```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

transaction_df = pd.read_excel("C:/Users/ADMIN/Documents/My projects/Customer-purchase-behavior-/QVI_transaction_data.xlsx")

purchase_behavior_df = pd.read_csv("C:/Users/ADMIN/Documents/My projects/Customer-purchase-behavior-/QVI_purchase_behaviour.csv")

transaction_df.head()

purchase_behavior_df.head()

## Data cleaning

#Checking for missing values

print(transaction_df.isnull().sum())

print(purchase_behavior_df.isnull().sum())

#There are no missing values

#Checking for duplicates

print(transaction_df.duplicated().sum())

print(purchase_behavior_df.duplicated().sum())

duplicated_rows = transaction_df[transaction_df.duplicated()]

print(duplicated_rows)

#Checking for outliers

plt.figure(figsize=(15, 10))

# Loop through each numeric column and create a box plot

for i, column in enumerate(transaction_df.select_dtypes(include=['float64', 'int64']).columns):

    plt.subplot(len(transaction_df.select_dtypes(include=['float64', 'int64']).columns), 1, i + 1)

    sns.boxplot(x=transaction_df[column])

    plt.title(f'Box Plot of {column}')


plt.tight_layout()
```

```
plt.show()


#Checking for data types
print(transaction_df.dtypes)
print(purchase_behavior_df.dtypes)
print(transaction_df.describe())
print(purchase_behavior_df.describe())
print(transaction_df.info())
print(purchase_behavior_df.info())
#merging the datasets
df = transaction_df.merge(purchase_behavior_df, on='LYLTY_CARD_NBR')
df.head()
## Exploratory Data analysis
# Define Metrics and Explore Key Statistics
#Total sales
total_sales = df['TOT_SALES'].sum()
print(f"Total Sales: {total_sales}")
#Sales by LIFESTAGE and PREMIUM_CUSTOMER: Group data by LIFESTAGE and PREMIUM_CUSTOMER to analyze
#purchasing patterns.
sales_by_segment = df.groupby(['LIFESTAGE', 'PREMIUM_CUSTOMER'])['TOT_SALES'].sum().reset_index()
print(sales_by_segment)


df.columns
top_products = df.groupby('PROD_NAME')['TOT_SALES'].sum().sort_values(ascending=False).head(10)
print(top_products)


bottom_products = df.groupby('PROD_NAME')['TOT_SALES'].sum().sort_values(ascending=False).tail(10)
print(bottom_products)
```

```
# Data Visualization
plt.figure(figsize=(10, 6))
top_products.plot(kind='barh', color='skyblue')
plt.title('Top 10 Products by Sales')
plt.xlabel('Total Sales')
plt.ylabel('Product Name')
plt.gca().invert_yaxis()  # Flip for better readability
plt.show()

#Total Sales by Segment (LIFESTAGE and PREMIUM_CUSTOMER):
# This bar plot will help visualize total sales across different customer segments.
plt.figure(figsize=(12, 6))
sns.barplot(data=sales_by_segment, x='LIFESTAGE', y='TOT_SALES', hue='PREMIUM_CUSTOMER')
plt.title('Total Sales by Customer Segment')
plt.ylabel('Total Sales')
plt.xticks(rotation=45)
plt.legend(title='Premium Customer')
plt.show()

#Distribution of Transaction Quantity: Check if packet size correlates with customer segments.
plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='LIFESTAGE', y='PROD_QTY', hue='PREMIUM_CUSTOMER')
plt.title('Product Quantity Distribution by Customer Segment')
plt.ylabel('Quantity Sold')
plt.xticks(rotation=45)
plt.show()
```