**🧠 Superstore Data Analysis — Problem List by Stages**

**✅ Stage 1: Basic Data Loading & Exploration**

*(These are inferred from your process; originally marked as Uncategorized)*

* Load the Superstore dataset.
* Convert order\_date and ship\_date columns to datetime format.
* Set order\_date as the DataFrame index and sort it by date.
* Create a new column order\_month in YYYY-MM format for time-based aggregation.
* Preview rows using .head(), .tail(), and check structure with .info().
* Simulate and preview tables: Orders, Customers, Products (if dataset was extended).
* Make copies of data before performing transformations to preserve the original.

**✅ Stage 2: Filtering, Indexing & Selection**

* Filter orders where Sales > 1000.
* Filter data for a specific region (e.g., "West").
* Retrieve all rows where the customer is “Erica Bern”.
* Select specific columns: Order ID, Sales, and Profit.
* Use .iloc to slice rows (e.g., rows 10 to 20) and display Order ID, Order Date, and Ship Date.
* Use .loc to filter rows where Category is "Technology" and display Product Name and Sales.
* Filter orders where Category is "Technology" and Region is "East".

**✅ Stage 3: Aggregation & Grouping**

* Group by Region to calculate total sales.
* Group by Category to compute average profit.
* Count total orders per Customer Segment.
* Group by State to calculate total sales and total profit.
* Identify top 3 products with the highest average discount.
* Create new column profit\_margin = profit / sales.
* Bin the sales column into Low, Medium, and High categories.
* Create a column to classify sales into categories.
* Resample the data by month to compute monthly sales totals.
* Identify the month with the highest profit.
* Group by order\_month and Category to analyze category-wise monthly sales trends.

**✅ Stage 4: Data Cleaning & Transformation**

* Handle missing values in the postal\_code column.
* Replace or fill missing values (NaN) using appropriate methods.
* Remove duplicates and correct inconsistent data formats.
* Derive shipping\_delay = ship\_date - order\_date.

**✅ Stage 5: Merging & Joining**

* Merge the Orders table with the Customers table using customer\_id.
* Join any additional reference tables (e.g., product or calendar tables) to enrich data.

**✅ Stage 6: Pivot Tables & MultiIndexing**

* Create a pivot table showing total sales by Region and Category.
* Work with multi-indexed DataFrames to compute insights like profit by Category and Segment.
* Unstack a multi-indexed DataFrame for better readability and access.

**✅ Stage 7: Advanced Analysis & Operational Insights**

* Perform rolling 3-month average of sales or profits.
* Compare monthly sales with previous month using .shift() or .diff().
* Visualize trends in monthly sales using line plots (Matplotlib/Pandas).
* Analyze how discounts impact profitability.
* Identify segments or categories that consistently lead to losses.
* Detect potential shipping delays by comparing order and ship dates across states or modes.