📌 Level 1: Foundations of Data Analysis

Scope: Python, Numpy, Pandas, Matplotlib, Seaborn, Exploratory Data Analysis (EDA), Descriptive Statistics.

Scenario: You've just been hired as a junior data analyst for **"FitLife"**, a company that sells fitness apparel and accessories online. You've been given a dataset (fitlife_sales.csv) containing sales data for the last quarter. Your first task is to perform an initial exploratory analysis to uncover basic insights.

Hypothetical Dataset Columns:

- OrderID (string)
- ProductID (string)
- Product (string: e.g., 'Yoga Mat', 'Running Shoes')
- Category (string: 'Apparel', 'Equipment', 'Accessories')
- Price_USD (float)
- Units_Sold (integer)
- Order_Date (string: format 'YYYY-MM-DD')
- Customer_ID (string)
- Customer_Age (integer)
- Customer_Gender (string: 'Male', 'Female', 'Non-binary')
- Region (string: 'North', 'South', 'East', 'West')

🧠 Conceptual Question (1)

1. **The Role of a Data Analyst:** Explain the difference between **Descriptive Statistics** and **Exploratory Data Analysis (EDA)**. While they are related, what unique purpose does each serve in a data analysis project? How would you use Pandas for descriptive statistics and Seaborn for EDA?

💻 Practical Questions (10)

For these questions, assume you have a file named fitlife_sales.csv that you will load into a Pandas DataFrame.

1. **Data Loading & Initial Inspection:** Load the fitlife_sales.csv dataset into a Pandas DataFrame called df. Display its first 5 rows, check the data types of each column, and

get a count of non-null values for all columns.

2. **Data Cleaning & Type Conversion:** The Order_Date column is currently a string. Convert it to a proper datetime format. Afterwards, report the earliest and latest order dates in the dataset.
3. **Feature Engineering:** Create a new column called Total_Revenue which is the result of multiplying Price_USD by Units_Sold.
4. **Descriptive Statistics:** Calculate the **mean**, **median**, and **standard deviation** for the Customer_Age and Total_Revenue columns. What can you infer from the difference between the mean and median of Total_Revenue?
5. **Filtering with NumPy & Pandas:** The marketing team wants to send a special discount to high-value customers. Using NumPy and Pandas, filter the DataFrame to show only the orders where Total_Revenue was greater than $200. How many such orders are there?
6. **Grouping & Aggregation:** Determine the total revenue generated by each Product Category. Which category is the most profitable for FitLife?
7. **Time-Series Visualization (Matplotlib):** The sales director wants to see the daily revenue trend. Create a line plot using **Matplotlib** that shows the total Total_Revenue for each day over the entire period. Ensure your plot has a title and labeled axes.
8. **Distribution Visualization (Seaborn):** To better understand your customer base, create a **histogram** using **Seaborn** to show the distribution of Customer_Age. Overlay a Kernel Density Estimate (KDE) on the plot to see the shape of the distribution more clearly.
9. **Categorical Data Visualization (Seaborn):** Which Region performs the best in terms of sales? Create a **bar plot** using **Seaborn** that shows the total Total_Revenue for each Region.
10. **Relationship Analysis (Matplotlib/Seaborn):** Is there a relationship between the price of an item (Price_USD) and the number of units sold (Units_Sold)? Create a **scatter plot** to visualize this. Calculate the correlation coefficient between these two variables and explain what it means.

🔍 Code Review & Debugging (1)

12. Your colleague wrote the following Python code to find which gender group (Customer_Gender) has the highest average transaction revenue. However, the code is inefficient and contains a logical error.
    Task: Identify the inefficiency and the error. Then, rewrite the code using a more direct and efficient "Pandas-native" approach.
    Python
    ```
    # Assume 'df' is the DataFrame with the 'Total_Revenue' column already created.
    genders = ['Male', 'Female', 'Non-binary']
    highest_avg_rev = 0
    top_gender = ''
    ```

```
for g in genders:
    gender_df = df[df['Customer_Gender'] == g]
    total_rev = gender_df['Total_Revenue'].sum()
    avg_rev = total_rev / len(gender_df) # Potential for DivisionByZeroError if a gender has no entries

    if avg_rev > highest_avg_rev:
        highest_avg_rev = total_rev # This is the logical error
        top_gender = g

print(f"The gender with the highest average revenue is: {top_gender}")
```

Good luck! I'm ready to review your solutions when you are.