PROJECT PAPER, DATED: 16/11/2024

Numerical Analysis of Stocks and Sales Data with NumPy

This project uses NumPy to analyze stocks and sales data, focusing on various financial metrics across different sectors. Through tasks like data cleaning, correlation calculations, and outlier detection, it demonstrates NumPy's power in deriving insights from real-world data. Each task applies key NumPy functions to reveal patterns, offering a clear view of financial trends without relying on additional libraries.

1. Data Loading and Preparation

Task: Load the CSV data into a NumPy structured array.
Explanation: Using np.genfromtxt() to read the CSV file into a NumPy array, handling missing values by replacing them with NaNs. The array should be structured to align with the dataset's column types, ensuring compatibility for analysis.
Key Functions: np.genfromtxt()

2. Data Cleaning: Handle Missing and Text Values

Task: Identify and handle missing values in numeric columns, replacing them with the column mean.
Explanation: Missing values are identified using np.isnan(), and replaced with the mean value using np.nanmean(). This completes the data, allowing consistent analysis across columns.
Key Functions: np.isnan(), np.nanmean()

3. Descriptive Statistics for Stock Prices by Sector

Task: Calculate the mean, standard deviation, min, and max for opening and closing prices by sector.
Explanation: Using np.mean(), np.std(), np.min(), and np.max() on stock prices provides a summary of central tendency and spread, revealing overall price variations across sectors.
Key Functions: np.mean(), np.std(), np.min(), np.max()

4. Price Comparison by Sector

Task: Calculate the average opening and closing prices for each sector.
Explanation: Grouping by 'Sector' and applying np.mean() to stock prices reveals sector-based differences in stock prices.
Key Functions: np.mean()

5. Correlation Analysis of Opening and Closing Prices

Task: Calculate the correlation between opening and closing prices.
Explanation: Using np.corrcoef() to find the Pearson correlation coefficient between opening and closing prices. This indicates if prices follow a consistent pattern.
Key Functions: np.corrcoef()

6. Sector Analysis of Market Capitalization

Task: Group stocks by their sector and calculate average market capitalization for each sector.
Explanation: Identifying unique sectors with np.unique() and then calculating the mean market capitalization for each gives insights into sectoral financial health.
Key Functions: np.unique(), np.mean()

7. Identifying Highest and Lowest Stock Prices

Task: Identify stocks with highest and lowest opening and closing prices.
Explanation: np.argmax() and np.argmin() locate the highest and lowest values, allowing retrieval of corresponding stock data.
Key Functions: np.argmax(), np.argmin()

8. Outlier Detection in Stock Prices

Task: Identify outliers in stock prices using the Interquartile Range (IQR) method.
Explanation: Calculate Q1 and Q3 using np.percentile(), find the IQR, and identify outliers outside the 1.5*IQR range.
Key Functions: np.percentile()

9. Comparing Stock Prices Against a Threshold

Task: Calculate the proportion of stocks with closing prices above a threshold (e.g., $200).
Explanation: Using logical operations in NumPy to count stocks exceeding the threshold and find the percentage.
Key Functions: Logical operations, np.sum()

10. Trend Over Time (If Period Data Exists)

Task: Calculate stock price changes over time for each stock.
Explanation: Use np.diff() on stock prices to determine changes across time periods, highlighting price trends.
Key Functions: np.diff()