import pandas as pd

import matplotlib.pyplot as plt

from matplotlib.ticker import PercentFormatter, MultipleLocator

# Step 1: Specify the list of files (both CSV and Excel)

csv\_files = [

    'second\_year\_in\_7th.csv',

    '2nd year (b-16) in 8th class.csv',

    '2nd year (b-16) in 9th class.xlsx',

    'second\_year\_in\_1st\_year\_2.csv',

    'batch 16 in 2nd year.csv'

]

# Step 2: Specify the columns to analyze

specified\_columns = [

    'Islamic', 'Islamic\_History', 'Personality', 'Short Story',

    'Poetry', 'Language', 'Geography', 'History', 'Scientific', 'Society', 'Novel'

]

# Convert specified columns to lowercase for standardization

specified\_columns = [col.lower().replace(" ", "\_") for col in specified\_columns]

# Initialize a list to hold the percentage data for each file

all\_percentages = []

# Step 3: Loop through each file

for file in csv\_files:

    # Check file extension and read accordingly

    if file.endswith('.csv'):

        df = pd.read\_csv(file, encoding='ISO-8859-1')  # Use 'ISO-8859-1' encoding for non-UTF-8 CSVs

    elif file.endswith('.xlsx'):

        df = pd.read\_excel(file)

    else:

        raise ValueError(f"Unsupported file type: {file}")

    # Standardize column names by stripping whitespace and converting to lowercase with underscores

    df.columns = df.columns.str.strip().str.replace(" ", "\_").str.lower()

    # Check if the specified columns exist in the DataFrame

    missing\_columns = [col for col in specified\_columns if col not in df.columns]

    if missing\_columns:

        print(f"Warning: The following columns are missing in {file}: {missing\_columns}")

        # Skip this file if any required column is missing

        continue

    # Calculate the sum of each specified column

    column\_sums = df[specified\_columns].sum()

    # Calculate the total sum of all these columns

    total\_sum = column\_sums.sum()

    # Calculate the percentage contribution of each column

    percentages = (column\_sums / total\_sum) \* 100

    # Append the percentages to the list

    all\_percentages.append((file, percentages))

# Step 4: Plot each dataset's percentages as a separate line

plt.figure(figsize=(12, 8))  # Adjust figure size if needed

# Plot each file's percentages

for file, percentages in all\_percentages:

    plt.plot(

        percentages.index,

        percentages.values,

        marker='o',

        linewidth=2,

        label=file  # Use the filename as the label for the legend

    )

# Step 5: Add labels, title, and formatting

plt.xlabel('Books Types', fontsize=12)

plt.ylabel('Reading Strategy (%)', fontsize=12)

plt.title('Percentage Comparison of Column Sums across Multiple Datasets', fontsize=14)

# Rotate x-axis labels vertically

plt.xticks(rotation=75)

# Format y-axis as percentage

plt.gca().yaxis.set\_major\_formatter(PercentFormatter())

# Set y-axis limit to 25%

plt.ylim(0, 35)

# Set y-axis ticks to include sub-integer values (0.1 steps)

plt.gca().yaxis.set\_major\_locator(MultipleLocator(1))

# Add a grid background

plt.grid(axis='both', linestyle='-', alpha=0.7)

# Add legend

plt.legend()

# Step 6: Display the line chart

plt.show()