#####Mean by Class

import pandas as pd

# Sample data

data = {

    'Class': ['A', 'A', 'A', 'B', 'B', 'C', 'C'],

    'Student': ['S1', 'S2', 'S3', 'S4', 'S5', 'S6', 'S7'],

    'Score': [85, 90, 88, 75, 78, 92, 95]

}

df = pd.DataFrame(data)

print("Original Data:\n", df)

df['Mean by Group'] = df.groupby('Class')['Score'].transform(lambda x: x.mean())

print("\nMean within each class:\n", df)

######Standard deviation by Class

import pandas as pd

# Sample data

data = {

    'Class': ['A', 'A', 'A', 'B', 'B', 'C', 'C'],

    'Student': ['S1', 'S2', 'S3', 'S4', 'S5', 'S6', 'S7'],

    'Score': [85, 90, 88, 75, 78, 92, 95]

}

df = pd.DataFrame(data)

print("Original Data:\n", df)

df['Standard deviation by Group'] = df.groupby('Class')['Score'].transform(lambda x: x.std())

print("\nSD within each class:\n", df)

######Mean and Standard deviation by Class

import pandas as pd

# Sample data

data = {

    'Class': ['A', 'A', 'A', 'B', 'B', 'C', 'C'],

    'Student': ['S1', 'S2', 'S3', 'S4', 'S5', 'S6', 'S7'],

    'Score': [85, 90, 88, 75, 78, 92, 95]

}

df = pd.DataFrame(data)

print("Original Data:\n", df)

df['Standard deviation by Group'] = df.groupby('Class')['Score'].transform(lambda x: x.std())

df['Mean by Class'] = df.groupby('Class')['Score'].transform(lambda x: x.mean())

print("\nSD and Mean within each class:\n", df)

######Mean by Class and Student Section

import pandas as pd

data = {

    'Class': ['A', 'A', 'A', 'A', 'B', 'B', 'B', 'B', 'B', 'B', 'C', 'C'],

    'section': ['S1', 'S2','S1', 'S2', 'S3', 'S4', 'S5', 'S3', 'S4','S5', 'S7','S7'],

    'Score': [85, 90, 88, 75, 78, 92, 95, 85, 90, 88, 75, 78]

}

df = pd.DataFrame(data)

print("Original Data:\n", df)

df['Mean by Class and section'] = df.groupby(['Class', 'section'])['Score'].transform(lambda x: x.mean())

print("\nSD within each class:\n", df)

##### Contigency table

import pandas as pd

# Sample categorical data

data = {

    'Gender': ['Male', 'Female', 'Female', 'Male', 'Male', 'Female', 'Male', 'Female', 'Female', 'Male'],

    'Class': ['A', 'A', 'B', 'B', 'A', 'B', 'A', 'A', 'B', 'B'],

    'Passed': ['Yes', 'No', 'Yes', 'No', 'Yes', 'Yes', 'No', 'Yes', 'No', 'Yes']

}

df = pd.DataFrame(data)

print(" Sample Data:\n", df)

contingency\_table = pd.crosstab([df['Class'], df['Gender']], df['Passed'], margins=True)

print("\n Contingency Table (Class + Gender vs Passed):\n", contingency\_table)

#### Pivot table

import pandas as pd

data = {

    'Class': ['A', 'A', 'B', 'B', 'A', 'B'],

    'Gender': ['Male', 'Female', 'Female', 'Male', 'Male', 'Female'],

    'Score': [85, 90, 88, 75, 78, 92]

}

df = pd.DataFrame(data)

pivot = pd.pivot\_table(df, values='Score', index='Class', columns='Gender', aggfunc='sum')

print("Pivot Table:\n", pivot)

##### Cross tab

crosstab = pd.crosstab(df['Gender'], df['Class'])

print("\nCross-Tabulation:\n", crosstab)

data2 = {

    'Gender': ['Male', 'Female', 'Female', 'Male', 'Male', 'Female'],

    'Passed': ['Yes', 'No', 'Yes', 'No', 'Yes', 'Yes']

}

df2 = pd.DataFrame(data2)

contingency = pd.crosstab(df2['Gender'], df2['Passed'])

print("\nContingency Table:\n", contingency)

###### Chi\_square table

import pandas as pd

from scipy.stats import chi2\_contingency

# Data

data2 = {

    'Gender': ['Male', 'Female', 'Female', 'Male', 'Male', 'Female','Female', 'Male', 'Male', 'Female'],

    'Passed': ['Yes', 'No', 'No', 'Yes', 'Yes', 'No', 'No','No','Yes','No']

}

df2 = pd.DataFrame(data2)

# Step 1: Create contingency table

contingency = pd.crosstab(df2['Gender'], df2['Passed'])

print("\n Contingency Table:\n", contingency)

# Step 2: Apply Chi-Square test

chi2, p, dof, expected = chi2\_contingency(contingency)

# Step 3: Print results

print("\n Chi-Square Test Results:")

print(f"Chi-square statistic = {chi2:.4f}")

print(f"Degrees of freedom = {dof}")

print(f"P-value = {p:.4f}")

print("\nExpected Frequencies:\n", pd.DataFrame(expected, index=contingency.index, columns=contingency.columns))

row\_percent = contingency.div(contingency.sum(axis=1), axis=0) \* 100

print("\nRow-wise Percentage (%):\n", row\_percent.round(2))

# Assuming 'contingency' is your contingency table

column\_percent = contingency.div(contingency.sum(axis=0), axis=1) \* 100

# Print the column-wise percentage, rounded to 2 decimal places

print("\nColumn-wise Percentage (%):\n", column\_percent.round(2))

combined = contingency.astype(str) + " (" + row\_percent.round(1).astype(str) + "%" ")"

print(" Contingency Table with Row-wise Percentages:\n", )

print(combined)