problem-1-1

March 17, 2024

Importing of the library

[ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Laoding of the data set

[ ]:
```
df=pd.read_csv("/content/shopping_trends_updated.csv")
```

[ ]:
```
df.head()
```

[ ]:

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	\
0	1	55	Male	Blouse	Clothing	53	
1	2	19	Male	Sweater	Clothing	64	
2	3	50	Male	Jeans	Clothing	73	
3	4	21	Male	Sandals	Footwear	90	
4	5	45	Male	Blouse	Clothing	49	

	Location	Size	Color	Season	Review Rating	Subscription Status	\
0	Kentucky	L	Gray	Winter	3.1	Yes	
1	Maine	L	Maroon	Winter	3.1	Yes	
2	Massachusetts	S	Maroon	Spring	3.1	Yes	
3	Rhode Island	M	Maroon	Spring	3.5	Yes	
4	Oregon	M	Turquoise	Spring	2.7	Yes	

	Shipping Type	Discount Applied	Promo Code Used	Previous Purchases	\
0	Express	Yes	Yes	14	
1	Express	Yes	Yes	2	
2	Free Shipping	Yes	Yes	23	
3	Next Day Air	Yes	Yes	49	
4	Free Shipping	Yes	Yes	31	

	Payment Method	Frequency of Purchases
0	Venmo	Fortnightly
1	Cash	Fortnightly
2	Credit Card	Weekly
3	PayPal	Weekly

```
4         PayPal                Annually
```

[ ]:
```
df.describe()
```

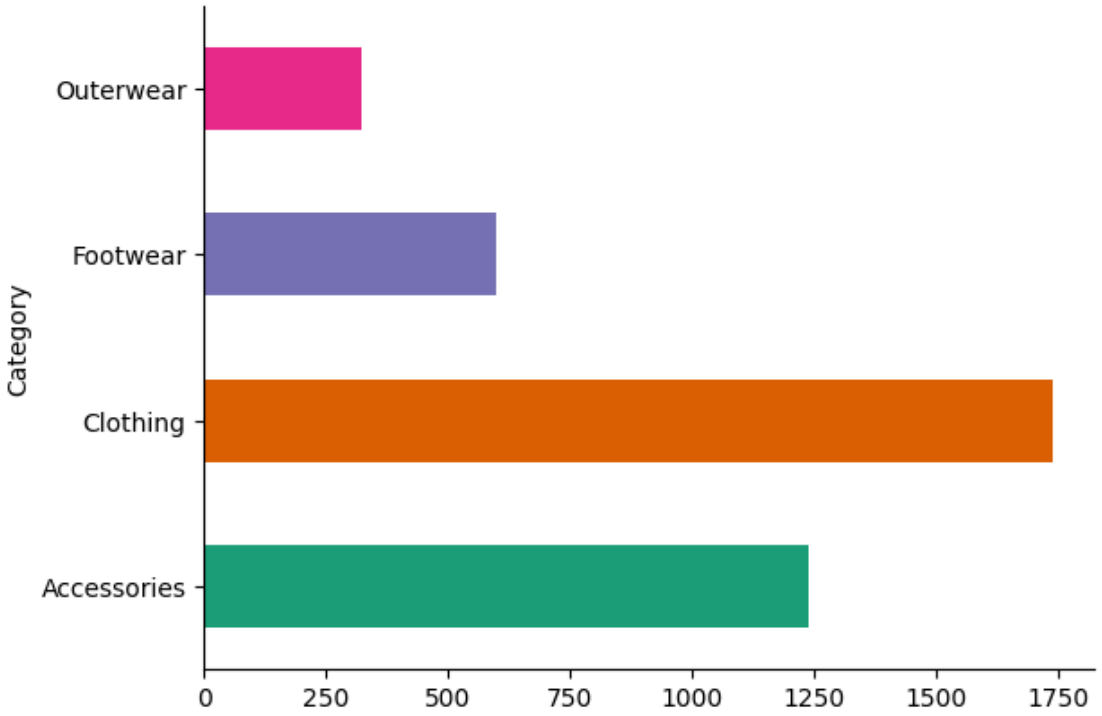
[ ]:
```
       Customer ID          Age  Purchase Amount (USD)  Review Rating  \
count  3900.000000  3900.000000            3900.000000    3900.000000
mean   1950.500000    44.068462              59.764359       3.749949
std    1125.977353    15.207589              23.685392       0.716223
min       1.000000    18.000000              20.000000       2.500000
25%     975.750000    31.000000              39.000000       3.100000
50%    1950.500000    44.000000              60.000000       3.700000
75%    2925.250000    57.000000              81.000000       4.400000
max    3900.000000    70.000000             100.000000       5.000000

       Previous Purchases
count         3900.000000
mean            25.351538
std             14.447125
min              1.000000
25%             13.000000
50%             25.000000
75%             38.000000
max             50.000000
```

[ ]:
```
df.info()
```

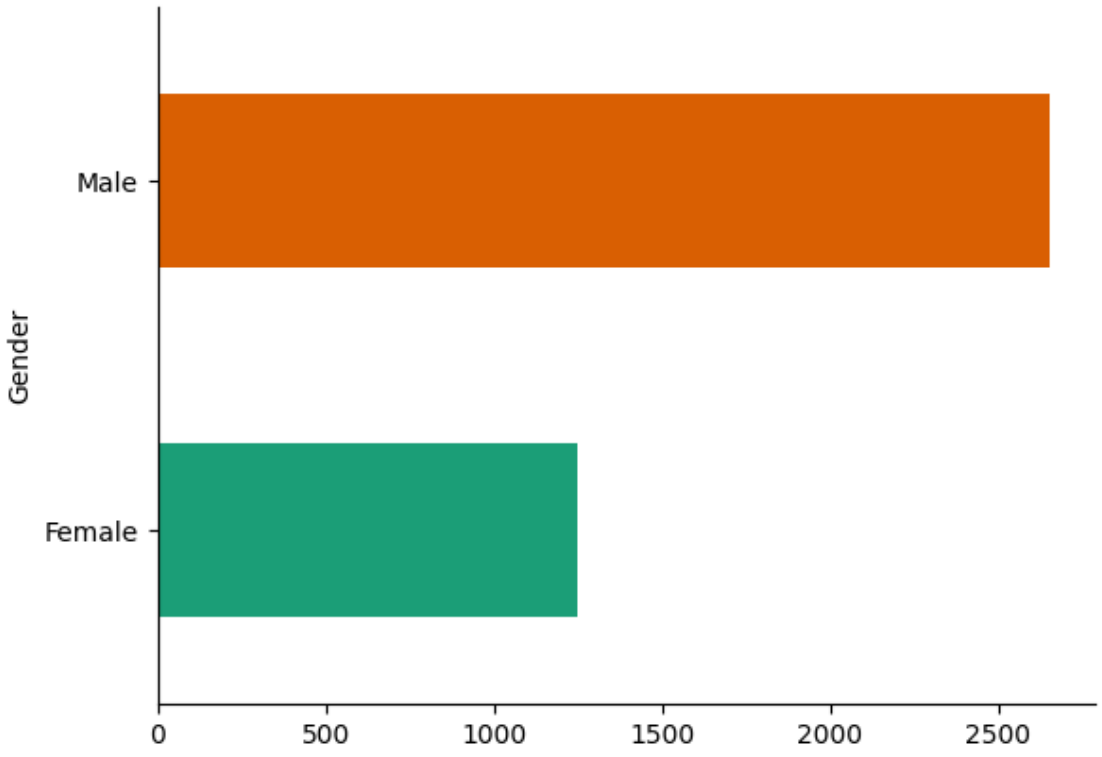
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3900 entries, 0 to 3899
Data columns (total 18 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Customer ID             3900 non-null   int64
 1   Age                     3900 non-null   int64
 2   Gender                  3900 non-null   object
 3   Item Purchased          3900 non-null   object
 4   Category                3900 non-null   object
 5   Purchase Amount (USD)   3900 non-null   int64
 6   Location                3900 non-null   object
 7   Size                    3900 non-null   object
 8   Color                   3900 non-null   object
 9   Season                  3900 non-null   object
 10  Review Rating           3900 non-null   float64
 11  Subscription Status     3900 non-null   object
 12  Shipping Type           3900 non-null   object
 13  Discount Applied        3900 non-null   object
 14  Promo Code Used         3900 non-null   object
 15  Previous Purchases      3900 non-null   int64
```

```
 16  Payment Method          3900 non-null   object
 17  Frequency of Purchases  3900 non-null   object
dtypes: float64(1), int64(4), object(13)
memory usage: 548.6+ KB
```

[ ]:
```
# @title Category

df.groupby('Category').size().plot(kind='barh', color=sns.palettes.
  ↪mpl_palette('Dark2'))
plt.gca().spines[['top', 'right',]].set_visible(False)
```



[ ]:
```
# @title Gender

df.groupby('Gender').size().plot(kind='barh', color=sns.palettes.
  ↪mpl_palette('Dark2'))
plt.gca().spines[['top', 'right',]].set_visible(False)
```



[ ]:
```
df['Item Purchased'].value_counts()
```

[ ]:
```
Blouse        171
Jewelry       171
Pants         171
Shirt         169
Dress         166
Sweater       164
Jacket        163
Belt          161
Sunglasses    161
Coat          161
Sandals       160
Socks         159
Skirt         158
Shorts        157
Scarf         157
Hat           154
Handbag       153
Hoodie        151
Shoes         150
T-shirt       147
```

```
Sneakers      145
Boots         144
Backpack      143
Gloves        140
Jeans         124
Name: Item Purchased, dtype: int64
```

[ ]:
```
df['Category'].value_counts()
```

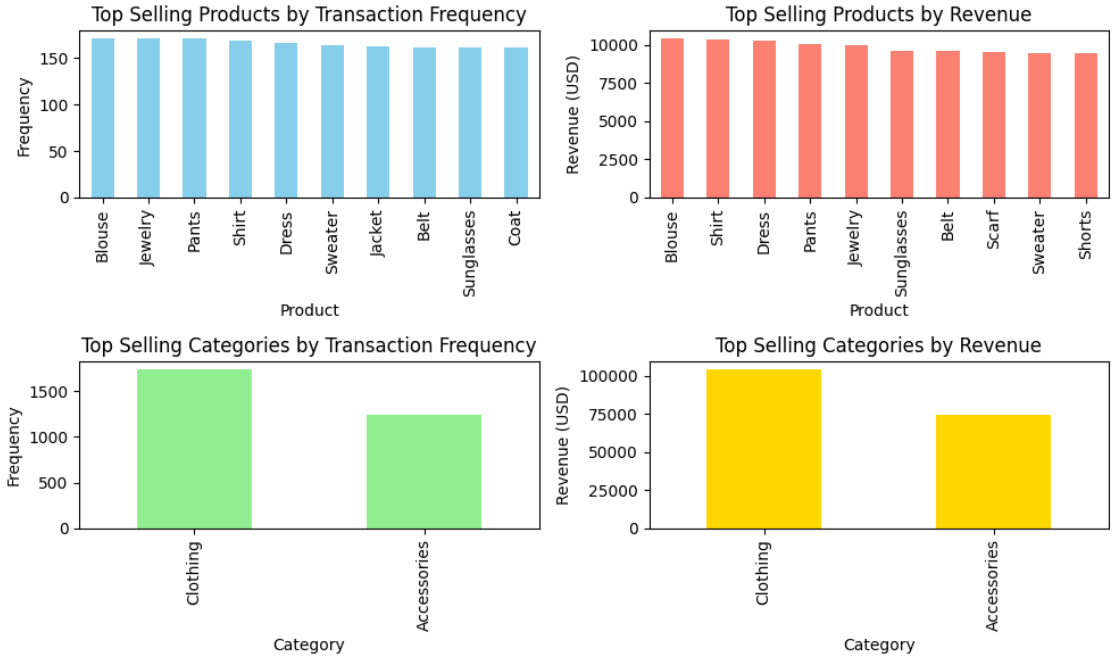
[ ]:
```
Clothing       1737
Accessories    1240
Footwear        599
Outerwear       324
Name: Category, dtype: int64
```

Exploratory Data Analysis

Top-Selling Products and Categories

[ ]:
```
# Calculate transaction frequency by product and category
transaction_freq_product = df['Item Purchased'].value_counts()
transaction_freq_category = df['Category'].value_counts()

# Calculate revenue by product and category
revenue_product = df.groupby('Item Purchased')['Purchase Amount (USD)'].sum()
revenue_category = df.groupby('Category')['Purchase Amount (USD)'].sum()

# Identify top-selling products and categories
top_products_by_freq = transaction_freq_product.head(10)  # Top 10 products by␣
 ↪transaction frequency
top_products_by_revenue = revenue_product.nlargest(10)    # Top 10 products by␣
 ↪revenue
top_categories_by_freq = transaction_freq_category.head(2) # Top 2 categories␣
 ↪by transaction frequency
top_categories_by_revenue = revenue_category.nlargest(2)   # Top 2 categories␣
 ↪by revenue

# Visualize the results
plt.figure(figsize=(10, 6))

plt.subplot(2, 2, 1)
top_products_by_freq.plot(kind='bar', color='skyblue')
plt.title('Top Selling Products by Transaction Frequency')
plt.xlabel('Product')
plt.ylabel('Frequency')

plt.subplot(2, 2, 2)
```

```
top_products_by_revenue.plot(kind='bar', color='salmon')
plt.title('Top Selling Products by Revenue')
plt.xlabel('Product')
plt.ylabel('Revenue (USD)')

plt.subplot(2, 2, 3)
top_categories_by_freq.plot(kind='bar', color='lightgreen')
plt.title('Top Selling Categories by Transaction Frequency')
plt.xlabel('Category')
plt.ylabel('Frequency')

plt.subplot(2, 2, 4)
top_categories_by_revenue.plot(kind='bar', color='gold')
plt.title('Top Selling Categories by Revenue')
plt.xlabel('Category')
plt.ylabel('Revenue (USD)')

plt.tight_layout()
plt.show()

# Analyze insights
print("Top Selling Products by Transaction Frequency:")
print(top_products_by_freq)
print("\nTop Selling Products by Revenue:")
print(top_products_by_revenue)
print("\nTop Selling Categories by Transaction Frequency:")
print(top_categories_by_freq)
print("\nTop Selling Categories by Revenue:")
print(top_categories_by_revenue)
```



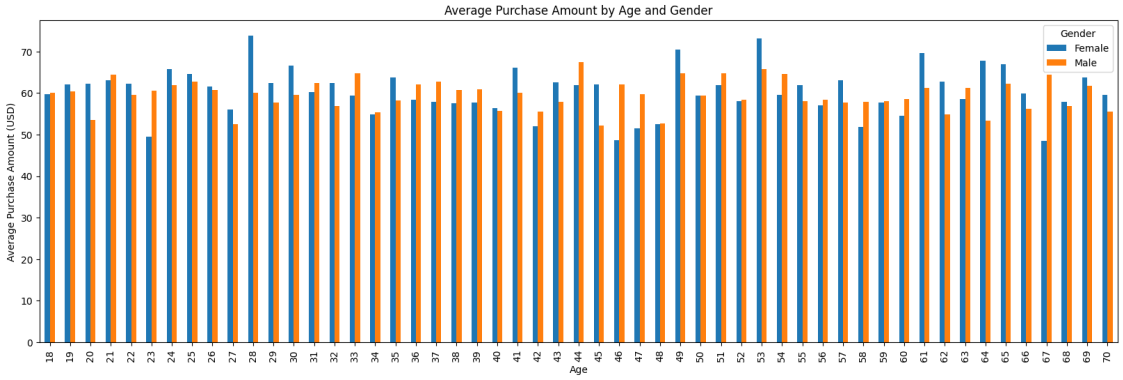

```
Top Selling Products by Transaction Frequency:
Blouse        171
Jewelry       171
Pants         171
Shirt         169
Dress         166
Sweater       164
Jacket        163
Belt          161
Sunglasses    161
Coat          161
Name: Item Purchased, dtype: int64

Top Selling Products by Revenue:
Item Purchased
Blouse        10410
Shirt         10332
Dress         10320
Pants         10090
Jewelry       10010
Sunglasses     9649
Belt           9635
Scarf          9561
Sweater        9462
Shorts         9433
Name: Purchase Amount (USD), dtype: int64
```

```
Top Selling Categories by Transaction Frequency:
Clothing       1737
Accessories    1240
Name: Category, dtype: int64

Top Selling Categories by Revenue:
Category
Clothing       104264
Accessories     74200
Name: Purchase Amount (USD), dtype: int64
```

Analyzing purchasing behavior

[ ]:
```
grouped_data = df.groupby(['Age', 'Gender'])


average_purchase_amount = grouped_data['Purchase Amount (USD)'].mean()
purchase_frequency = grouped_data['Frequency of Purchases'].value_counts()
popular_categories = grouped_data['Category'].value_counts()
```

[ ]:
```
# Average Purchase Amount by Age and Gender
average_purchase_amount.unstack().plot(kind='bar', figsize=(20, 6))
plt.title('Average Purchase Amount by Age and Gender')
plt.xlabel('Age')
plt.ylabel('Average Purchase Amount (USD)')
plt.legend(title='Gender')
plt.show()
```



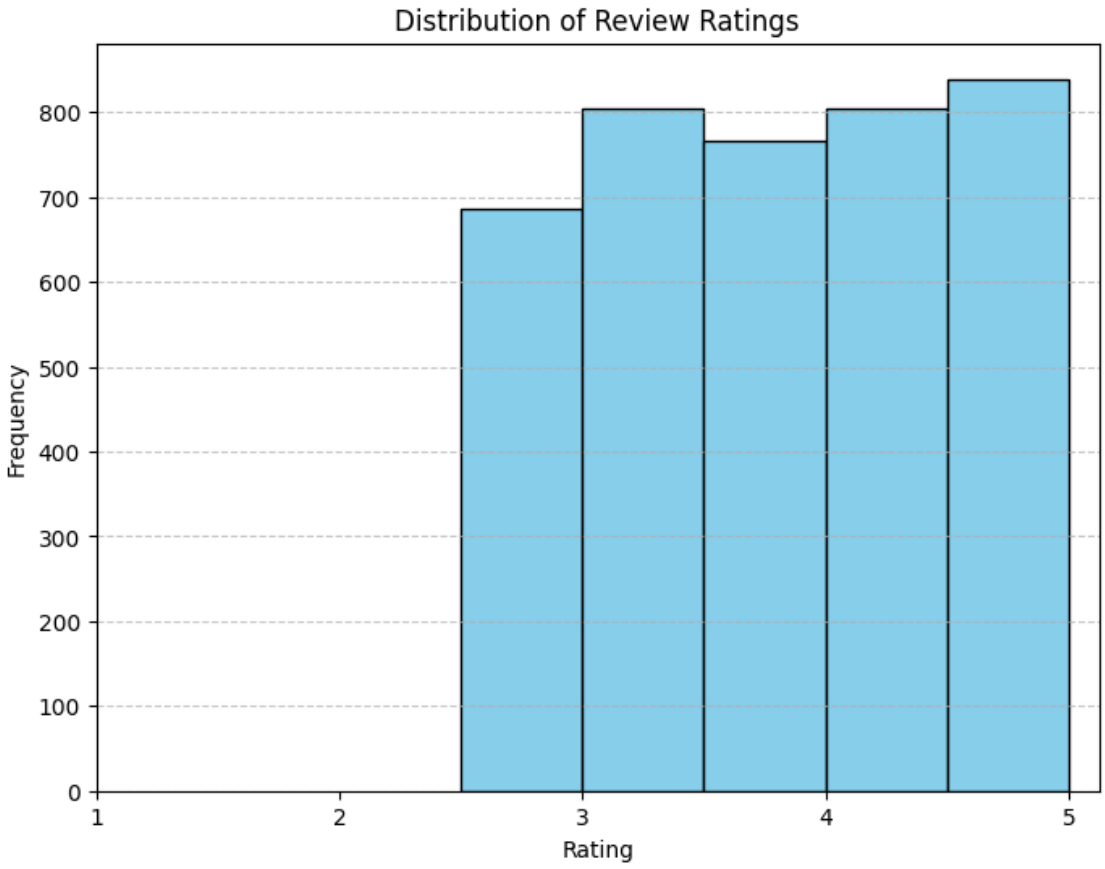
Preferred payment methods

[ ]:
```
df['Payment Method'].value_counts()/len(df)*100
```

[ ]:
```
PayPal            17.358974
Credit Card       17.205128
Cash              17.179487
Debit Card        16.307692
Venmo             16.256410
Bank Transfer     15.692308
Name: Payment Method, dtype: float64
```

Customer Ratings and Product Satisfaction:

[ ]:
```
# Aggregate review ratings by product
product_ratings = df.groupby('Item Purchased')['Review Rating'].mean()

# Visualize ratings distribution
plt.figure(figsize=(8, 6))
plt.hist(df['Review Rating'], bins=5, color='skyblue', edgecolor='black')
plt.title('Distribution of Review Ratings')
plt.xlabel('Rating')
plt.ylabel('Frequency')
plt.xticks(range(1, 6))
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



Improvement Areas(Rating Less than 3.75)

[ ]:
```
low_rated_products = product_ratings[product_ratings <3.75]

# Print low-rated products
print("Low Rated Products:")
print(low_rated_products)
```

```
Low Rated Products:
Item Purchased
Blouse        3.683626
Coat          3.730435
Hoodie        3.719205
Jeans         3.648387
Pants         3.718713
Scarf         3.700000
Shirt         3.629586
Shoes         3.747333
Shorts        3.711465
Sunglasses    3.744720
Name: Review Rating, dtype: float64
```

Impact of Discounts or Promotions

[ ]:
```
discount_promo_used_data = df[(df['Discount Applied'] == 'Yes') & (df['Promo␣
↪Code Used'] == 'Yes')]

# Count the number of unique customers
num_customers = discount_promo_used_data['Customer ID'].nunique()

print("Number of customers who have purchased with discount applied and promo␣
↪code used:", num_customers)
print("Percentage of customers who have purchased with discount applied and␣
↪promo code used: ",num_customers/len(df)*100,"%")
```

```
Number of customers who have purchased with discount applied and promo code
used: 1677
Percentage of customers who have purchased with discount applied and promo code
used:  43.0 %
```

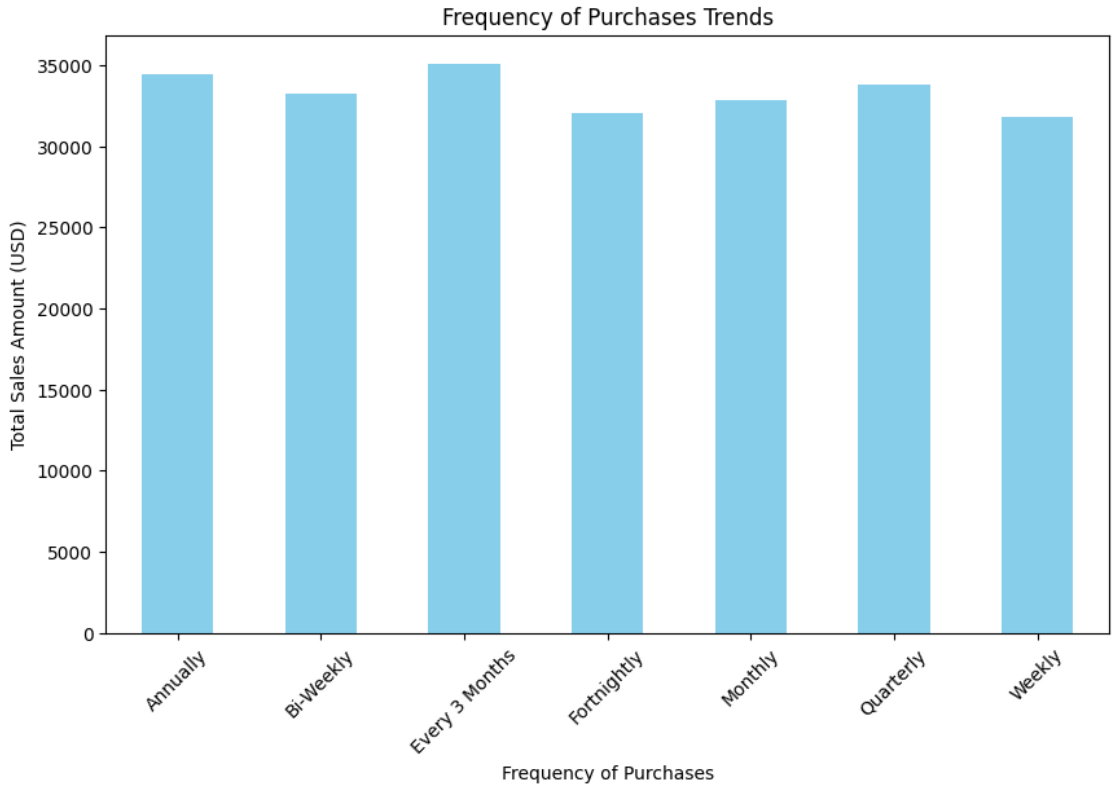
[ ]:
```
# Filter data for transactions with and without discounts applied or promo␣
↪codes used
discount_promo_applied_data = df[(df['Discount Applied'] == 'Yes') | (df['Promo␣
↪Code Used'] == 'Yes')]
no_discount_promo_data = df[(df['Discount Applied'] == 'No') & (df['Promo Code␣
↪Used'] == 'No')]
```

```
# Calculate metrics for transactions with discounts applied or promo codes used
discount_promo_sales_amount = discount_promo_applied_data['Purchase Amount␣
 ↪(USD)'].sum()
discount_promo_num_transactions = discount_promo_applied_data.shape[0]
discount_promo_avg_purchase_amount = discount_promo_sales_amount /␣
 ↪discount_promo_num_transactions

# Calculate metrics for transactions without discounts applied or promo codes␣
 ↪used
no_discount_promo_sales_amount = no_discount_promo_data['Purchase Amount␣
 ↪(USD)'].sum()
no_discount_promo_num_transactions = no_discount_promo_data.shape[0]
no_discount_promo_avg_purchase_amount = no_discount_promo_sales_amount /␣
 ↪no_discount_promo_num_transactions

# Visualize impact
labels = ['With Discount/Promo Applied', 'Without Discount/Promo Applied']
sales_amounts = [discount_promo_sales_amount, no_discount_promo_sales_amount]
num_transactions = [discount_promo_num_transactions,␣
 ↪no_discount_promo_num_transactions]
avg_purchase_amounts = [discount_promo_avg_purchase_amount,␣
 ↪no_discount_promo_avg_purchase_amount]

plt.figure(figsize=(10, 6))

plt.subplot(2, 1, 1)
plt.bar(labels, sales_amounts, color=['blue', 'green'])
plt.title('Total Sales Amount')
plt.ylabel('Sales Amount (USD)')

plt.subplot(2, 1, 2)
plt.bar(labels, num_transactions, color=['blue', 'green'])
plt.title('Number of Transactions')
plt.ylabel('Number of Transactions')

plt.tight_layout()
plt.show()

print("With Discount/Promo Applied:", discount_promo_avg_purchase_amount)
print("Without Discount/Promo Applied:", no_discount_promo_avg_purchase_amount)
```

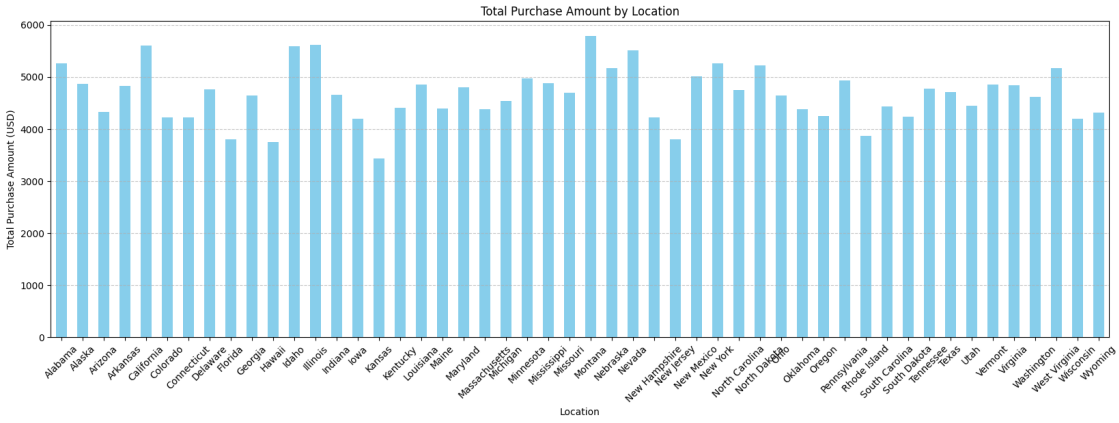

```
With Discount/Promo Applied: 59.27906976744186
Without Discount/Promo Applied: 60.130454340980656
```

Seasonality of Product Purchases:

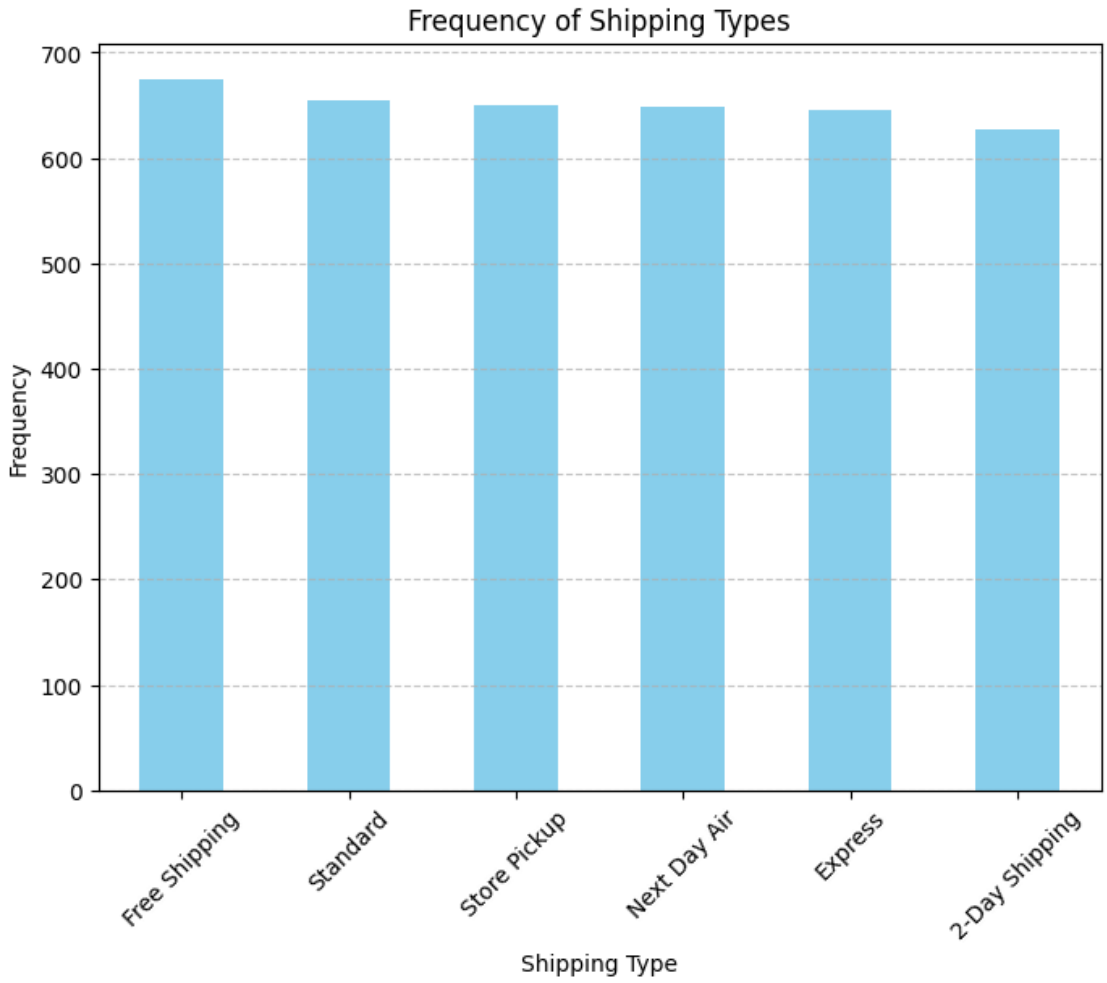
[ ]:
```
frequency_sales = df.groupby('Frequency of Purchases')['Purchase Amount (USD)'].
↪sum()

# Visualize frequency of purchases trends
plt.figure(figsize=(10, 6))
frequency_sales.plot(kind='bar', color='skyblue')
plt.title('Frequency of Purchases Trends')
plt.xlabel('Frequency of Purchases')
plt.ylabel('Total Sales Amount (USD)')
plt.xticks(rotation=45)
plt.show()
```



[ ]:
```
# Group data by location and calculate total purchase amount for each location
total_purchase_by_location = df.groupby('Location')['Purchase Amount (USD)'].
  ↪sum()

# Plot total purchase amount by location
plt.figure(figsize=(20, 6))
total_purchase_by_location.plot(kind='bar', color='skyblue')
plt.title('Total Purchase Amount by Location')
plt.xlabel('Location')
plt.ylabel('Total Purchase Amount (USD)')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



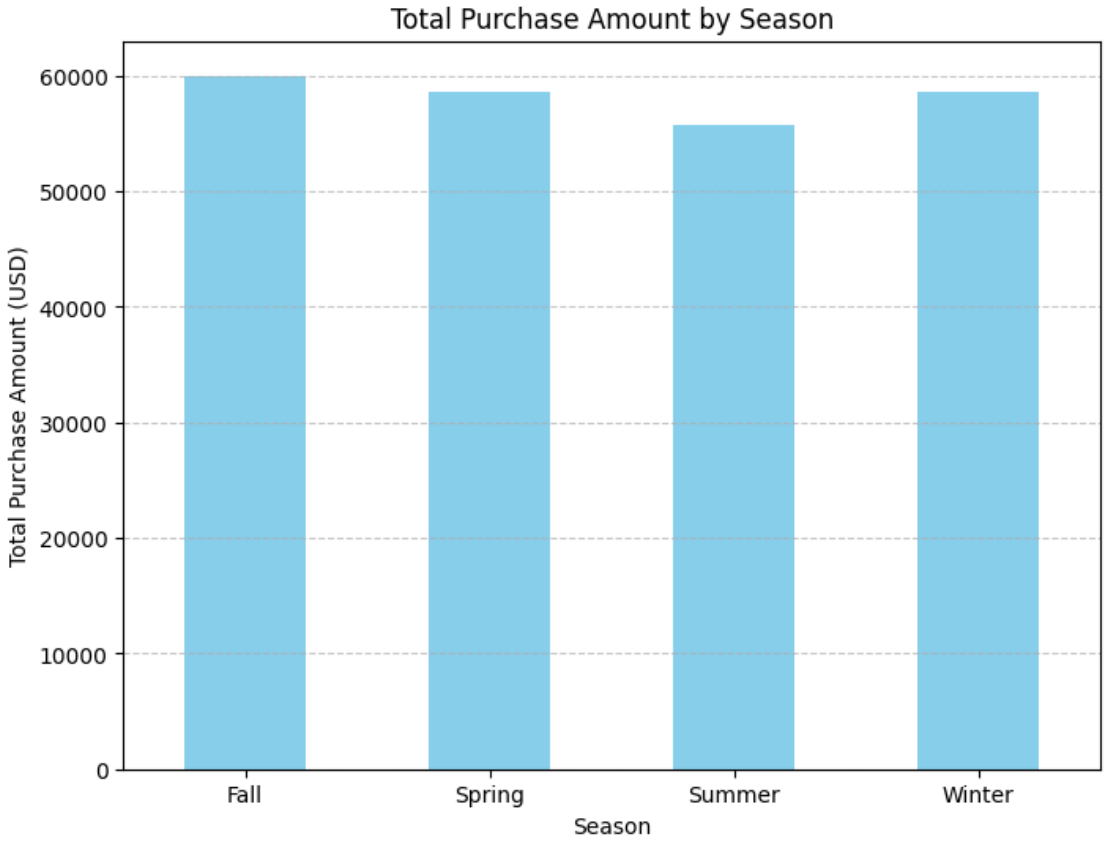
[ ]:
```python
# Group data by shipping type and count the frequency of each type
shipping_type_counts = df['Shipping Type'].value_counts()

# Plot the frequency of each shipping type
plt.figure(figsize=(8, 6))
shipping_type_counts.plot(kind='bar', color='skyblue')
plt.title('Frequency of Shipping Types')
plt.xlabel('Shipping Type')
plt.ylabel('Frequency')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()

# Find the most preferred shipping type
most_preferred_shipping_type = shipping_type_counts.idxmax()
print("Most preferred shipping type:", most_preferred_shipping_type)
```



Most preferred shipping type: Free Shipping

[ ]:
```
total_purchase_by_season = df.groupby('Season')['Purchase Amount (USD)'].sum()

# Plot total purchase amount by season
plt.figure(figsize=(8, 6))
total_purchase_by_season.plot(kind='bar', color='skyblue')
plt.title('Total Purchase Amount by Season')
plt.xlabel('Season')
plt.ylabel('Total Purchase Amount (USD)')
plt.xticks(rotation=0)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()

# Display total purchase amount by season
print("Total Purchase Amount by Season:")
```

```
print(total_purchase_by_season)
```




```
Total Purchase Amount by Season:
Season
Fall      60018
Spring    58679
Summer    55777
Winter    58607
Name: Purchase Amount (USD), dtype: int64
```

[ ]: