 0.01, test whether the actual average weight differs from the claimed 60 grams.
2. Extend the chi-square program to handle the goodness-of-fit test. Check the scenarios below:
   Scenario 1: A candy manufacturer claims that their candies come in four colours in equal proportions: red, green, blue, and yellow. You want to test if a sample of 100 candies matches this claim. Data collected as Red: 28 candies; Green: 24 candies; Blue: 30 candies; Yellow: 18 candies. Using a significance level of 0.05, test if the observed colour distribution fits the expected equal distribution.