Marta Kuźma, Gdańsk University of Technology
ORCID: 0000-0002-0761-9362

Documentation of the problem

The goal of the project is to analyse data on immovable monuments in Poland across voivodeships for three selected years: 2016, 2024 and 2025. The input data is stored in an Excel file containing the voivodeship name and the number of monuments for each of the three years.
The program must:

- read the data from the Excel file into an internal data structure,
- identify those voivodeships where the number of monuments in 2025 is higher than in both 2016 and 2024,
- compute the growth in the number of monuments between 2016 and 2025 for each voivodeship and return it as an array of numeric values,
- present the number of monuments for the selected years on a bar chart.

Documentation of the design

The program is implemented in Python and uses the pandas library for reading and processing tabular data and matplotlib for creating the bar chart. The data file "Immovable-monuments-by-voivodeship.xlsx" contains four columns: VOIVODESHIP, YEAR 2016, YEAR 2024 and YEAR 2025.
The design is modular and based on a set of functions with clearly defined responsibilities:

- load_monuments_data(filepath: str) -> pd.DataFrame
  This function reads the Excel file using pandas.read_excel and returns the data as a DataFrame.
  Column names are stripped of whitespace to ensure consistent access by column name.
- voivodeships_with_increase(df: pd.DataFrame) -> pd.DataFrame
  This function identifies voivodeships where the number of monuments in 2025 is greater than in 2016 and 2024.
  A boolean mask is constructed using a decision expression:
  $\text{YEAR 2025} > \text{YEAR 2016} \land \text{YEAR 2025} > \text{YEAR 2024}$
  The function returns a filtered DataFrame containing only the rows (voivodeships) that meet this condition.
- compute_growth_array(df: pd.DataFrame, year_start: str = "YEAR 2016", year_end: str = "YEAR 2025")
  This function computes the growth in the number of monuments between two years for each voivodeship. The growth is calculated as the difference between the values in year_end and year_start columns, and the resulting Series is converted to a numeric array using to_numpy().
- plot_monuments_by_year(df: pd.DataFrame, years: list, title: str)
  This function produces a grouped bar chart showing the number of monuments in the selected years for each voivodeship.
  It uses a loop over the list of year column names (years) to create a bar for each year at a shifted position, so the bars for different years appear side by side for the same voivodeship.
  The x-axis displays voivodeship names, rotated for better readability, and the y-axis shows the number of monuments.

- main()
  The main function orchestrates the workflow of the program.
  First, it calls load_monuments_data to read the data from the file.
  Next, it calls voivodeships_with_increase and prints the names of the voivodeships that meet the increase condition, using a loop over the VOIVODESHIP column of the filtered DataFrame.
  Then it calls compute_growth_array and prints the resulting array of growth values.
  Finally, it calls plot_monuments_by_year with the list ["YEAR 2016", "YEAR 2024", "YEAR 2025"] to generate the bar chart.

Documentation of the tests

Testing focused on verifying both correctness of the results and fulfilment of the assignment requirements:

- Verify that load_monuments_data returns the expected number of rows and columns.
- Manually confirm that selected voivodeships printed by voivodeships_with_increase satisfy the numeric condition in the Excel file.
- Compare a few growth values from compute_growth_array with hand-calculated differences.
- Run the full main() flow and confirm that the console output and bar chart match the values in the Excel sheet for several sample voivodeships.