DATa Processing

Documentation ~ Nevin Tom 09/03/2024

Task 2

# Overview

This Python script is designed to perform data cleaning, analysis, and visualization tasks on CSV files. It reads a CSV file, calculates column totals and statistical summaries, generates a cleaned data file with totals and summary statistics, creates a bar plot for visualizing the total values of each column, and provides an interactive feature to sort and display data based on a selected column.

import pandas as pd  
import os  
import matplotlib.pyplot as plt

You'll need to install the following Python packages:

1. pandas: This is a powerful data manipulation and analysis library for Python. You can install it using pip:

pip install pandas

1. matplotlib: This is a plotting library for Python, which produces publication-quality figures in a variety of hardcopy formats and interactive environments. You can install it using pip:

pip install matplotlib

Make sure you have these packages installed before running the **main.py** script. If you're using an Anaconda distribution, these packages might be pre-installed, but you can also install them manually using the respective **conda** commands.

conda install pandas  
conda install matplotlib

Once you have the required packages installed, you should be able to run the main.py script without any issues.

# Functions

1. cleanData(file):
   * This function reads a CSV file specified by the **file** parameter.
   * It uses the**pandas** library to read the CSV file with the specified separator (**;**) and header row (0).
   * The cleaned data is returned as a pandas DataFrame.
2. calculateTotals4ColumnsAndStatistics(data, file):
   * This function calculates the totals for each numeric column in the input **data** DataFrame.
   * It also generates summary statistics (count, mean, std, min, max, and quantiles) for the data.
   * The totals and summary statistics are concatenated with the original data and saved to a new CSV file with the name **<original\_filename>CleansedData.csv**.
3. createNameofFile2Save(f, ext):
   * This function creates the name and path for the output file to be saved.
   * It takes two parameters: **f** (the input file path) and **ext** (the desired file extension).
   * The function returns the full path of the output file with the specified extension, using the input file's basename and the current working directory.
4. plotGraph(data, f):
   * This function generates a bar plot to visualize the total values of each column in the input **data** DataFrame.
   * It calculates the column totals and creates a bar plot using the **matplotlib** library.
   * The plot is saved as a PNG file with the name **<original\_filename>CleansedData.png**.

main():

* + This is the main function that orchestrates the execution of the script.
  + It prompts the user to enter the full path of the CSV file to be processed.
  + It calls the **cleanData()** function to read and clean the data.
  + It calls the **calculateTotals4ColumnsAndStatistics()** function to calculate totals and summary statistics, and save the cleaned data to a CSV file.
  + It calls the **plotGraph()** function to generate and save the bar plot visualization.
  + It enters an interactive loop where the user can choose a column to sort and display the data. The loop continues until the user enters 'EXIT'.

# Usage

To run the script, navigate to the directory containing the **main.py** file in your terminal or command prompt, and execute the following command:

python main.py

The script will prompt you to enter the full path of the CSV file you want to process. After providing the path, the script will perform the data cleaning, analysis, and visualization tasks, and generate the corresponding output files.

Once the initial processing is complete, the user will input the **path of the csv** file; the script will enter an interactive loop where you can enter the name of a column (case-insensitive) to sort and display the data based on that column. To exit the loop, enter 'EXIT'.

Note: The script assumes that the input CSV file has a header row and uses the semicolon (;) as the delimiter.

# When code is run

1. The script prompts the user to enter the full path of the CSV file they want to process:

C:\Users\User\Documents>python main.py  
Enter the full path of the csv file:

1. The user enters the path to the CSV file **(e.g., "C:\path\to\file.csv").**

"C:\Users\User\Documents\cereal.csv"

1. The **cleanData()** function is called, which reads the CSV file using pandas and returns a cleaned DataFrame.
2. The **calculateTotals4ColumnsAndStatistics()** function is called with the cleaned DataFrame and the input file path. This function:
   1. Calculates the totals for each numeric column in the DataFrame.
   2. Generates summary statistics (count, mean, std, min, max, and quantiles) for the data.
   3. Concatenates the totals and summary statistics with the original data.
   4. Saves the resulting DataFrame to a new CSV file with the name **<original\_filename>CleansedData.csv**.
3. The **plotGraph()** function is called with the cleaned DataFrame and the input file path. This function:
   1. Calculates the total values for each column in the DataFrame.
   2. Creates a bar plot to visualize the total values of each column.
   3. Saves the bar plot as a PNG file with the name **<original\_filename>CleansedData.png**.
4. After the data cleaning, analysis, and visualization tasks are completed, the script enters an interactive loop where the user can filter and sort the data based on a chosen column.
5. The script displays a list of column names from the cleaned DataFrame and prompts the user to enter the name of the column they wish to filter:

Enter the column name you wish to filter:  
Name

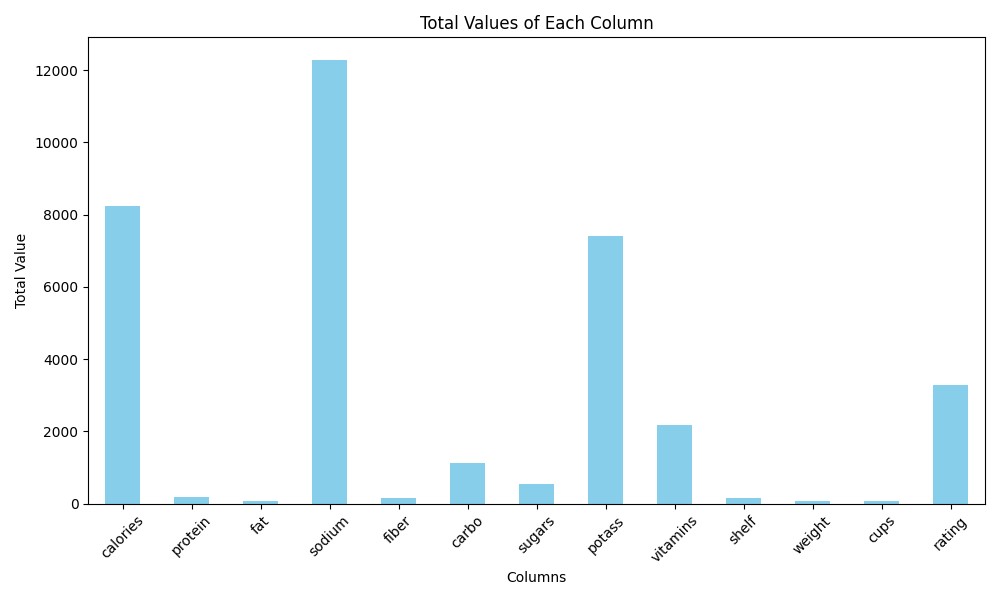
Mfr

Type  
...  
(Enter 'EXIT' to exit)

1. If the user enters a valid column name, the script sorts the DataFrame based on that column and prints the sorted data to the console.
2. If the user enters 'EXIT', the script exits the loop and terminates.
3. If the user enters an invalid column name, the script displays an error message: "Column selected does not exist!"

So, in summary, the user provides the CSV file path, the script performs data cleaning, analysis, and visualization tasks, and then allows the user to interactively filter and sort the data based on a chosen column.

# Data Visualization from CSV file



# Conclusion

This Python script provides a convenient way to clean, analyze, and visualize data from CSV files. By automating tasks such as calculating column totals, generating summary statistics, creating visual representations, and enabling interactive data filtering, it streamlines the data analysis workflow.

It's important to note that the script assumes a specific CSV file format, with a header row and semicolon (**;**) as the delimiter. If your CSV files have a different format, you may need to modify the **cleanData()** function accordingly.

Additionally, the script relies on the **pandas** and **matplotlib** libraries, which provide powerful data manipulation and visualization capabilities. Familiarity with these libraries can be beneficial for understanding and extending the functionality of the script.

Overall, this script serves as a starting point for data analysis tasks, and you can further enhance it by adding more features or integrating it into larger data analysis pipelines as per your specific requirements.