In [101…

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [102…

```
df = pd.read_csv('sales_data.csv') #load csv file
#test header
df.head(5)
```

Out[102…

	Date	Time	StoreID	CustomerID	OrderID	Product Name	Product Price
0	2024-04-09	15:35:25	460	619	10	Daves Killer Bread	6.85
1	2024-10-13	15:35:25	460	619	1	Goodfellow Grey T-shirt	19.99
2	2024-04-18	15:35:25	460	619	1	Alisan Kitchen Mats	29.99
3	2024-09-23	15:35:25	460	619	1	Driscolls Blueberries	5.99
4	2024-03-17	15:35:25	460	382	10	Driscolls Blueberries	5.99

In [103…

```
#Question 1
most_prevalent_pd = df['Product Name'].value_counts().head(1)

product_name = most_prevalent_pd.index[0]
product_count = most_prevalent_pd.values[0]

print(f"Most prevalent product: {product_name} with {product_count} sales.")
```

```
Most prevalent product: Driscolls Blueberries with 12641 sales.
```

In [129…

```
#Question 1
product_counts = df['Product Name'].value_counts()

most_prevalent_products = product_counts[product_counts == product_counts.max()]

print("Most Prevalent Product")
for product, count in most_prevalent_products.items():
    print(f"{product}: {count} sales")
```

```
Most Prevalent Product
Driscolls Blueberries: 12641 sales
```

In [105…

```
#Question 2
order_product_count = df.groupby(['CustomerID', 'OrderID'])['Product Name'].nunique

#5 or greater is considered "large"
large_basket_orders = order_product_count[order_product_count['Product Name'] >= 10

total_large_purchases = len(large_basket_orders)
```

```
print("\nFrequency of Large Baskets:", total_large_purchases, "occurrences")
```

```
Frequency of Large Baskets: 1186 occurrences
```

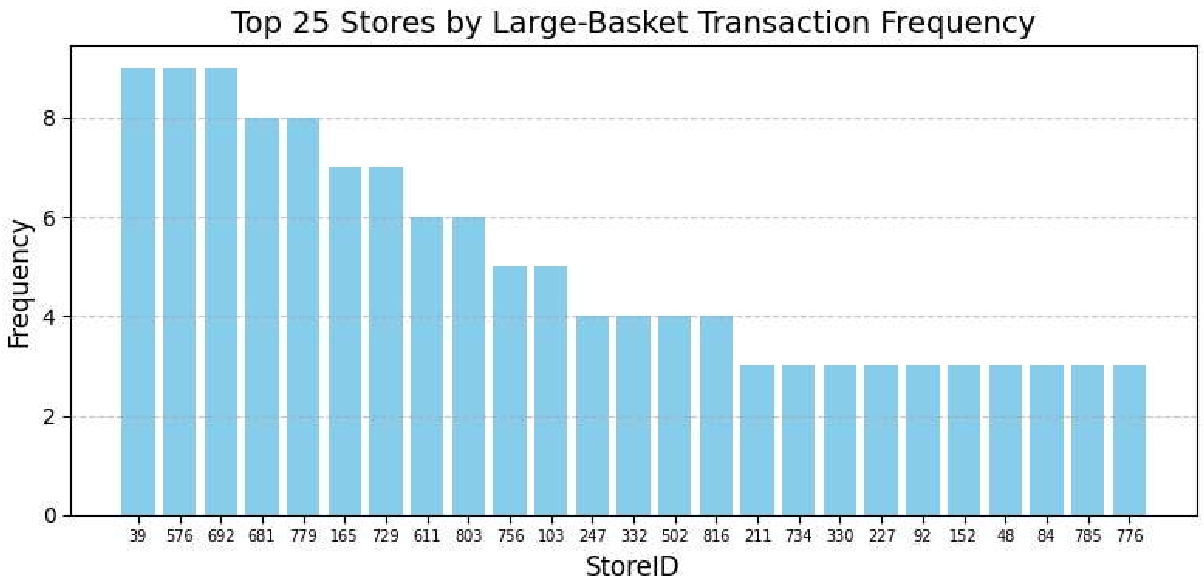
In [ ]:

```
#Question 3
order_product_count = df.groupby(['CustomerID', 'OrderID'])['Product Name'].nunique

large_basket_orders = order_product_count[order_product_count['Product Name'] >= 10

large_basket_orders = large_basket_orders.merge(df[['StoreID', 'OrderID']], on='Ord

store_large_basket_counts = large_basket_orders['StoreID'].value_counts()

total_stores_with_large_baskets = len(store_large_basket_counts)

print(f"Stores containing at least one large basket: {total_stores_with_large_baske

print("\n5 stores with the most filled-up baskets:")
for store_id, count in store_large_basket_counts.head(5).items():
    print(f"StoreID {store_id} had {count} large purchases")
```

```
Stores containing at least one large basket: 315

5 stores with the most filled-up baskets:
StoreID 576 had 156821 large purchases
StoreID 39 had 149701 large purchases
StoreID 681 had 141008 large purchases
StoreID 165 had 130106 large purchases
StoreID 692 had 128360 large purchases
```

In [128…

```
#Question 4
df['Product Price'] = pd.to_numeric(df['Product Price'], errors='coerce')
df.dropna(subset=['Product Price'], inplace=True)

basket_sizes = df.groupby(['StoreID', 'OrderID'])['Product Price'].sum().reset_inde

large_basket_threshold = basket_sizes['Product Price'].quantile(0.95)  # Top 5% of

large_basket_data = basket_sizes[basket_sizes['Product Price'] > large_basket_thres

top_stores = large_basket_data['StoreID'].value_counts().head(25)

plt.figure(figsize=(8, 4))
plt.bar(top_stores.index.astype(str), top_stores.values, color='skyblue')
plt.title('Top 25 Stores by Large-Basket Transaction Frequency', fontsize=14)
plt.xlabel('StoreID', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.xticks(fontsize=7)
plt.yticks(fontsize=10)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()

plt.show()
```



In [110…

```
#Question 5
df['ProductCount'] = df.groupby('OrderID')['Product Name'].transform('nunique')
large_basket_orders = df[df['ProductCount'] >= 5]

product_counts_in_large_baskets = large_basket_orders['Product Name'].value_counts(
top_n_products_large_basket = product_counts_in_large_baskets.head(5)

print("Top 5 products linked to large basket shoppers:")

for rank, (product, count) in enumerate(top_n_products_large_basket.items(), start=
    print(f"{rank}. {product}, Sold {count} times")
```

```
Top 5 products linked to large basket shoppers:
1. Driscolls Blueberries, Sold 12641 times
2. Goodfellow Grey T-shirt, Sold 12621 times
3. Afflux Type-C, Sold 12575 times
4. Daves Killer Bread, Sold 12564 times
5. Organic 2% Milk, Sold 12531 times
```

In [115…

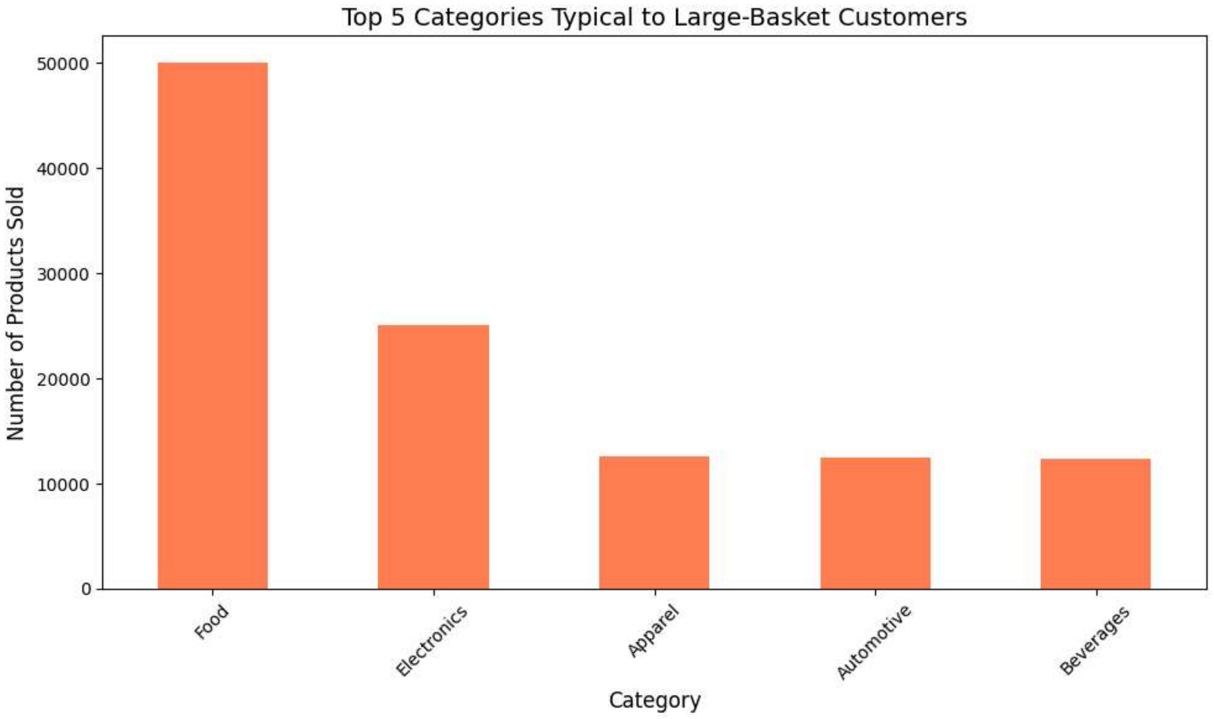
```
#Question 6
category_map = {
    "Driscolls Blueberries": "Food",
    "Alisan Kitchen Mats": "Home",
    "Organic 2% Milk": "Food",
    "Goodfellow Grey T-shirt": "Apparel",
    "Apple AirPods Pro": "Electronics",
    "Tropicana Orange Juice": "Beverages",
    "Toll House Cookie Dough": "Food",
    "Daves Killer Bread": "Food",
    "Afflux Type-C": "Electronics",
    "Mobil 1 5W30 Oil": "Automotive",
}

df['Category'] = df['Product Name'].map(category_map)
large_basket_orders = df[df['ProductCount'] >= 10]
category_counts_in_large_baskets = large_basket_orders['Category'].value_counts()
top_5_categories_large_basket = category_counts_in_large_baskets.head(5)
```

```
print("Top 5 categories typical to large-basket customers:")
for i, (category, count) in enumerate(top_5_categories_large_basket.items(), start=
    print(f"{i}: {category}, Sold {count} times")
```

```
Top 5 categories typical to large-basket customers:
1: Food, Sold 50054 times
2: Electronics, Sold 25094 times
3: Apparel, Sold 12621 times
4: Automotive, Sold 12458 times
5: Beverages, Sold 12399 times
```

In [126…

```
#Question 7
plt.figure(figsize=(10, 6))
top_5_categories_large_basket.plot(kind='bar', color='coral')

plt.title("Top 5 Categories Typical to Large-Basket Customers", fontsize=14)
plt.xlabel("Category", fontsize=12)
plt.ylabel("Number of Products Sold", fontsize=12)

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```


Top 5 Categories Typical to Large-Basket Customers
Number of Products Sold
Food
Electronics
Apparel
Automotive
Beverages
Category