Dataset Preprocessing

**Stocks Dataset:**

**Overview**

The script processes stock market data, focusing on preparing two sets of data: 'truth values and 'source values.' The truth values serve as a benchmark or reference. In contrast, the source values come from various sources and must be compared against the truth values. The script organizes and cleans these data sets, ensuring consistency and reliability for further analysis.

**Truth Values Preprocessing**

1. Data Extraction and Cleaning:
   1. The script starts by reading files containing truth values. These are assumed to be reliable and accurate representations of stock market data.
   2. It cleans and formats the data by removing unnecessary characters (like commas and dollar signs), ensuring it is numeric for easy analysis.
2. Selection of Relevant Variables:
   1. Only continuous variables (like 'Volume,' 'Shares Outstanding,' and 'Market Cap') are retained for each stock symbol. This focus helps in tracking variables that change over time and have a direct impact on market analysis.
3. Handling Missing Data:
   1. The script identifies if any stock symbols have missing data for a particular day.
   2. If data is missing, it's filled with the most recent available values. This method ensures the dataset is complete and allows consistent comparisons over time.
4. Data Structuring:
   1. The data is organized into a 'truth matrix,' a structured format that aligns data across different days and variables, making it easier to compare and analyze.

**Source Values Preprocessing**

1. Data Extraction and Initial Cleaning:
   1. Similar to truth values, source values are extracted and cleaned. These values come from different sources and may have inconsistencies.
   2. The script ensures that only data for stock symbols found in the truth values are considered, maintaining consistency across datasets.
2. Replacing Textual Representations:
   1. Textual representations of numbers (like 'billion,' 'mil,' etc.) are converted into numerical equivalents (e.g., 'billion' to 'e9'). This standardizes the data, making it numerical and easier to analyze.
3. Handling Missing Data:
   1. For any missing stock symbols in the source data, the script assigns NaN (Not a Number), a standard way to denote missing or undefined values.
4. Data Normalization:
   1. The script normalizes each source variable by the standard deviation of themselves. This step is crucial for bringing data from different sources and variables onto a standard scale, enabling fair comparison and analysis.

**Final Output**

The script produces two main outputs: a 'truth matrix' and a 'sources matrix.' These matrices are structured to quickly compare the truth values and the various source values. By the end of the preprocessing, the data is clean, consistent, and structured, ready for subsequent analysis, such as comparing the accuracy of different sources against the truth values.

**Weather Dataset:**

**Overview**

This script processes weather data from multiple sources. Its primary goal is to standardize and clean data for consistency, allowing for accurate comparisons across different weather reporting services. These truth values are observations of the climatic phenomena that took place, different from the source values that are predictions.

**Truth Values Preprocessing:**

1. Data Extraction and Cleaning:
   1. The script begins by reading a file's 'truth values.' These values are assumed to be accurate weather data.
   2. It discards unnecessary headers and sets variable names corresponding to weather conditions like temperature, humidity, etc.
2. Data Standardization:
   1. The script ensures that the data is in a consistent format and physic units, focusing on continuous variables like temperature, humidity, and wind speed, share the same units for all observations.
   2. Timestamps are formatted for uniformity and sorted to align the data chronologically.
3. Handling Missing Data:
   1. Missing values are identified and filled using linear interpolation. This method provides a best guess for missing data points based on the available data, ensuring a complete dataset for analysis.
4. Structuring Data:
   1. The data is structured into a 'truth matrix.' This matrix aligns weather data across different timestamps and locations, creating a comprehensive view of the weather conditions.

**Source Values Preprocessing**

1. Data Extraction and Initial Cleaning:
   1. Similar to the truth values, the script processes data from various sources. Each source has its format and set of variables.
   2. The script renames and reorders variables to match the format of the truth values, focusing on the relevant continuous variables.
2. Unit Conversion and Data Standardization:
   1. Units are standardized (e.g., pressure and wind speed) to ensure consistency across different sources.
   2. Textual representations are converted to numeric values; for instance, windspeeds with text values of 'Calm' are set to zero.
3. Handling Location and Time Data:
   1. The script ensures that only data for locations and timestamps matching those in the truth values are considered.
   2. It addresses duplications in timestamps and locations, ensuring that each timestamp has a unique weather value.
4. Data Normalization:
   1. The source values are normalized by the standard deviation of the properties. This step is crucial for comparing data from different sources and variables on a standard scale.

**Final Output**

The script produces a 'truth matrix' and a 'sources matrix.' The truth matrix contains reliable weather data, while the sources matrix contains data from various prediction services. These matrices are designed to easily compare the truth values and the values reported by different sources.

**Pedestrian Dataset:**

**Overview**

This section outlines the process of reading and simulating data for a pedestrian dataset. Unlike previous datasets, this script focuses solely on generating source data from the truth data by introducing controlled noise. The simulated data mimics variations in real-world measurements from different sources.

**Truth Values Preprocessing**

1. Data Extraction:
   1. The script begins by reading the truth data from CSV files. This data represents accurate pedestrian counts on various streets.
   2. Each file corresponds to a street, and the data includes counts of people entering (IN), exiting (OUT), and the total count (Street Count).
2. Data Cleaning and Interpolation:
   1. Missing values in the truth data are filled using linear interpolation. This approach estimates missing values based on neighboring data points, ensuring a complete dataset.
3. Data Normalization:
   1. The truth data is scaled to ensure all values are positive. This is crucial for the subsequent noise addition process, as it avoids negative counts.
   2. Outliers in the data are identified and replaced with linear interpolations, smoothing extreme variations that might not represent typical pedestrian traffic.

**Simulating Source Data**

1. Initializing the Sources Matrix:
   1. A sources matrix is initialized and structured to contain data for a specified number of sources over the same time and property dimensions as the truth data.
2. Generating Noise for Simulation:
   1. The script generates random noise levels for each source within specified minimum and maximum bounds. This randomness introduces variability in the data quality across different sources.
   2. The noise level for each source is a factor of the variance of the property values in the truth data, ensuring that the noise is proportionate and realistic.
3. Applying Noise to Simulate Sources:
   1. The script applies noise to the truth data using the 'awgn' function, which adds white Gaussian noise to the signal. The noise level is adjusted to maintain a specific signal-to-noise ratio (SNR), making the simulated source data realistically varied.
   2. Each property in the truth matrix is processed separately, allowing for unique noise characteristics for each data series.
4. Normalization of Sources Matrix:
   1. Finally, the sources matrix is normalized by the properties' standard deviation. This normalization is crucial for fair comparison and analysis across different sources and properties.

**Final Output**

The script produces a `simulated\_data` structure containing the truth matrix, the normalized sources matrix, the properties' standard deviation, the noise levels for each source, and the sources' names. This simulated data mimics real-world scenarios where different sources report data with varying accuracy and noise levels.