Analyzing Data with Pandas and Visualizing Results with Matplotlib

Description

Task 1: Load and Explore the Dataset

1. Choose a dataset in CSV format (for example, you can use datasets like the Iris dataset, a sales dataset, or any dataset of your choice).
2. Load the dataset using pandas.
3. Display the first few rows of the dataset using .head() to inspect the data.
4. Explore the structure of the dataset by checking the data types and any missing values.
5. Clean the dataset by either filling or dropping any missing values.

Task 2: Basic Data Analysis

1. Compute the basic statistics of the numerical columns (e.g., mean, median, standard deviation) using .describe().
2. Perform groupings on a categorical column (for example, species, region, or department) and compute the mean of a numerical column for each group.
3. Identify any patterns or interesting findings from your analysis.

Task 3: Data Visualization

1. Create at least four different types of visualizations:
    - **Line chart** showing trends over time (for example, a time-series of sales data).
    - **Bar chart** showing the comparison of a numerical value across categories (e.g., average petal length per species).
    - **Histogram** of a numerical column to understand its distribution.
    - **Scatter plot** to visualize the relationship between two numerical columns (e.g., sepal length vs. petal length).
2. Customize your plots with titles, labels for axes, and legends where necessary.

Additional Instructions

1. **Dataset Suggestions:**
    - You can use publicly available datasets from sites like Kaggle or UCI Machine Learning Repository.
    - The Iris dataset (a classic dataset for classification problems) can be accessed via sklearn.datasets.load_iris(), which can be used for the analysis.
2. **Plot Customization:**
    - Customize the plots using the matplotlib library to add titles, axis labels, and legends.
    - Use seaborn for additional plotting styles, which can make your charts more visually appealing.
3. **Error Handling:**
    - Handle possible errors during the file reading (e.g., file not found), missing data, or incorrect data types by using exception-handling mechanisms (try, except).
4. **Submission:**
    - Ensure your submission is complete with all necessary code and explanations. Make sure that each plot is properly labeled and provides insights into the dataset.

Objective For this Assignment:

- To load and analyze a dataset using the pandas library in Python.
- To create simple plots and charts with the matplotlib library for visualizing the data.

Submission Requirements

- Submit a Jupyter notebook (.ipynb file) or Python script (.py file) containing:
    - Data loading and exploration steps.
    - Basic data analysis results.
    - Visualizations.
    - Any findings or observations.