**Data Analysis** - On Raw\_Data   
  
Based on the structure and content of the dataset you provided, which includes transactions with details such as transaction ID, product ID, customer ID, transaction date, whether the order was placed online, order status, product details (brand, line, class, size), list price, standard cost, and product first sold date, here are some problem statements for data visualization or analysis:

1. **Trend Analysis**:
   * Visualize the trend of sales over time. You can further break it down by product line or brand to see which products or brands are trending upwards or downwards.­­­
   * Analyze the seasonality in the sales data. Determine if there are specific months or seasons when sales peak.

# Step 1: Import necessary libraries

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Step 2: Load the dataset

data = pd.read\_excel(r'C:/Users/Lenovo/Downloads/Study material/Tasks/230224/New folder/Raw\_data.xlsx')

data1 = data;

data1.drop(columns = ["transaction\_id","product\_id","customer\_id"], inplace = True);

# Step 3: Data Preprocessing (if necessary)

# - Convert date columns to datetime format

# data1['transaction\_date'] = pd.to\_datetime(data['transaction\_date'])

# data1['product\_first\_sold\_date'] = pd.to\_datetime(data['product\_first\_sold\_date'])

# Extract month and year from 'transaction\_date'

data1['transaction\_month'] = data1['transaction\_date'].dt.to\_period('M')

# Group by 'transaction\_month' and 'brand', and calculate the total sales

sales\_by\_month\_brand = data1.groupby(['transaction\_month', 'brand'])['list\_price'].sum().reset\_index()

sales\_by\_month\_brand.info()

# Plot the trend using Seaborn barplot

plt.figure(figsize=(12, 8))

sns.barplot(x='transaction\_month', y='list\_price', hue='brand', data=sales\_by\_month\_brand)

plt.title('Total Sales Trend by Brand (Month-wise)')

plt.xlabel('Transaction Month')

plt.ylabel('Total Sales (List Price)')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()

sales\_by\_month\_brand['list\_price'] = pd.to\_numeric(sales\_by\_month\_brand['list\_price'], errors='ignore')

sns.boxplot(sales\_by\_month\_brand['list\_price'])

sales\_by\_month\_brand['list\_price'] = sales\_by\_month\_brand['list\_price'].astype(int)

# Convert 'transaction\_month' to string format for plotting

sales\_by\_month\_brand['transaction\_month\_string'] = sales\_by\_month\_brand['transaction\_month'].dt.strftime('%Y-%m')

from seaborn import color\_palette

palette = color\_palette("colorblind").as\_hex()

sns.set\_palette(palette)

# Plot the trend using Seaborn lineplot

plt.figure(figsize=(12, 8))

sns.lineplot(x='transaction\_month\_string', y='list\_price', hue='brand', data=sales\_by\_month\_brand, )

plt.title('Total Sales Trend by Brand (Month-wise)')

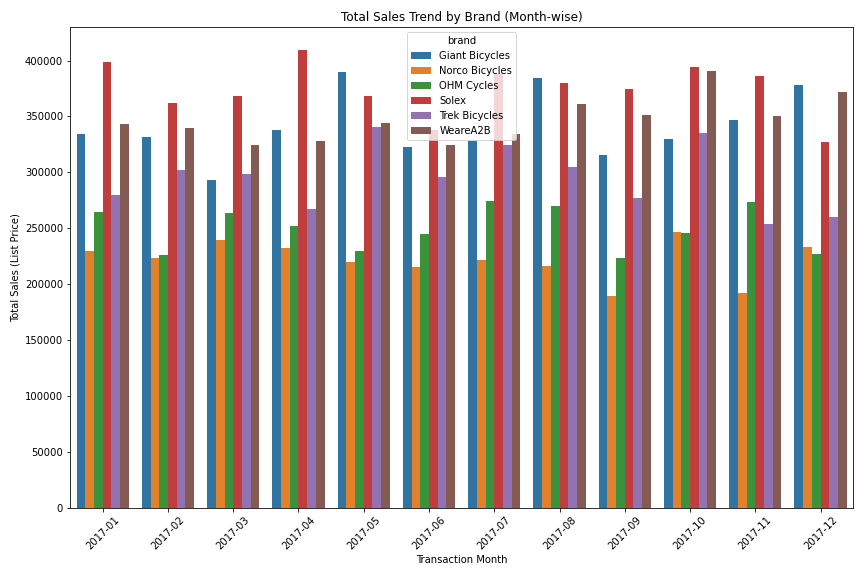
plt.xlabel('Transaction Month')

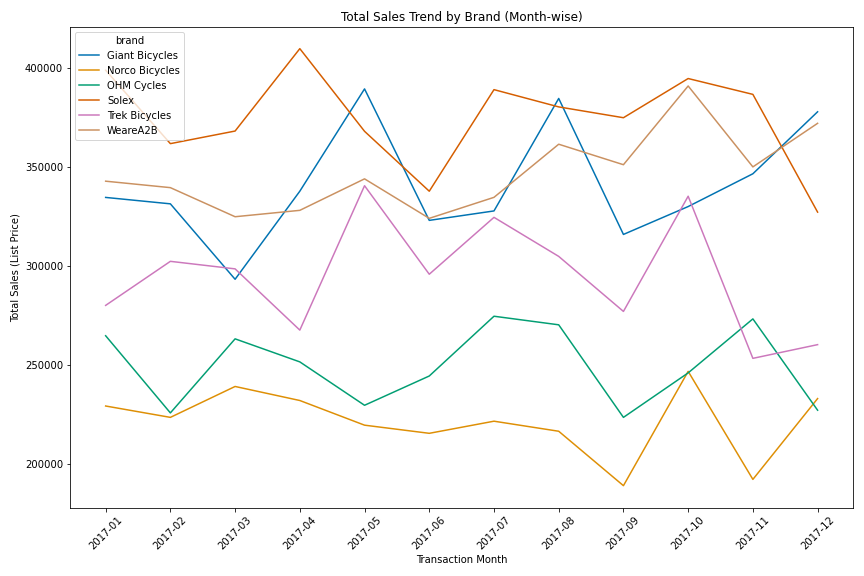
plt.ylabel('Total Sales (List Price)')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()





1. **Sales Performance Analysis**:
   * Compare the sales performance of different product lines and brands. Identify the top-performing and underperforming products based on revenue or quantity sold.
   * Analyze the sales performance of different product sizes and classes. Determine which size and class are the most popular and profitable.
2. **Customer Segmentation**:
   * Segment the customers based on their purchasing behavior, such as frequency of purchases, average spending, and product preferences. This can help in targeting specific customer groups with marketing strategies.
   * Analyze the differences in online and offline purchasing patterns. Determine if there are significant differences in the types of products purchased, order size, or customer demographics.
3. **Profitability Analysis**:
   * Calculate the profitability of each product by comparing the list price and standard cost. Identify the most and least profitable products.
   * Analyze the overall profitability trends over time and identify factors that influence profitability, such as changes in product mix or cost variations.
4. **Order Status Analysis**:
   * Analyze the distribution of order statuses (e.g., Approved, Cancelled) to identify patterns in order cancellations or returns. Investigate if certain products or times are more prone to order issues.
   * Explore the relationship between order status and other factors such as online vs. offline orders, product lines, or customer segments.
5. **Market Basket Analysis**:
   * Conduct a market basket analysis to identify products that are frequently bought together. This can help in creating product bundles or cross-promotional strategies.
   * Analyze the impact of product bundling on sales performance and customer purchasing behavior.
6. **Customer Feedback for Product Improvement**:
   * Assuming you have or can link this data with customer feedback or ratings, analyze the feedback to identify product strengths and areas for improvement.
   * Correlate customer satisfaction with product features such as brand, product line, or price to identify what factors most influence customer satisfaction.

These problem statements can be adapted or expanded based on specific business needs or additional data available (e.g., customer demographics, detailed sales channel information). They aim to help in understanding sales patterns, customer behavior, and product performance, providing actionable insights for strategic decision-making. ​

​