Data cleaning pseudo code

1. IMPORT THE IMPORTANT AND REQUIRED PYTHON LIBRARIES

To begin the data cleaning process, firstly the three Python libraries, NumPy, Pandas, and Matplotlib must be imported. The libraries must be imported before they can be used.

2. THE SOURCE DATASET

Create a variable data location to store the location of the dataset. Use the pandas library to read the comma-separated dataset.

3. EXPLORATORY DATA ANALYSIS

This task will start by performing an exploratory data analysis (EDA) on the dataset to determine whether there are any discrepancies. We'll move forward as follows:

* Use the **df.shape** attribute to determine the dimension or size of our dataset as well as the **df.head() and df.tail()** methods to view the top five and bottom five rows of the dataset respectively.
* To continue, use the **df.info()** method to print the information of the dataFrame including the index dtype and column dtypes, non-null values and memory usage.
* Use the **df.dtypes** attribute to check the data types of each column in the dataframe. This command returns the data type of each column.
* Use the **df.describe**() funtion to display the descriptive statistics of the dataset.
* To Handle invalid values coded as "xx","emsg" and "?" in the dataset, coerce them to "NaN" using the errors keyword.
* Use the **df.columns** attribute to display the column names.

4. VISUAL EXPLORATORY DATA ANALYSIS

Now conduct data visualization to find discrepancies in the data. This is a great way to find errors in the data and detect outliers. This helps with detecting patterns in the data. This can be done with the utilization of various types of plots such as Bar plots, Histograms, Box plots and Scatter plots for data visualization purposes.

Use Young’s modulus and density from the dataset as the markers to display the data.

5. SPLIT ‘PVelocity\_SVelocity’ COLUMN

Split the 'PVelocity\_SVelocity’ column into two separate columns using the **df.str.split()** function. Once this column has been split, use the **df.drop()** method to drop the original combined column.

6. Dealing with negative numerical values

To deal with negative values suppress the setting with copy warning, and replace all the negative values with their positive counterparts. For example, the -54 in Young’s modulus will be replaced by 54 and so is all the other negative values.

7. DEALING WITH MISSING NUMERICAL VALUES

Use the command **df.isnull.sum()** to find the total number of missing values in each column in the dataset. Fill in the missing values with a test statistic mean using the column names that consist of missing values.