Data Processing Script Documentation

Introduction

This Python script is designed to perform basic data processing tasks on a dataset using the Iris dataset as an example. The script is versatile and can be used with different datasets. It includes functions for reading a dataset, calculating summary statistics, filtering data based on specific criteria, generating histograms, and saving the processed data to a new file.

Dependencies

The following Python libraries should be installed before running the script:

- pandas
- matplotlib
- seaborn
- numpy
- scipy

How to Use the Script

1. Load Required Libraries

In a Jupyter notebook or Python script, start by importing the necessary libraries:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
```

2. Load the Iris Dataset

```
iris = sns.load_dataset('iris')
```

3. Summary Statistics Calculation

Calculate mean, median, and mode for the numeric columns in the dataset:

```
# Select only numeric columns
numeric_data = iris.select_dtypes(include='number')
# Calculate mean, median, and mode
mean_values = numeric_data.mean()
median_values = numeric_data.median()
mode_values = numeric_data.mode().iloc[0]
```

4. Data Processing Functions

The script provides several functions for data processing:

- `read_dataset(file_path)`: Reads a dataset from a CSV file.
- `calculate_summary_statistics(data)`: Calculates and displays summary statistics for the dataset.
- `filter_data(data, column_name, criteria)`: Filters data based on a specific criteria.
- `generate_histogram(data, column_name)`: Generates a histogram for a specific column.
- `save_processed_data(data, output_file)`: Saves the processed data to a new CSV file.

5. Example Usage

The script includes an example usage section that demonstrates how to use the functions with the Iris dataset:

```
# Read the dataset
iris_data = read_dataset(input_file_path)
# Calculate summary statistics
calculate_summary_statistics(iris_data)
# Filter data based on specific criteria
filtered_data = filter_data(iris_data, 'SepalLengthCm', 5.0)
# Generate histogram for a specific column
generate_histogram(iris_data, 'SepalLengthCm')
# Save processed data to a new file
save_processed_data(filtered_data, output_file_path)
```

8. Inspect Outputs

Inspect the printed summary statistics, filtered data, and generated histograms. The processed data will be saved to the specified output file.

9. Customize as Needed

Customize the script as needed, such as adjusting filtering criteria, column names, or the number of bins in histograms.

10. Repeat for Different Datasets

The script is designed to be versatile and can be used with different datasets. Simply replace the dataset and file paths as needed.