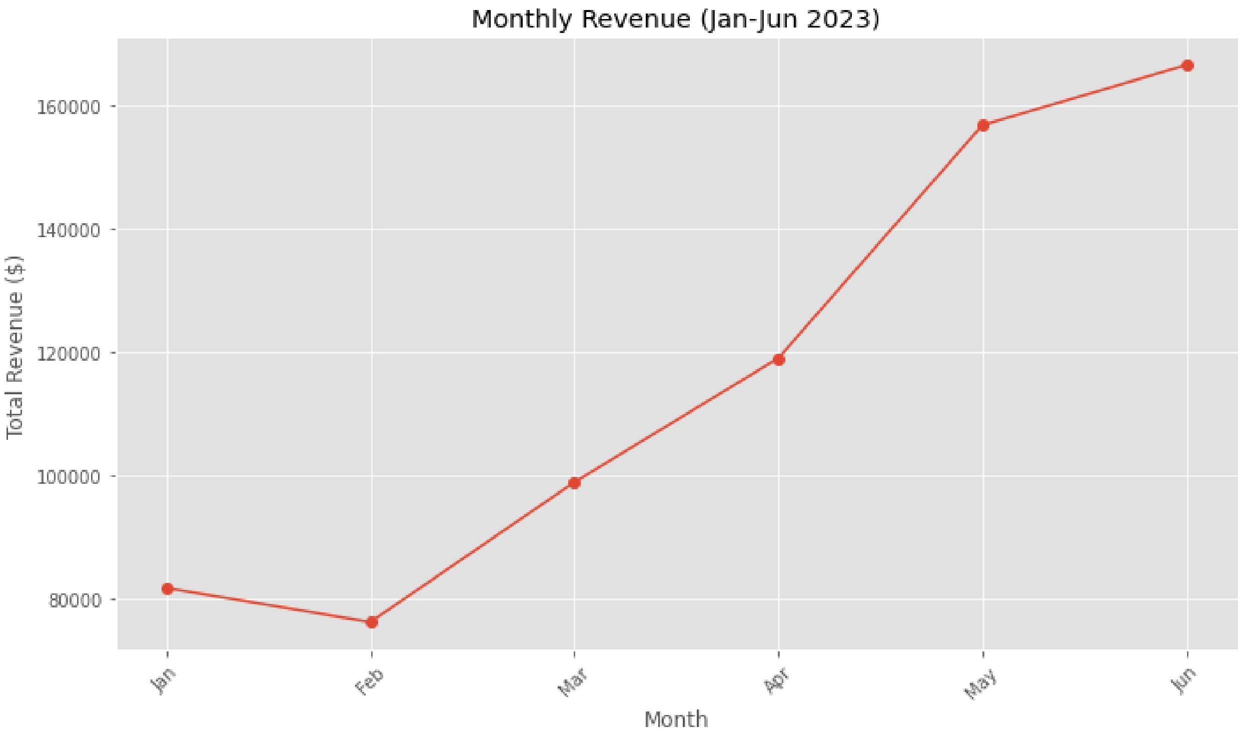
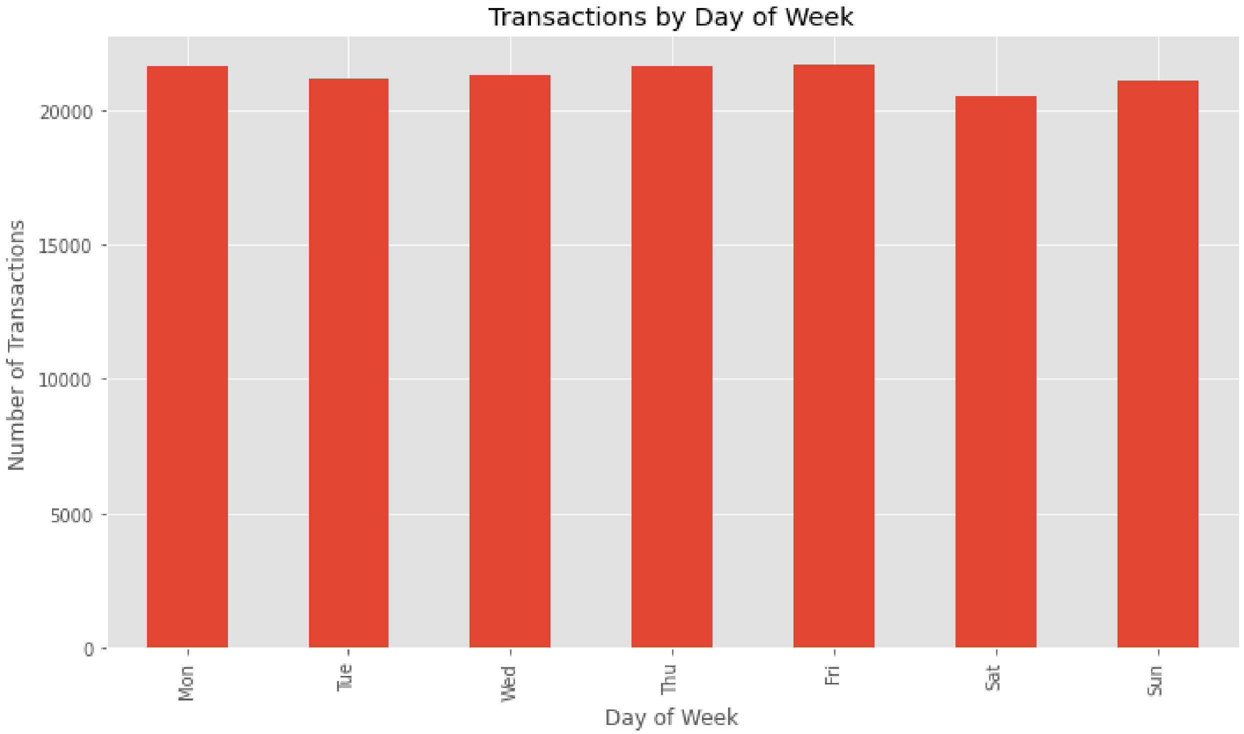
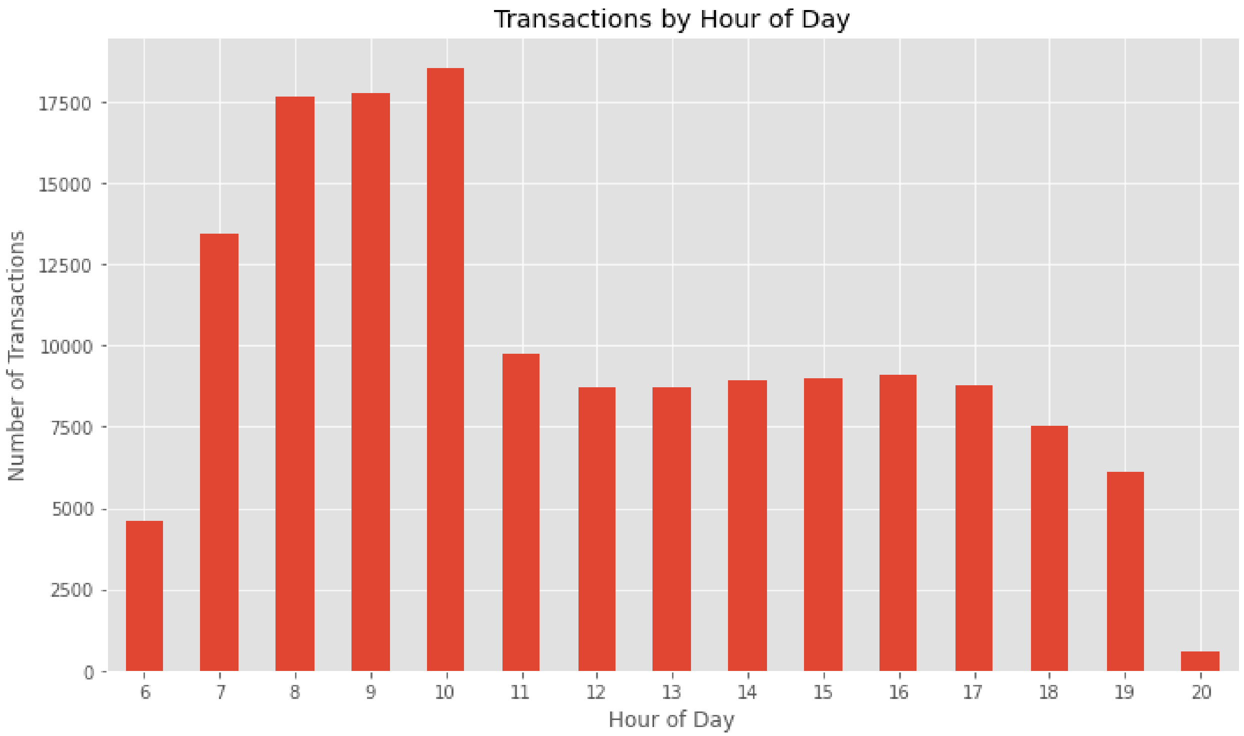
In [1]:

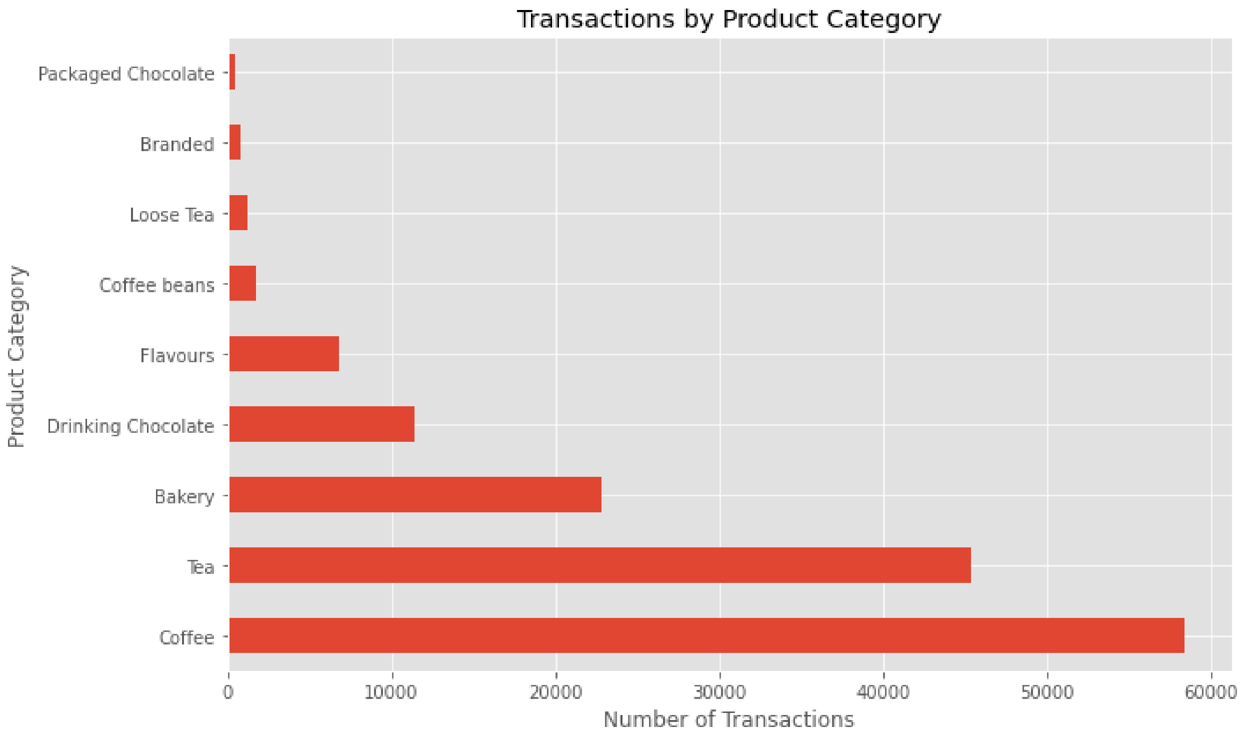
```
# Coffee Shop Sales Dashboard Project
# Objective: Analyze and visualize coffee shop sales data (Jan–Jun 2023)

# Importing required libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
file_path = 'Coffee Shop Sales.xlsx'
excel_data = pd.ExcelFile(file_path)
df = excel_data.parse('Transactions')

# Step 1: Data Preparation
# Add 'Revenue' column (unit_price * transaction_qty)
df['Revenue'] = df['unit_price'] * df['transaction_qty']

# Add 'Month' column (e.g., 'Jan', 'Feb')
df['Month'] = df['transaction_date'].dt.strftime('%b')

# Add 'Day_of_Week' column (e.g., 'Mon', 'Tue')
df['Day_of_Week'] = df['transaction_date'].dt.strftime('%a')

# Add 'Hour' column extracted from 'transaction_time'
df['Hour'] = pd.to_datetime(df['transaction_time'], format='%H:%M:%S').dt.hour

# Step 2: Data Exploration and Visualization
plt.style.use('ggplot')

# 1. Revenue by Month
monthly_revenue = df.groupby('Month')['Revenue'].sum().reindex(['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun'])
plt.figure(figsize=(10, 6))
monthly_revenue.plot(kind='line', marker='o')
plt.title('Monthly Revenue (Jan–Jun 2023)')
plt.xlabel('Month')
plt.ylabel('Total Revenue ($)')
plt.xticks(rotation=45)
plt.grid(True)
plt.tight_layout()
plt.show()

# 2. Transactions by Day of Week
transactions_by_day = df.groupby('Day_of_Week')['transaction_id'].count().reindex(['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun'])
plt.figure(figsize=(10, 6))
transactions_by_day.plot(kind='bar')
plt.title('Transactions by Day of Week')
plt.xlabel('Day of Week')
plt.ylabel('Number of Transactions')
```

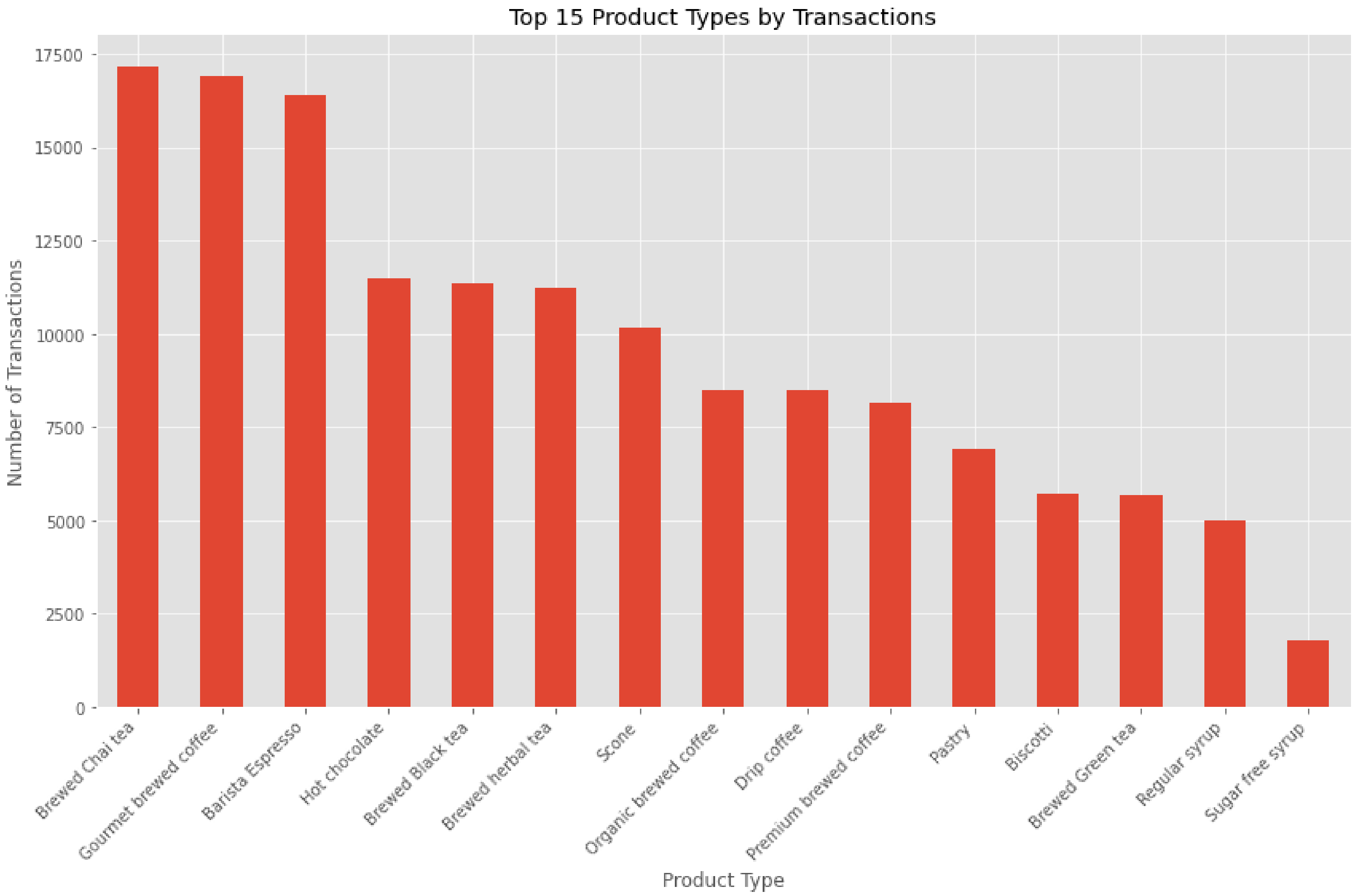
```python
plt.tight_layout()
plt.show()

# 3. Transactions by Hour of Day
transactions_by_hour = df.groupby('Hour')['transaction_id'].count()
plt.figure(figsize=(10, 6))
transactions_by_hour.plot(kind='bar')
plt.title('Transactions by Hour of Day')
plt.xlabel('Hour of Day')
plt.ylabel('Number of Transactions')
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()

# 4. Transactions by Product Category
product_category_transactions = df.groupby('product_category')['transaction_id'].count().sort_values(ascending=False)
plt.figure(figsize=(10, 6))
product_category_transactions.plot(kind='barh')
plt.title('Transactions by Product Category')
plt.xlabel('Number of Transactions')
plt.ylabel('Product Category')
plt.tight_layout()
plt.show()

# 5. Top 15 Product Types by Transactions
top_15_products = df.groupby('product_type')['transaction_id'].count().sort_values(ascending=False).head(15)
plt.figure(figsize=(12, 8))
top_15_products.plot(kind='bar')
plt.title('Top 15 Product Types by Transactions')
plt.xlabel('Product Type')
plt.ylabel('Number of Transactions')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

# Step 3: Final Project Question
# Count transactions on Wednesdays at Lower Manhattan
wednesday_lower_manhattan = df[(df['Day_of_Week'] == 'Wed') & (df['store_location'] == 'Lower Manhattan')]
wednesday_transaction_count = wednesday_lower_manhattan['transaction_id'].count()
print(f'Transactions on Wednesdays at Lower Manhattan: {wednesday_transaction_count}')
```


Monthly Revenue (Jan-Jun 2023)
Total Revenue ($)
160000
140000
120000
100000
80000
Jan
Feb
Mar
Apr
May
Jun
Month


Transactions by Day of Week
Number of Transactions
0
5000
10000
15000
20000
Mon
Tue
Wed
Thu
Fri
Sat
Sun
Day of Week


Transactions by Hour of Day
Number of Transactions
0
2500
5000
7500
10000
12500
15000
17500
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Hour of Day


Transactions by Product Category
Product Category
Packaged Chocolate
Branded
Loose Tea
Coffee beans
Flavours
Drinking Chocolate
Bakery
Tea
Coffee
0
10000
20000
30000
40000
50000
60000
Number of Transactions



```
Transactions on Wednesdays at Lower Manhattan: 6767
```