

Project 2 Section

```
import pandas as pd

df = pd.read_csv('sales_data.csv')

print(df.head(10))
```

 Show hidden output

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

# Load the sales data
file_path = 'sales_data.csv'
sales_data = pd.read_csv(file_path)
```

```
# Display the first few rows of the dataframe
print(sales_data.head())
```

 Show hidden output

```
# 1. The most prevalent products in customer baskets
prevalent_products = sales_data['Product Name'].value_counts()
print("Most prevalent products in customer baskets:")
print(prevalent_products)
```

 Show hidden output

```
# 2. The frequency by which customers were large buyers or filled up large baskets
# Assuming large basket is defined as having more than 3 items in a single order
large_basket_orders = sales_data.groupby('OrderID').size()
large_basket_frequency = large_basket_orders[large_basket_orders > 3].count()
print(f"Frequency of large basket orders: {large_basket_frequency}")
```

 Show hidden output

```
# 3. Which stores contained the large-basket buyers and by how much
large_basket_store_counts = sales_data[sales_data['OrderID'].isin(large_basket_orders[large_basket_orders > 3].index)][ 'StoreID'].value_counts()
print("Stores containing large-basket buyers and their counts:")
print(large_basket_store_counts)
```

 Show hidden output

```
# 4. A visualization that ranks the top large-basket customer stores by frequency
plt.figure(figsize=(10, 6))
if not large_basket_store_counts.empty:
    large_basket_store_counts.plot(kind='bar')
    plt.title('Top Large-Basket Customer Stores by Frequency')
    plt.xlabel('Store ID')
    plt.ylabel('Frequency')
    plt.xticks(rotation=45)
    plt.show()
else:
    print("No large-basket orders found to plot.")
```

 Show hidden output

```
# 5. A top-n list of products which were typical to customers in this demographic
top_n_products = prevalent_products.head(10)
print("Top-N list of products typical to customers in this demographic:")
print(top_n_products)
```

 Show hidden output

```
# 6. A categorical approach to the above demographic - what is the categoric makeup of their baskets on average?
categoric_makeup = sales_data[sales_data['OrderID'].isin(large_basket_orders[large_basket_orders > 3].index)][ 'Product Name'].value_counts(n
```

```
print("Categoric makeup of their baskets on average:")
print(categoric_makeup)
```

 Show hidden output


```
# 7. Formulate a visualization for item 6
plt.figure(figsize=(10, 6))
if not categoric_makeup.empty:
    categoric_makeup.plot(kind='bar')
    plt.title('Categoric Makeup of Large-Basket Orders')
    plt.xlabel('Product Name')
    plt.ylabel('Proportion')
    plt.xticks(rotation=45)
    plt.show()
else:
    print("No large-basket orders found to plot categoric makeup.")
```

 Show hidden output

Project 3 Section

1. Most Prevalent Products in Customer Baskets

```
prevalent_products = sales_data['Product Name'].value_counts()
print("Most prevalent products in customer baskets:")
print(prevalent_products)
```

 Most prevalent products in customer baskets:

Product Name	
Mouse	218
Monitor	212
Keyboard	200
Laptop	192
Printer	177

Name: count, dtype: int64


2. Frequency of Large Basket Orders

```
large_basket_orders = sales_data.groupby('OrderID').size()
large_basket_frequency = large_basket_orders[large_basket_orders > 3].count()
print(f"Frequency of large basket orders: {large_basket_frequency}")
```

 Frequency of large basket orders: 0

3. Stores Containing Large-Basket Buyers

```
large_basket_store_counts = sales_data[sales_data['OrderID'].isin(large_basket_orders[large_basket_orders > 3].index)][ 'StoreID'].value_counts()
print("Stores containing large-basket buyers and their counts:")
print(large_basket_store_counts)
```

 Stores containing large-basket buyers and their counts:

Series([], Name: count, dtype: int64)

4. Visualization of Top Large-Basket Customer Stores by Frequency

```
plt.figure(figsize=(10, 6))
if not large_basket_store_counts.empty:
    large_basket_store_counts.plot(kind='bar')
    plt.title('Top Large-Basket Customer Stores by Frequency')
    plt.xlabel('Store ID')
    plt.ylabel('Frequency')
    plt.xticks(rotation=45)
    plt.show()
else:
    print("No large-basket orders found to plot.")
```

 No large-basket orders found to plot.
<Figure size 1000x600 with 0 Axes>

5. Top-N List of Products Typical to Customers

```
top_n_products = prevalent_products.head(10)
print("Top-N list of products typical to customers in this demographic:")
print(top_n_products)
```

```
Top-N list of products typical to customers in this demographic:
Product Name
Mouse      218
Monitor    212
Keyboard   200
Laptop     192
Printer    177
Name: count, dtype: int64
```

6. Categorical Makeup of Their Baskets on Average

```
categoric_makeup = sales_data[sales_data['OrderID'].isin(large_basket_orders[large_basket_orders > 3].index)][['Product Name']].value_counts(n
print("Categoric makeup of their baskets on average:")
print(categoric_makeup)
```

```
Categoric makeup of their baskets on average:
Series([], Name: proportion, dtype: float64)
```

7. Visualization for Categorical Makeup

```
plt.figure(figsize=(10, 6))
if not categoric_makeup.empty:
    categoric_makeup.plot(kind='bar')
    plt.title('Categoric Makeup of Large-Basket Orders')
    plt.xlabel('Product Name')
    plt.ylabel('Proportion')
    plt.xticks(rotation=45)
    plt.show()
else:
    print("No large-basket orders found to plot categoric makeup.")
```

```
No large-basket orders found to plot categoric makeup.
<Figure size 1000x600 with 0 Axes>
```