Assignment Title: Comprehensive Data Cleaning, Manipulation, and Merging for Business Insights

Problem Statement:

A retail company has collected data from multiple sources, including sales, products, and customers. These datasets are inconsistent and need preprocessing before analysis. You are tasked with cleaning, manipulating, and merging the data to prepare a unified dataset for deriving actionable business insights.

Dataset Description:

1. **Sales Data**: Contains OrderID, CustomerID, ProductID, OrderDate, Quantity, and SalesAmount.
2. **Customer Data**: Includes CustomerID, Name, City, MembershipTier, and JoinDate.
3. **Product Data**: Includes ProductID, ProductName, Category, Price, and Cost.

Detailed Tasks:

Part 1: Data Cleaning

1. **Inspect the Data**:
    - Load datasets and inspect for missing (NaN), null, or duplicate entries.
    - Use functions like info(), describe(), and isnull() to identify inconsistencies.
2. **Handle Missing Data**:
    - Use fillna() to replace missing numerical data with the mean/median.
    - Replace missing categorical data using the mode or a placeholder (e.g., "Unknown").
    - Use dropna() to remove rows/columns with excessive missing values, providing justification.
3. **Standardize Data**:
    - Use replace() to standardize category names (e.g., "electronics" → "Electronics").
    - Convert textual data to lowercase for uniformity.
    - Ensure dates are in a consistent format (e.g., YYYY-MM-DD).
4. **Remove Duplicates**:
    - Use drop_duplicates() to remove duplicate rows while keeping the first occurrence.
5. **Sort the Data**:
    - Sort sales data by OrderDate and customer data by MembershipTier.

Part 2: Data Manipulation

1. **Filter Data**:
    - Use filter() by label to select only ProductID, Category, and Price columns from the product dataset.
    - Use conditional filtering to extract sales data for orders with a quantity greater than 10.
    - Use isin() to filter customers belonging to specific cities (e.g., "New York" and "Los Angeles").
2. **Add Derived Columns**:
    - Add a TotalRevenue column by multiplying Quantity and Price.
    - Create a ProfitMargin column (Profit = SalesAmount - Cost).
3. **Group and Aggregate**:
    - Group sales data by Category to calculate total revenue and average quantity.
    - Aggregate customer data to find the total spend per MembershipTier.
4. **Apply Operators**:
    - Use logical operators to create a new column HighValueOrder (True if TotalRevenue > $100).

Part 3: Data Merging

1. **Merge Datasets**:
    - Use merge() to join sales and product datasets on ProductID.
    - Merge the resulting dataset with the customer dataset on CustomerID.
    - Experiment with inner, left, right, and outer joins to observe differences.
2. **Concatenate Datasets**:
    - Simulate appending new monthly sales data using concat().
    - Concatenate two customer datasets from different regions into a single dataset.
3. **Resolve Conflicts**:
    - Use replace() to address inconsistencies (e.g., same ProductID with slightly different names).
4. **Validate the Merged Data**:
    - Check for duplicate rows and resolve conflicts if any.
    - Ensure all columns are relevant and retain only essential ones.

Part 4: Analysis and Visualization

1. **Perform Analysis**:
    - Find the top 5 customers by total revenue.
    - Identify the product category generating the highest profit margin.
2. **Create Visualizations**:
    - Use a bar chart to display revenue by product category.
    - Use a line chart to show monthly revenue trends.
3. **Generate Insights**:
    - Write a brief report summarizing insights from the data analysis.

Expected Deliverables:

1. **Preprocessed Data**: Cleaned, manipulated, and merged dataset in CSV format.
2. **Code Implementation**: Python scripts or Jupyter Notebook containing:
    - fillna, dropna, drop_duplicates, sort_values, filter(), merge(), concat(), and isin() usage.
3. **Visualizations**: Charts and graphs created using libraries like matplotlib or seaborn.
4. **Summary Report**: Business insights and challenges encountered during the process.

Expected Outcome:

This assignment will provide practical experience in:

- Data cleaning techniques like handling missing values, duplicates, and standardization.
- Using Python data manipulation techniques with pandas, including merging and concatenation.
- Filtering data with labels, conditions, and logical operators.
- Performing data analysis and creating visualizations for actionable insights.