

# Customer Shopping Trends Data Analysis and Visualization

## Importing Libraries

```
In [179...]:  
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
from wordcloud import WordCloud, STOPWORDS  
from IPython.display import Image  
import warnings  
warnings.filterwarnings("ignore")
```

## Loading the Dataset

```
In [180...]: df = pd.read_csv("/input/shopping_trends_updated.csv")  
df.sample(5)
```

Out[180]:

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season
2563	2564	69	Male	Backpack	Accessories	81	Massachusetts	XL	Gray	Winter
2498	2499	18	Male	Coat	Outerwear	39	Massachusetts	S	Green	Fall
3630	3631	65	Female	Hat	Accessories	73	Oklahoma	M	Lavender	Winter
3632	3633	27	Female	Sneakers	Footwear	73	Utah	L	Green	Spring
1271	1272	70	Male	Jewelry	Accessories	57	Kentucky	S	Gray	Spring

## Checking the shape of the Dataset

```
In [181...]: df.shape
```

Out[181]: (3900, 18)

## Checking the number of Columns in the Dataset

```
In [182...]: df.columns
```

```
Out[182]: Index(['Customer ID', 'Age', 'Gender', 'Item Purchased', 'Category',
       'Purchase Amount (USD)', 'Location', 'Size', 'Color', 'Season',
       'Review Rating', 'Subscription Status', 'Shipping Type',
       'Discount Applied', 'Promo Code Used', 'Previous Purchases',
       'Payment Method', 'Frequency of Purchases'],
      dtype='object')
```

#### Information about the Dataset

In [183...]

```
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
```

In [184...]

```
df.describe()
```

Out[184]:

	Customer ID	Age	Purchase Amount (USD)	Review Rating	Previous Purchases
<b>count</b>	3,900.00	3,900.00	3,900.00	3,900.00	3,900.00
<b>mean</b>	1,950.50	44.07	59.76	3.75	25.35
<b>std</b>	1,125.98	15.21	23.69	0.72	14.45
<b>min</b>	1.00	18.00	20.00	2.50	1.00
<b>25%</b>	975.75	31.00	39.00	3.10	13.00
<b>50%</b>	1,950.50	44.00	60.00	3.70	25.00
<b>75%</b>	2,925.25	57.00	81.00	4.40	38.00
<b>max</b>	3,900.00	70.00	100.00	5.00	50.00

Checking if there are any null values present in the dataset

In [185...]

```
df.isnull().sum()
```

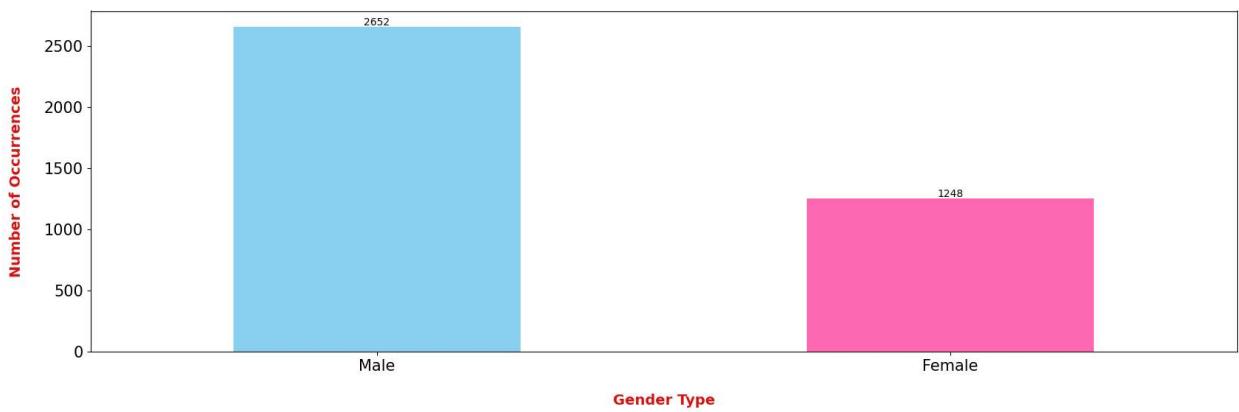
```
Out[185]: Customer ID      0  
Age                  0  
Gender                0  
Item Purchased        0  
Category              0  
Purchase Amount (USD) 0  
Location              0  
Size                  0  
Color                 0  
Season                0  
Review Rating         0  
Subscription Status   0  
Shipping Type          0  
Discount Applied       0  
Promo Code Used       0  
Previous Purchases    0  
Payment Method         0  
Frequency of Purchases 0  
dtype: int64
```

Checking if there are any duplicate values present in the dataset

```
In [186... df.duplicated().sum()
```

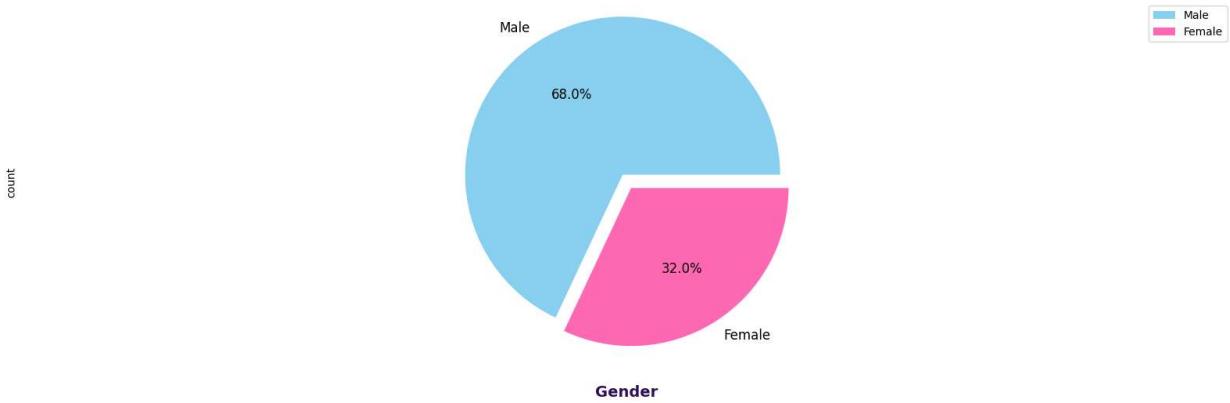
```
Out[186]: 0
```

```
In [187... plt.figure(figsize = (20, 6))  
ax = df["Gender"].value_counts().plot(kind = 'bar', color = colors, rot = 0)  
ax.set_xticklabels(['Male', 'Female'])  
  
for p in ax.patches:  
    ax.annotate(int(p.get_height()), (p.get_x() + 0.25, p.get_height() + 1), ha = 'center')  
    ax.tick_params(axis = 'both', labelsize = 15)  
plt.xlabel('Gender Type', weight = "bold", color = "#D71313", fontsize = 14, labelpad = 10)  
plt.ylabel('Number of Occurrences', weight = "bold", color = "#D71313", fontsize = 14,
```



```
In [188... plt.figure(figsize = (20, 6))
```

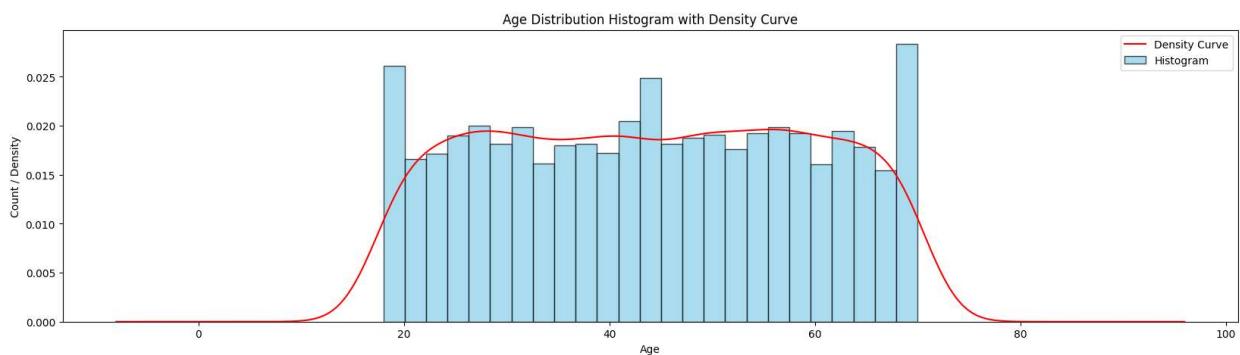
```
counts = df["Gender"].value_counts()  
explode = (0, 0.1)  
  
counts.plot(kind = 'pie', fontsize = 12, colors = colors, explode = explode, autopct = '%1.1f%%',  
plt.xlabel('Gender', weight = "bold", color = "#2F0F5D", fontsize = 14, labelpad = 20)  
plt.axis('equal')  
plt.legend(labels = counts.index, loc = "best")  
plt.show()
```



```
In [189]: fig, ax = plt.subplots(figsize = (20, 5))

ax.hist(df['Age'], bins = 25, edgecolor = 'black', alpha = 0.7, color = 'skyblue', density = True)
df['Age'].plot(kind = 'kde', color = 'red', ax = ax)

ax.set_xlabel('Age')
ax.set_ylabel('Count / Density')
ax.set_title('Age Distribution Histogram with Density Curve')
ax.legend(['Density Curve', 'Histogram'])
plt.show()
```

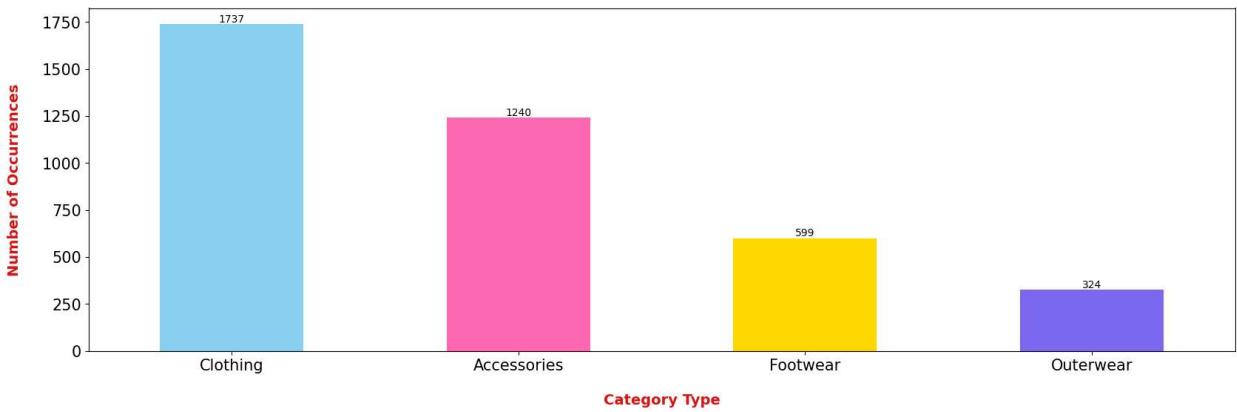


```
In [190]: df["Category"].value_counts()
```

```
Out[190]: Category
Clothing      1737
Accessories    1240
Footwear       599
Outerwear      324
Name: count, dtype: int64
```

```
In [191]: plt.figure(figsize = (20, 6))
ax = df["Category"].value_counts().plot(kind = 'bar', color = colors, rot = 0)
ax.set_xticklabels(['Clothing', 'Accessories', 'Footwear', 'Outerwear'])

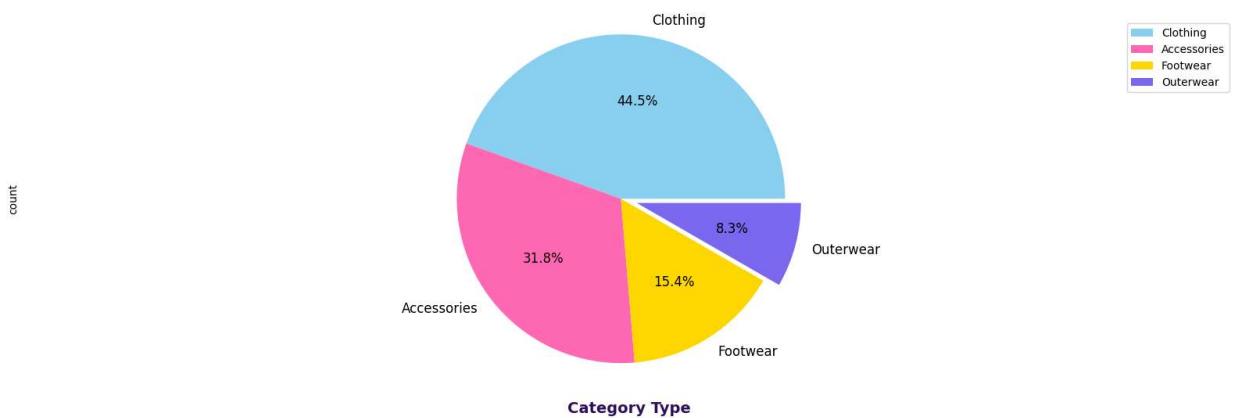
for p in ax.patches:
    ax.annotate(int(p.get_height()), (p.get_x() + 0.25, p.get_height() + 1), ha = 'center')
    ax.tick_params(axis = 'both', labelsize = 15)
plt.xlabel('Category Type', weight = "bold", color = "#D71313", fontsize = 14, labelpad = 10)
plt.ylabel('Number of Occurrences', weight = "bold", color = "#D71313", fontsize = 14,
```



```
In [192...]: plt.figure(figsize = (20, 6))

counts = df["Category"].value_counts()
explode = (0, 0.0, 0.0, 0.1)

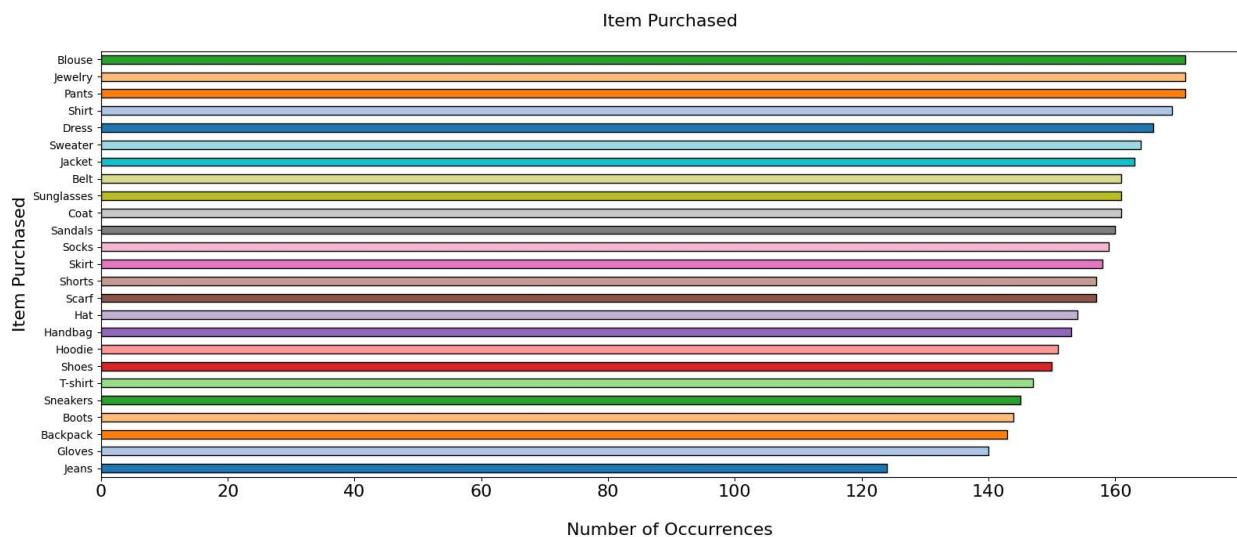
counts.plot(kind = 'pie', fontsize = 12, colors = colors, explode = explode, autopct =
plt.xlabel('Category Type', weight = "bold", color = "#2F0F5D", fontsize = 14, labelpa
plt.axis('equal')
plt.legend(labels = counts.index, loc = "best")
plt.show()
```



```
In [193...]: df["Item Purchased"].value_counts()
```

```
Out[193]: Item Purchased
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: count, dtype: int64
```

```
In [194... plt.figure(figsize = (16, 7))
df["Item Purchased"].value_counts().sort_values(ascending = True).plot(kind = 'barh',
plt.ylabel('Item Purchased', fontsize = 16)
plt.xlabel('\nNumber of Occurrences', fontsize = 16)
plt.title('Item Purchased\n', fontsize = 16)
plt.xticks(rotation = 0, ha = 'center', fontsize = 16)
plt.tight_layout()
plt.show()
```

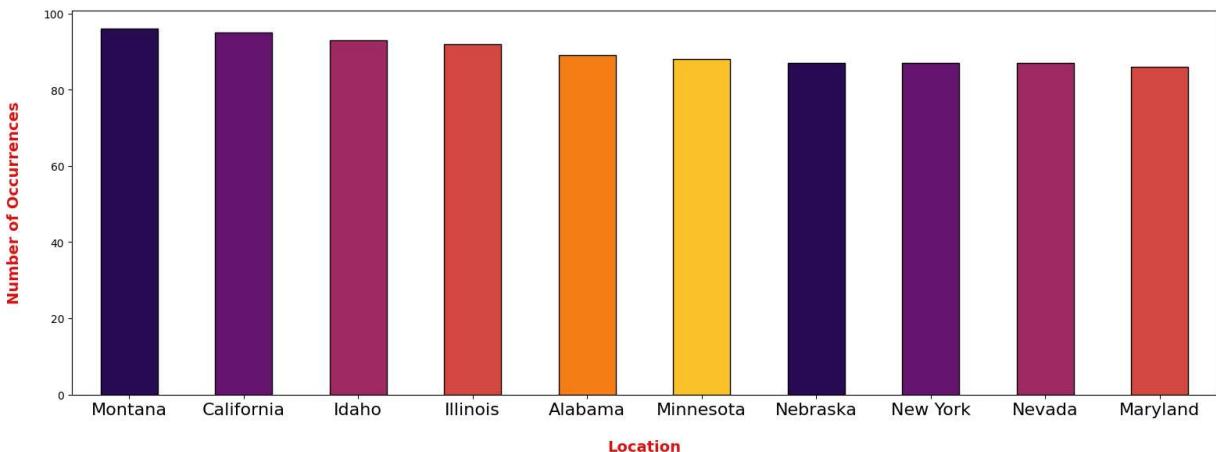


```
In [195... df["Location"].value_counts()
```

```
Out[195]: Location
Montana          96
California       95
Idaho            93
Illinois          92
Alabama           89
Minnesota         88
Nebraska          87
New York          87
Nevada            87
Maryland           86
Delaware           86
Vermont            85
Louisiana          84
North Dakota       83
Missouri           81
West Virginia       81
New Mexico          81
Mississippi         80
Indiana             79
Georgia              79
Kentucky             79
Arkansas              79
North Carolina        78
Connecticut           78
Virginia              77
Ohio                  77
Tennessee             77
Texas                  77
Maine                  77
South Carolina          76
Colorado                75
Oklahoma                75
Wisconsin               75
Oregon                  74
Pennsylvania             74
Washington              73
Michigan                 73
Alaska                  72
Massachusetts             72
Wyoming                  71
Utah                     71
New Hampshire              71
South Dakota              70
Iowa                      69
Florida                  68
New Jersey                 67
Hawaii                     65
Arizona                     65
Kansas                      63
Rhode Island                  63
Name: count, dtype: int64
```

```
In [196...]: plt.figure(figsize = (16, 6))
df["Location"].value_counts()[:10].sort_values(ascending = False).plot(kind = 'bar', color = "#D71313")
plt.xlabel('Location', weight = "bold", color = "#D71313", fontsize = 14, labelpad = 2)
plt.ylabel('\nNumber of Occurrences', weight = "bold", color = "#D71313", fontsize = 14, labelpad = 2)
plt.xticks(rotation = 0, ha = 'center', fontsize = 16)
```

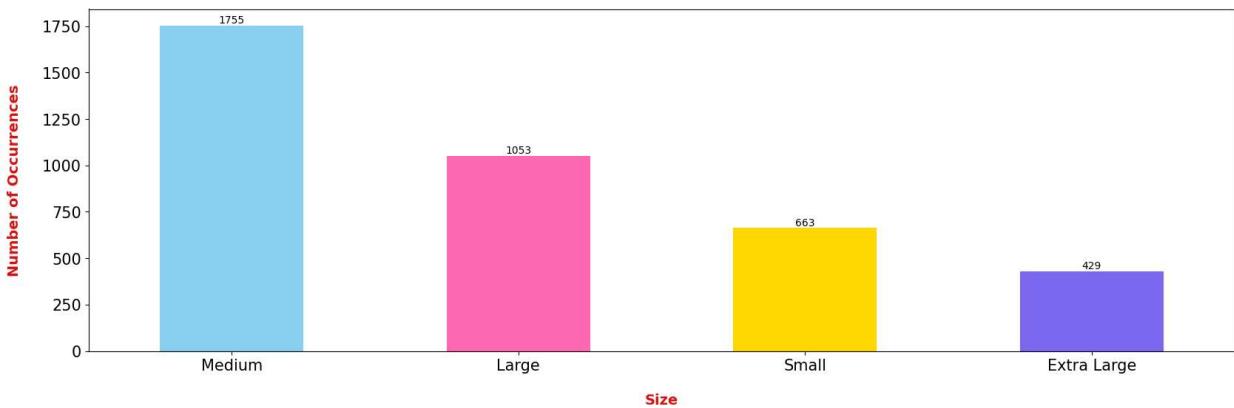
```
plt.tight_layout()  
plt.show()
```



```
In [197]: df["Size"].value_counts()
```

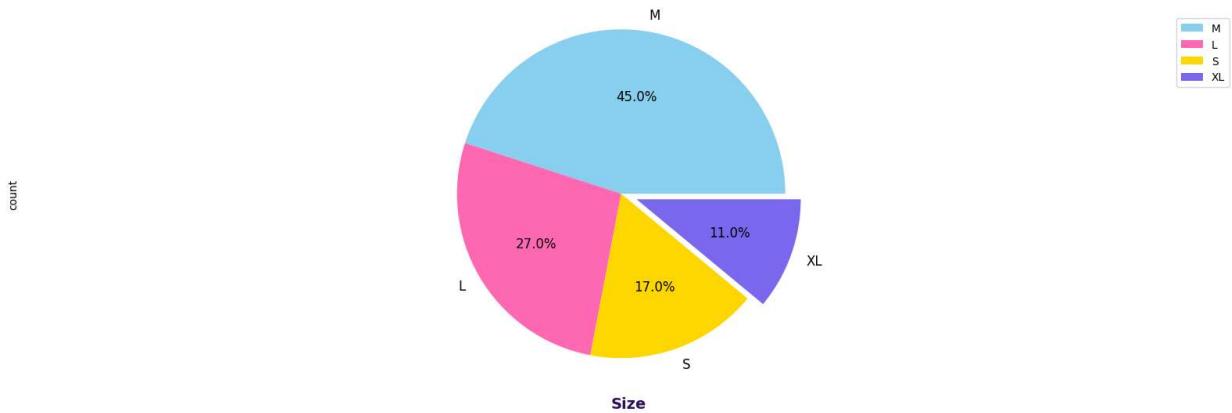
```
Out[197]: Size  
M    1755  
L    1053  
S     663  
XL    429  
Name: count, dtype: int64
```

```
In [198]: plt.figure(figsize = (20, 6))  
ax = df["Size"].value_counts().plot(kind = 'bar', color = colors, rot = 0)  
ax.set_xticklabels(('Medium', 'Large', 'Small', 'Extra Large'))  
  
for p in ax.patches:  
    ax.annotate(int(p.get_height()), (p.get_x() + 0.25, p.get_height() + 1), ha = 'center')  
    ax.tick_params(axis = 'both', labelsize = 15)  
plt.xlabel('Size', weight = "bold", color = "#D71313", fontsize = 14, labelpad = 20)  
plt.ylabel('Number of Occurrences', weight = "bold", color = "#D71313", fontsize = 14,
```



```
In [199]: plt.figure(figsize = (20, 6))
```

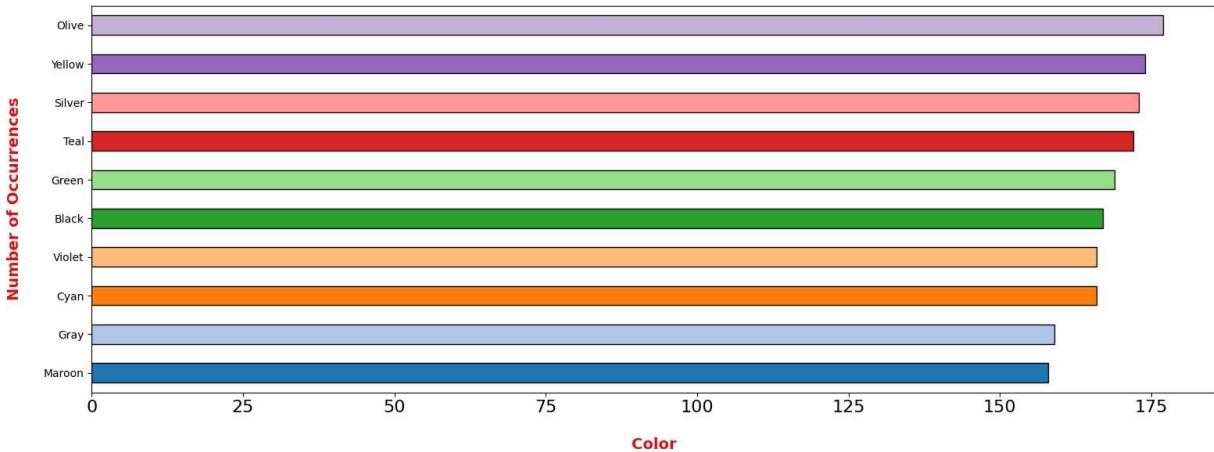
```
counts = df["Size"].value_counts()  
explode = (0, 0.0, 0.0, 0.1)  
  
counts.plot(kind = 'pie', fontsize = 12, colors = colors, explode = explode, autopct =  
plt.xlabel('Size', weight = "bold", color = "#2F0F5D", fontsize = 14, labelpad = 20)  
plt.axis('equal')  
plt.legend(labels = counts.index, loc = "best")  
plt.show()
```



```
In [200]: df["Color"].value_counts()
```

```
Out[200]: Color
Olive      177
Yellow     174
Silver     173
Teal       172
Green      169
Black      167
Cyan       166
Violet     166
Gray       159
Maroon     158
Orange     154
Charcoal   153
Pink       153
Magenta    152
Blue       152
Purple     151
Peach      149
Red        148
Beige      147
Indigo     147
Lavender   147
Turquoise  145
White      142
Brown      141
Gold       138
Name: count, dtype: int64
```

```
In [201]: plt.figure(figsize = (16, 6))
df["Color"].value_counts()[:10].sort_values(ascending = True).plot(kind = 'barh', color = "#D71313")
plt.xlabel('Color', weight = "bold", color = "#D71313", fontsize = 14, labelpad = 20)
plt.ylabel('\nNumber of Occurrences', weight = "bold", color = "#D71313", fontsize = 14)
plt.xticks(rotation = 0, ha = 'center', fontsize = 16)
plt.tight_layout()
plt.show()
```



```
In [202]: df["Season"].value_counts()
```

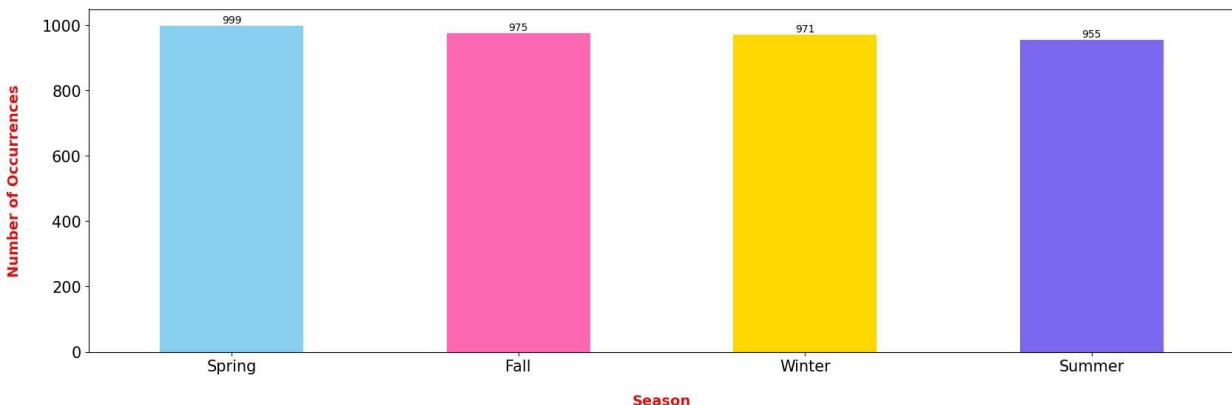
Out[202]:

Season	count
Spring	999
Fall	975
Winter	971
Summer	955

Name: count, dtype: int64

```
In [203]: plt.figure(figsize = (20, 6))
ax = df["Season"].value_counts().plot(kind = 'bar', color = colors, rot = 0)
ax.set_xticklabels(['Spring', 'Fall', 'Winter', 'Summer'])

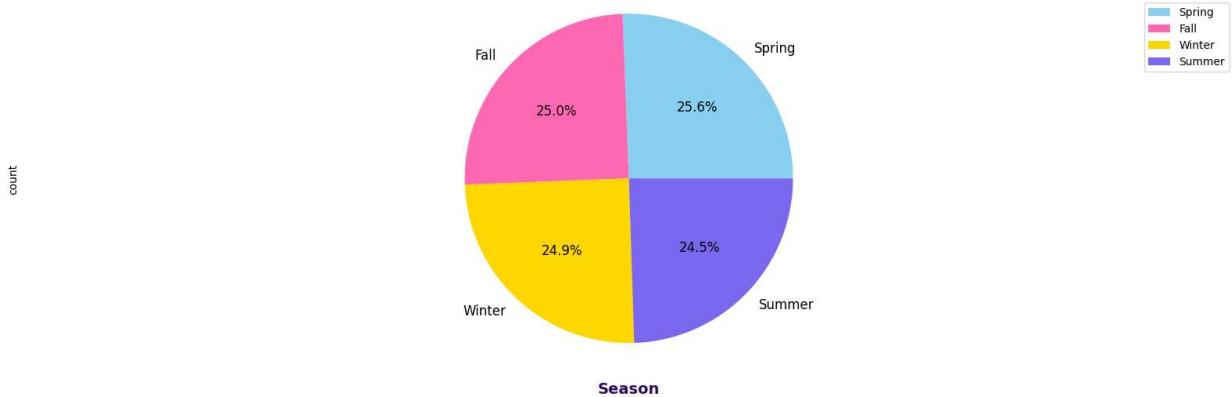
for p in ax.patches:
    ax.annotate(int(p.get_height()), (p.get_x() + 0.25, p.get_height() + 1), ha = 'center')
    ax.tick_params(axis = 'both', labelsize = 15)
plt.xlabel('Season', weight = "bold", color = "#D71313", fontsize = 14, labelpad = 20)
plt.ylabel('Number of Occurrences', weight = "bold", color = "#D71313", fontsize = 14,
```



```
In [204]: plt.figure(figsize = (20, 6))
```

```
counts = df["Season"].value_counts()
explode = (0, 0, 0, 0)

counts.plot(kind = 'pie', fontsize = 12, colors = colors, explode = explode, autopct =
plt.xlabel('Season', weight = "bold", color = "#2F0F5D", fontsize = 14, labelpad = 20)
plt.axis('equal')
plt.legend(labels = counts.index, loc = "best")
plt.show()
```



```
In [205]: df["Subscription Status"].value_counts()
```

Out[205]: Subscription Status

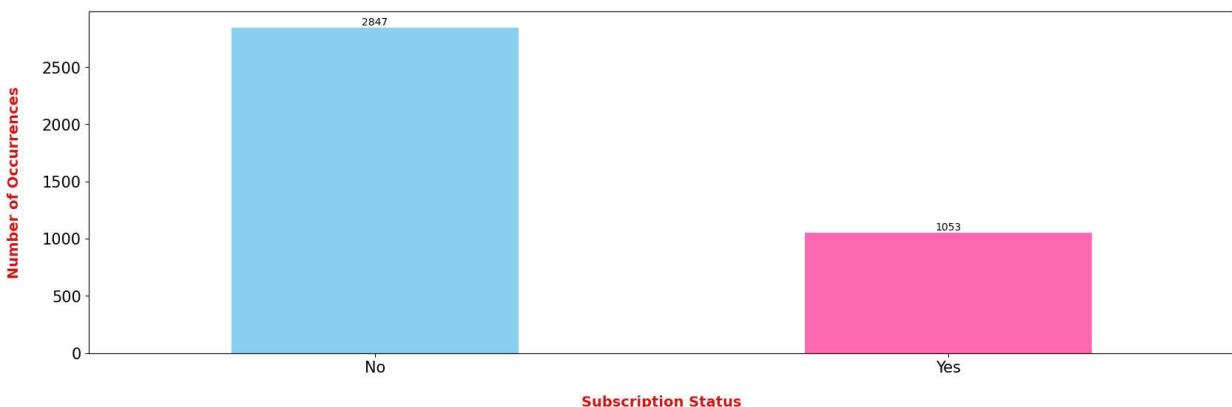
No 2847

Yes 1053

Name: count, dtype: int64

```
In [206]: plt.figure(figsize = (20, 6))
ax = df["Subscription Status"].value_counts().plot(kind = 'bar', color = colors, rot =
ax.set_xticklabels(['No', 'Yes'])

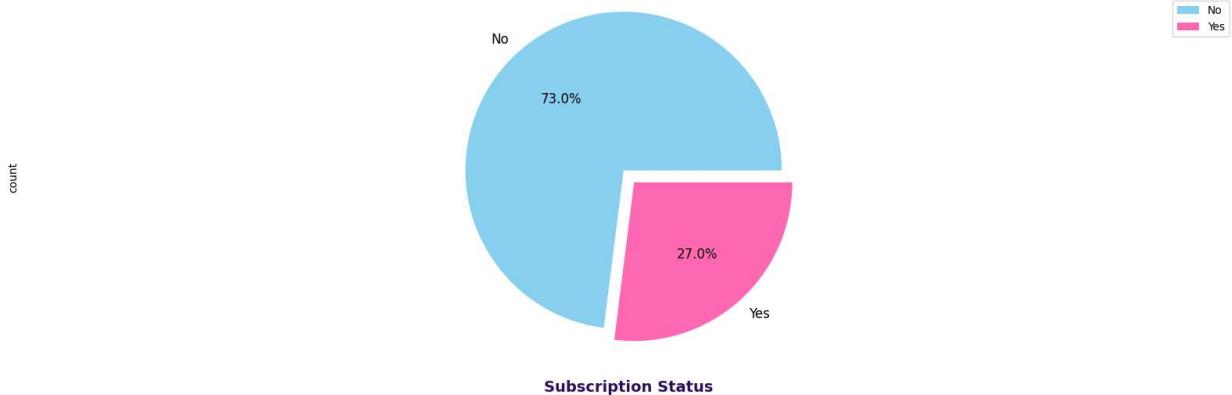
for p in ax.patches:
    ax.annotate(int(p.get_height()), (p.get_x() + 0.25, p.get_height() + 1), ha = 'center')
    ax.tick_params(axis = 'both', labelsize = 15)
plt.xlabel('Subscription Status', weight = "bold", color = "#D71313", fontsize = 14, )
plt.ylabel('Number of Occurrences', weight = "bold", color = "#D71313", fontsize = 14,
```



```
In [207]: plt.figure(figsize = (20, 6))
```

```
counts = df["Subscription Status"].value_counts()
explode = (0, 0.1)
```

```
counts.plot(kind = 'pie', fontsize = 12, colors = colors, explode = explode, autopct =
plt.xlabel('Subscription Status', weight = "bold", color = "#2F0F5D", fontsize = 14, )
plt.axis('equal')
plt.legend(labels = counts.index, loc = "best")
plt.show()
```

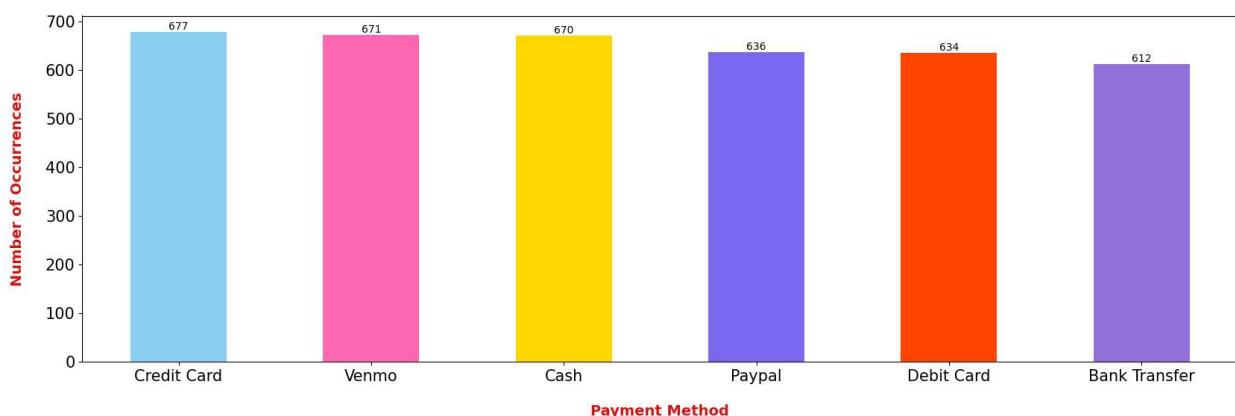


```
In [208]: df["Payment Method"].value_counts()
```

Out[208]: Payment Method

PayPal	677
Credit Card	671
Cash	670
Debit Card	636
Venmo	634
Bank Transfer	612
Name: count, dtype: int64	

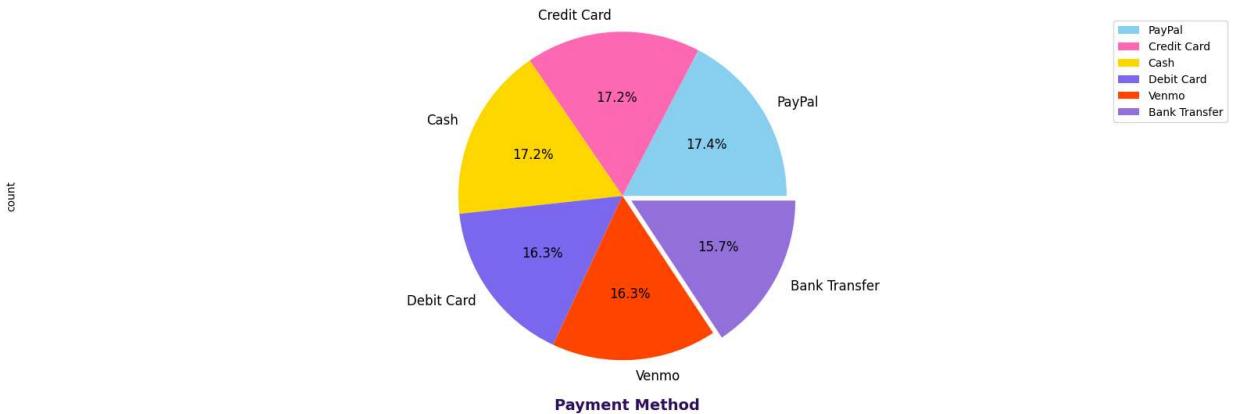
```
In [209... plt.figure(figsize = (20, 6))
ax = df["Payment Method"].value_counts().plot(kind = 'bar', color = colors, rot = 0)
ax.set_xticklabels(('Credit Card', 'Venmo', 'Cash', 'Paypal', 'Debit Card', 'Bank Tran
for p in ax.patches:
    ax.annotate(int(p.get_height()), (p.get_x() + 0.25, p.get_height() + 1), ha = 'ce
    ax.tick_params(axis = 'both', labelsize = 15)
plt.xlabel('Payment Method', weight = "bold", color = "#D71313", fontsize = 14, labelp
plt.ylabel('Number of Occurrences', weight = "bold", color = "#D71313", fontsize = 14,
```



```
In [210... plt.figure(figsize = (20, 6))
```

```
counts = df["Payment Method"].value_counts()
explode = (0, 0, 0, 0, 0.0, 0.06)

counts.plot(kind = 'pie', fontsize = 12, colors = colors, explode = explode, autopct =
plt.xlabel('Payment Method', weight = "bold", color = "#2F0F5D", fontsize = 14, labelp
plt.axis('equal')
plt.legend(labels = counts.index, loc = "best")
plt.show()
```



```
In [211]: df["Shipping Type"].value_counts()
```

```
Out[211]: Shipping Type
Free Shipping    675
Standard        654
Store Pickup    650
Next Day Air    648
Express          646
2-Day Shipping   627
Name: count, dtype: int64
```

```
In [212]: plt.figure(figsize = (20, 6))
ax = df["Shipping Type"].value_counts().plot(kind = 'bar', color = colors, rot = 0)
ax.set_xticklabels(['Free Shipping', 'Standard', 'Store Pickup', 'Next Day Air', 'Express', '2-Day Shipping'])

for p in ax.patches:
    ax.annotate(int(p.get_height()), (p.get_x() + 0.25, p.get_height() + 1), ha = 'center')
    ax.tick_params(axis = 'both', labelsize = 15)
plt.xlabel('Shipping Type', weight = "bold", color = "#D71313", fontsize = 14, labelpad = 10)
plt.ylabel('Number of Occurrences', weight = "bold", color = "#D71313", fontsize = 14, labelpad = 10)
```



```
In [213]: plt.figure(figsize = (20, 6))
```

```
counts = df["Shipping Type"].value_counts()
explode = (0, 0, 0, 0, 0.0, 0.06)

counts.plot(kind = 'pie', fontsize = 12, colors = colors, explode = explode, autopct =
plt.xlabel('Shipping Type', weight = "bold", color = "#2F0F5D", fontsize = 14, labelpad = 10)
plt.axis('equal')
plt.legend(labels = counts.index, loc = "best")
plt.show()
```



```
In [214...  
text = " ".join(title for title in df["Frequency of Purchases"])  
word_cloud = WordCloud(collocations = False, background_color = 'white').generate(text)  
plt.figure(figsize = (20, 5))  
plt.imshow(word_cloud, interpolation = 'bilinear')  
plt.axis("off")  
plt.title('WorldCloud for Frequency of Purchases\n', fontsize = 12, color = "#FF0000")  
plt.show()
```

WorldCloud for Frequency of Purchases



```
In [215... df["Frequency of Purchases"].value_counts()
```

Out[215]:

Frequency of Purchases	count
Every 3 Months	584
Annually	572
Quarterly	563
Monthly	553
Bi-Weekly	547
Fortnightly	542
Weekly	539

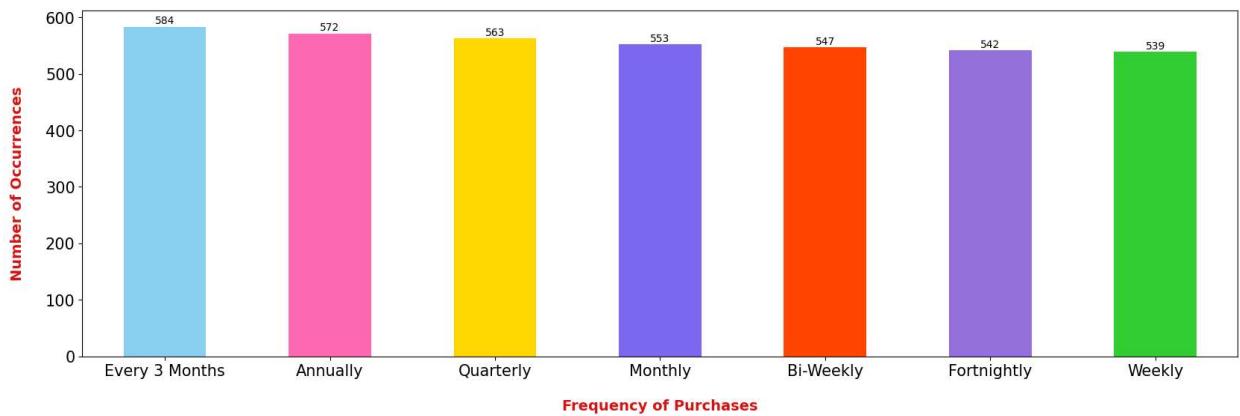
Name: count, dtype: int64

```
In [216... plt.figure(figsize = (20, 6))  
ax = df["Frequency of Purchases"].value_counts().plot(kind = 'bar', color = colors, rc  
ax.set_xticklabels(['Every 3 Months', 'Annually', 'Quarterly', 'Monthly', 'Bi-Weekly',
```

```

for p in ax.patches:
    ax.annotate(int(p.get_height()), (p.get_x() + 0.25, p.get_height() + 1), ha = 'center')
ax.tick_params(axis = 'both', labelsize = 15)
plt.xlabel('Frequency of Purchases', weight = "bold", color = "#D71313", fontsize = 14)
plt.ylabel('Number of Occurrences', weight = "bold", color = "#D71313", fontsize = 14)

```



In [217...]

```

plt.figure(figsize = (20, 6))

counts = df["Frequency of Purchases"].value_counts()
explode = (0, 0, 0, 0, 0.0, 0, 0.06)

counts.plot(kind = 'pie', fontsize = 12, colors = colors, explode = explode, autopct = '%.1f%%',
            plt.xlabel('Frequency of Purchases', weight = "bold", color = "#2F0F5D", fontsize = 14),
            plt.axis('equal'))
plt.legend(labels = counts.index, loc = "best")
plt.show()

```



What is the average age of customers in the dataset ?

In [218...]

```

average_age = df['Age'].mean()
print("Average Age:", average_age)

```

Average Age: 44.06846153846154

What is the most common item purchased ?

In [219...]

```

most_common_item = df['Item Purchased'].mode()[0]
print("Most Common Item Purchased:", most_common_item)

```

Most Common Item Purchased: Blouse

What is the total purchase amount for each category ?

```
In [220...]: total_purchase_by_category = df.groupby('Category')['Purchase Amount (USD)'].sum()
print("Total Purchase Amount by Category:")
print(total_purchase_by_category)
```

Total Purchase Amount by Category:

Category

Accessories	74200
Clothing	104264
Footwear	36093
Outerwear	18524

Name: Purchase Amount (USD), dtype: int64

What is the average review rating for male customers and female customers separately ?

```
In [221...]: average_rating_male = df[df['Gender'] == 'Male']['Review Rating'].mean()
average_rating_female = df[df['Gender'] == 'Female']['Review Rating'].mean()
print("Average Review Rating for Male Customers:", average_rating_male)
print("Average Review Rating for Female Customers:", average_rating_female)
```

Average Review Rating for Male Customers: 3.7539592760180995

Average Review Rating for Female Customers: 3.741426282051282

What is the most common payment method used by customers ?

```
In [222...]: most_common_payment_method = df['Payment Method'].mode()[0]
print("Most Common Payment Method:", most_common_payment_method)
```

Most Common Payment Method: PayPal

What is the median purchase amount (USD) ?

```
In [223...]: median_purchase_amount = df['Purchase Amount (USD)'].median()
print("Median Purchase Amount (USD):", median_purchase_amount)
```

Median Purchase Amount (USD): 60.0

How many customers have opted for the Subscription ?

```
In [224...]: subscription_count = df[df['Subscription Status'] == 'Yes']['Customer ID'].count()
print("Number of Customers with Subscription: ", subscription_count)
```

Number of Customers with Subscription: 1053

What is the average purchase amount for customers with a subscription status of 'Yes' and 'No' ?

```
In [225...]: avg_purchase_subscription_yes = df[df['Subscription Status'] == 'Yes']['Purchase Amount'].mean()
avg_purchase_subscription_no = df[df['Subscription Status'] == 'No']['Purchase Amount'].mean()
print("Average Purchase Amount for Subscription 'Yes':", avg_purchase_subscription_yes)
print("Average Purchase Amount for Subscription 'No':", avg_purchase_subscription_no)
```

Average Purchase Amount for Subscription 'Yes': 59.49192782526116

Average Purchase Amount for Subscription 'No': 59.865121180189675

What is the most common season for purchases ?

```
In [226...]: most_common_season = df['Season'].mode()[0]
print("Most Common Season for Purchases:", most_common_season)
```

Most Common Season for Purchases: Spring

What is the total purchase amount for each gender ?

```
In [227...]: total_purchase_by_gender = df.groupby('Gender')['Purchase Amount (USD)'].sum()
print("Total Purchase Amount by Gender:")
print(total_purchase_by_gender)
```

Total Purchase Amount by Gender:  
Gender  
Female 75191  
Male 157890  
Name: Purchase Amount (USD), dtype: int64

What is the average age of customers who made purchases in the Summer season ?

```
In [228...]: average_age_summer = df[df['Season'] == 'Summer']['Age'].mean()
print("Average Age of Customers in the Summer Season:", average_age_summer)
```

Average Age of Customers in the Summer Season: 43.973821989528794

How many customers used a promo code for their purchase ?

```
In [229...]: promo_code_count = df[df['Promo Code Used'] == 'Yes']['Customer ID'].count()
print("Number of Customers who used Promo Code:", promo_code_count)
```

Number of Customers who used Promo Code: 1677

What is the maximum and minimum review rating in the dataset ?

```
In [230...]: max_review_rating = df['Review Rating'].max()
min_review_rating = df['Review Rating'].min()
print("Maximum Review Rating:", max_review_rating)
print("Minimum Review Rating:", min_review_rating)
```

Maximum Review Rating: 5.0  
Minimum Review Rating: 2.5

What is the most common shipping type for customers with a review rating above 4 ?

```
In [231...]: common_shipping_high_rating = df[df['Review Rating'] > 4]['Shipping Type'].mode()[0]
print("Most Common Shipping Type for High Review Ratings:", common_shipping_high_rating)
```

Most Common Shipping Type for High Review Ratings: Standard

How many customers have made more than 30 previous purchases ?

```
In [232...]: customers_above_30_previous_purchases = df[df['Previous Purchases'] > 30]['Customer ID']
print("Number of Customers with more than 30 Previous Purchases:", customers_above_30_
```

Number of Customers with more than 30 Previous Purchases: 1549

What is the average purchase amount for customers who have made more than 30 previous purchases ?

```
In [233...]: avg_purchase_above_30_previous_purchases = df[df['Previous Purchases'] > 30]['Purchase Amount (USD)'].mean()
print("Average Purchase Amount for Customers with more than 30 Previous Purchases:", avg_purchase_above_30_
```

Average Purchase Amount for Customers with more than 30 Previous Purchases: 60.028405  
42285345

What is the most common payment method for customers who shop in the Winter season ?

```
In [234...]: total_purchase_free_shipping = df[df['Shipping Type'] == 'Free Shipping']['Purchase Amount (USD)'].sum()
print("Total Purchase Amount for 'Free Shipping' Shipping Type:", total_purchase_free_shipping)
```

Total Purchase Amount for 'Free Shipping' Shipping Type: 40777

What is the total purchase amount for customers with a 'Free Shipping' shipping type ?

In [235...]

```
total_purchase_free_shipping = df[df['Shipping Type'] == 'Free Shipping']['Purchase Amount'].sum()
print("Total Purchase Amount for 'Free Shipping' Shipping Type:", total_purchase_free_shipping)
```

Total Purchase Amount for 'Free Shipping' Shipping Type: 40777

What is the average purchase amount for customers who used a discount ?

In [236...]

```
avg_purchase_with_discount = df[df['Discount Applied'] == 'Yes']['Purchase Amount (USD)'].mean()
print("Average Purchase Amount for Customers with Discount Applied:", avg_purchase_with_discount)
```

Average Purchase Amount for Customers with Discount Applied: 59.27906976744186

What is the most common category of items purchased by female customers with a review rating below 3?

In [237...]

```
common_category_low_rating_female = df[(df['Gender'] == 'Female') & (df['Review Rating'] < 3)]['Category'].mode()
print("Most Common Category for Low Review Rating Female Customers:", common_category_low_rating_female)
```

Most Common Category for Low Review Rating Female Customers: Clothing

What is the average age of customers who made purchases with a review rating above 4 and used a promo code?

In [238...]

```
average_age_high_rating_promo = df[(df['Review Rating'] > 4) & (df['Promo Code Used'] == 'Yes')]['Age'].mean()
print("Average Age of Customers with High Review Ratings and Promo Code Used:", average_age_high_rating_promo)
```

Average Age of Customers with High Review Ratings and Promo Code Used: 43.9872

What is the total purchase amount for customers in each location?

In [239...]

```
total_purchase_by_location = df.groupby('Location')['Purchase Amount (USD)'].sum()
print("Total Purchase Amount by Location:")
print(total_purchase_by_location)
```

Total Purchase Amount by Location:

Location	Purchase Amount (USD)
Alabama	5261
Alaska	4867
Arizona	4326
Arkansas	4828
California	5605
Colorado	4222
Connecticut	4226
Delaware	4758
Florida	3798
Georgia	4645
Hawaii	3752
Idaho	5587
Illinois	5617
Indiana	4655
Iowa	4201
Kansas	3437
Kentucky	4402
Louisiana	4848
Maine	4388
Maryland	4795
Massachusetts	4384
Michigan	4533
Minnesota	4977
Mississippi	4883
Missouri	4691
Montana	5784
Nebraska	5172
Nevada	5514
New Hampshire	4219
New Jersey	3802
New Mexico	5014
New York	5257
North Carolina	4742
North Dakota	5220
Ohio	4649
Oklahoma	4376
Oregon	4243
Pennsylvania	4926
Rhode Island	3871
South Carolina	4439
South Dakota	4236
Tennessee	4772
Texas	4712
Utah	4443
Vermont	4860
Virginia	4842
Washington	4623
West Virginia	5174
Wisconsin	4196
Wyoming	4309

Name: Purchase Amount (USD), dtype: int64

What is the average purchase amount for customers who have a subscription and used Venmo as the payment method?

In [240...]

```
avg_purchase_subscription_venmo = df[(df['Subscription Status'] == 'Yes') & (df['Payment Method'] == 'Venmo')].mean()
```

Average Purchase Amount for Customers with Subscription and Venmo Payment Method: 57.  
51149425287356

What is the frequency distribution of the 'Frequency of Purchases' column?

```
In [241...]: purchase_frequency_distribution = df['Frequency of Purchases'].value_counts()  
print("Frequency Distribution of Purchase Frequency:")  
print(purchase_frequency_distribution)
```

Frequency Distribution of Purchase Frequency:  
Frequency of Purchases  
Every 3 Months 584  
Annually 572  
Quarterly 563  
Monthly 553  
Bi-Weekly 547  
Fortnightly 542  
Weekly 539  
Name: count, dtype: int64

What is the average purchase amount for each color of items?

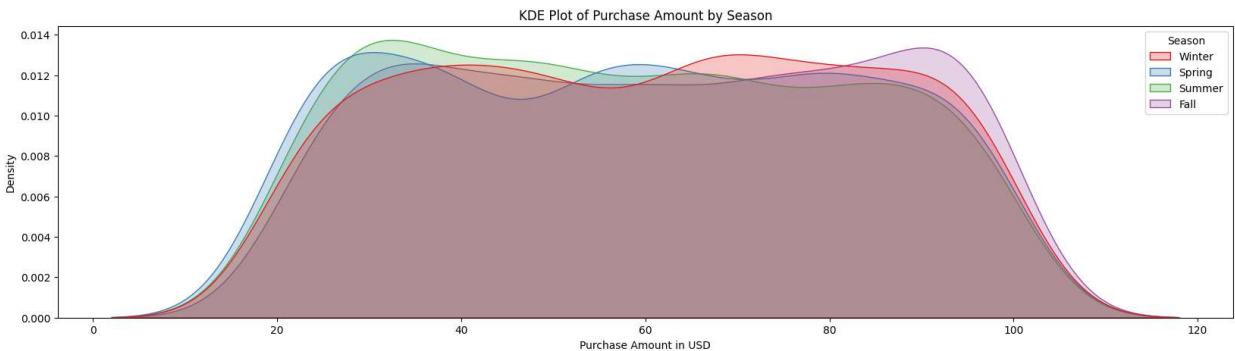
```
In [242...]: avg_purchase_by_color = df.groupby('Color')['Purchase Amount (USD)'].mean()  
print("Average Purchase Amount by Color:")  
print(avg_purchase_by_color)
```

Average Purchase Amount by Color:  
Color  
Beige 60.41  
Black 58.40  
Blue 56.95  
Brown 59.06  
Charcoal 60.63  
Cyan 61.89  
Gold 61.01  
Gray 62.49  
Green 65.70  
Indigo 56.25  
Lavender 59.13  
Magenta 57.13  
Maroon 59.53  
Olive 58.15  
Orange 60.89  
Peach 59.19  
Pink 60.59  
Purple 60.01  
Red 59.32  
Silver 56.83  
Teal 60.81  
Turquoise 55.61  
Violet 61.72  
White 62.64  
Yellow 59.24  
Name: Purchase Amount (USD), dtype: float64

KDE plot for Purchase Amount by Season

```
In [244...]: plt.figure(figsize=(20, 5))  
sns.kdeplot(data = df, x = 'Purchase Amount (USD)', hue = 'Season', common_norm = False)  
plt.title('KDE Plot of Purchase Amount by Season')
```

```
plt.xlabel('Purchase Amount in USD')
plt.show()
```



What is the most common payment method for customers who purchased items in the Fall season?

```
In [245...]: common_payment_fall = df[df['Season'] == 'Fall']['Payment Method'].mode()[0]
print("Most Common Payment Method for Fall Season Purchases:", common_payment_fall)
```

Most Common Payment Method for Fall Season Purchases: Cash

How many customers have made a purchase in each category?

```
In [246...]: purchase_count_by_category = df['Category'].value_counts()
print("Purchase Count by Category:")
print(purchase_count_by_category)
```

Purchase Count by Category:

Category	Count
Clothing	1737
Accessories	1240
Footwear	599
Outerwear	324

Name: count, dtype: int64

What is the average age of customers who purchased accessories with a discount applied?

```
In [247...]: avg_age_accessories_discount = df[(df['Category'] == 'Accessories') & (df['Discount Applied'] == True)]['Age'].mean()
print("Average Age of Customers who purchased Accessories with Discount Applied:", avg_age_accessories_discount)
```

Average Age of Customers who purchased Accessories with Discount Applied: 44.23941068  
139963

What is the total purchase amount for each size of clothing items (XL, L, M, S)?

```
In [248...]: total_purchase_by_size = df[df['Category'] == 'Clothing'].groupby('Size')['Purchase Amount'].sum()
print("Total Purchase Amount by Size for Clothing Items:")
print(total_purchase_by_size)
```

Total Purchase Amount by Size for Clothing Items:

Size	Purchase Amount (USD)
L	27864
M	47041
S	17416
XL	11943

Name: Purchase Amount (USD), dtype: int64

What is the total purchase amount for customers who have made more than 40 previous purchases?

```
In [249... total_purchase_above_40_previous = df[df['Previous Purchases'] > 40]['Purchase Amount  
print("Total Purchase Amount for Customers with more than 40 Previous Purchases:", tot
```

Total Purchase Amount for Customers with more than 40 Previous Purchases: 46590

What is the most common location for customers who purchased socks with a discount applied?

```
In [250... common_location_socks_discount = df[(df['Category'] == 'Clothing') & (df['Discount App  
print("Most Common Location for Socks Purchases with Discount Applied:", common_locati
```

Most Common Location for Socks Purchases with Discount Applied: Alabama

What is the average purchase amount for customers who have a subscription and used Venmo as the payment method, but did not use a promo code?

```
In [251... avg_purchase_subscription_venmo_no_promo = df[(df['Subscription Status'] == 'Yes') &  
print("Average Purchase Amount for Customers with Subscription, Venmo Payment, and No
```

Average Purchase Amount for Customers with Subscription, Venmo Payment, and No Promo Code: nan

What is the most common item purchased by customers in Louisiana with a review rating of 4 or higher?

```
In [252... common_item_high_rating_louisiana = df[(df['Location'] == 'Louisiana') & (df['Review R  
print("Most Common Item Purchased by High-Rating Customers in Louisiana:", common_item
```

Most Common Item Purchased by High-Rating Customers in Louisiana: Sweater

What is the total purchase amount for customers who made purchases in the Fall season and used a credit card as the payment method?

```
In [253... total_purchase_fall_credit_card = df[(df['Season'] == 'Fall') & (df['Payment Method']  
print("Total Purchase Amount for Fall Season Purchases with Credit Card Payment Method
```

Total Purchase Amount for Fall Season Purchases with Credit Card Payment Method: 9905

What is the most common category of items purchased by male customers in the Winter season with a review rating below 3?

```
In [254... common_category_low_rating_male_winter = df[(df['Gender'] == 'Male') & (df['Season'] ==  
print("Most Common Category for Low-Rating Male Customers in Winter Season:", common_cat
```

Most Common Category for Low-Rating Male Customers in Winter Season: Clothing

How many customers have a subscription status of "Yes" and used a promo code for their purchase?

```
In [255... subscription_promo_count = df[(df['Subscription Status'] == 'Yes') & (df['Promo Code U  
print("Number of Customers with Subscription and Promo Code Used: ", subscription_prom
```

Number of Customers with Subscription and Promo Code Used: 1053

What is the correlation between a customer's age and their total previous purchases? Calculate the Pearson correlation coefficient.

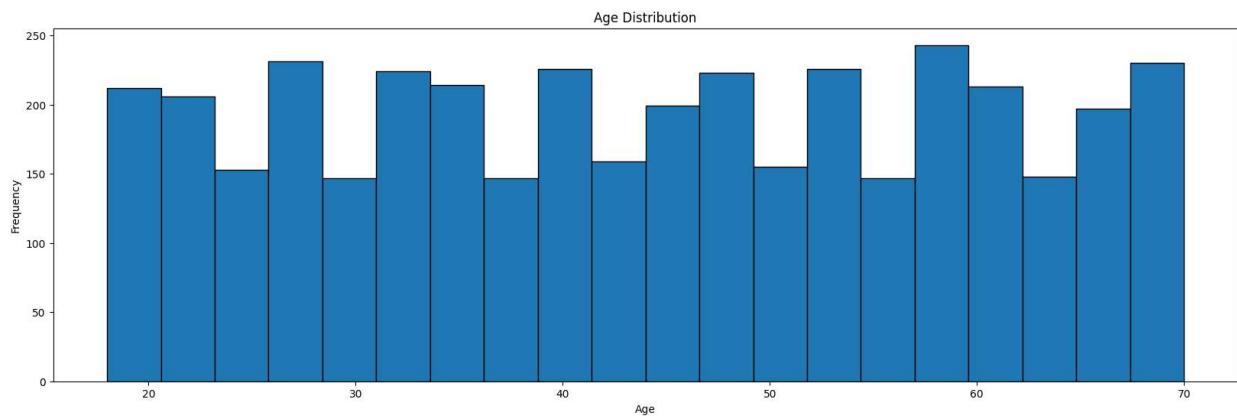
```
In [256... correlation_age_previous_purchases = df['Age'].corr(df['Previous Purchases'], method='  
print("Pearson Correlation between Age and Previous Purchases:", correlation_age_previ
```

Pearson Correlation between Age and Previous Purchases: 0.04044453114289469

Histogram of Age Distribution

In [257...]

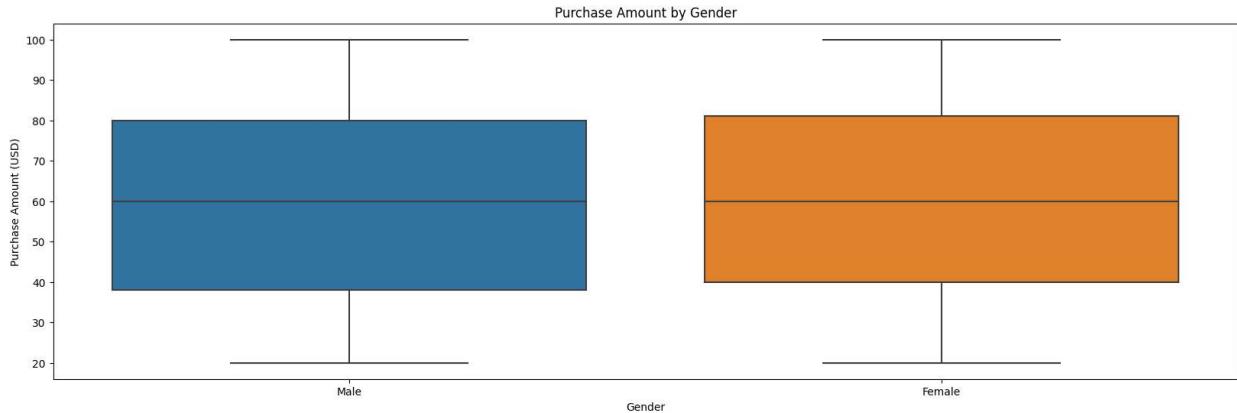
```
plt.figure(figsize = (20, 6))
plt.hist(df['Age'], bins = 20, edgecolor = 'k')
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



### Box Plot of Purchase Amount by Gender

In [258...]

```
plt.figure(figsize = (20, 6))
sns.boxplot(x='Gender', y='Purchase Amount (USD)', data=df)
plt.title('Purchase Amount by Gender')
plt.xlabel('Gender')
plt.ylabel('Purchase Amount (USD)')
plt.show()
```



### Violin Plot of Review Rating by Category

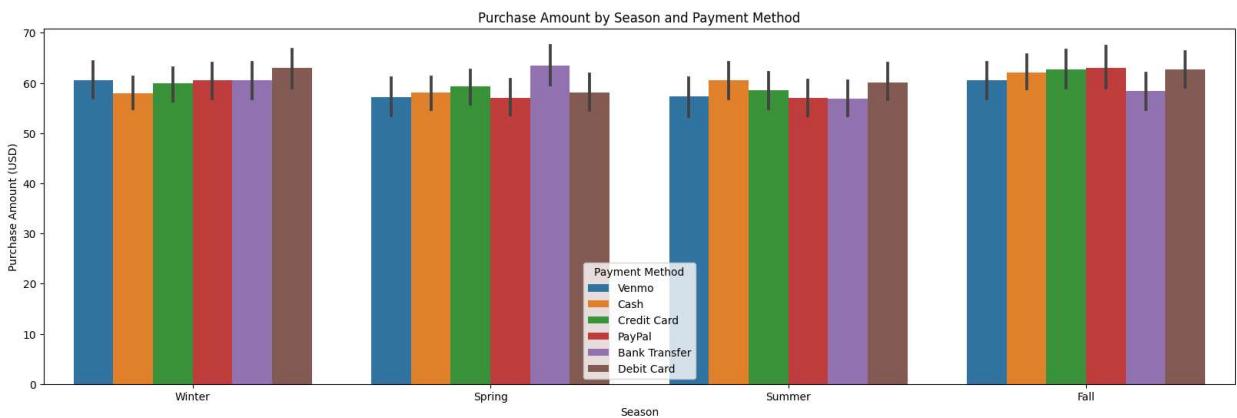
In [261...]

```
plt.figure(figsize=(20, 6))
sns.violinplot(x='Category', y='Review Rating', data=df)
plt.title('Review Rating by Category')
plt.xlabel('Category')
plt.ylabel('Review Rating')
plt.xticks(rotation=45)
plt.show()
```



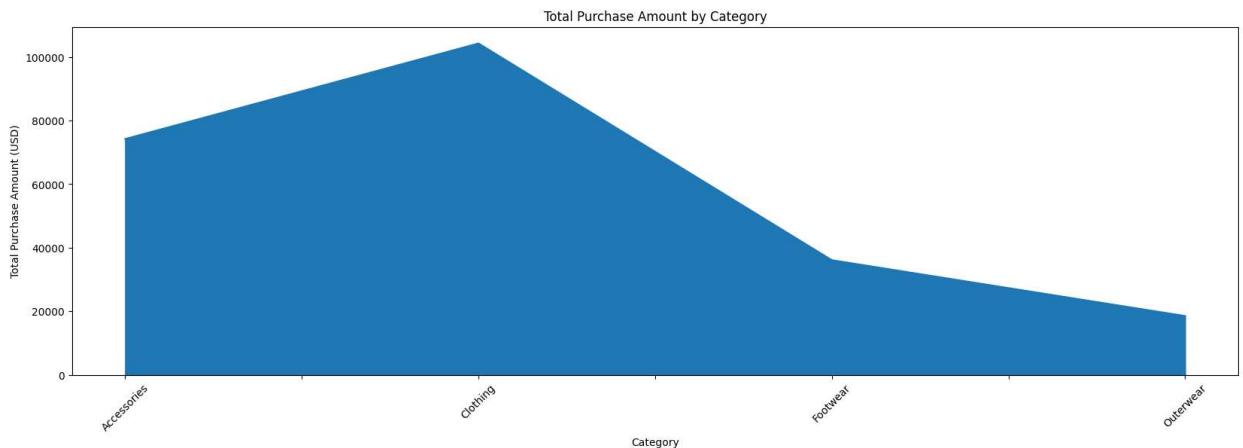
Bar Plot of Payment Method by Season

```
In [262...]  
plt.figure(figsize=(20, 6))  
sns.barplot(x='Season', y='Purchase Amount (USD)', hue='Payment Method', data=df)  
plt.title('Purchase Amount by Season and Payment Method')  
plt.xlabel('Season')  
plt.ylabel('Purchase Amount (USD)')  
plt.xticks(rotation = 0)  
plt.show()
```



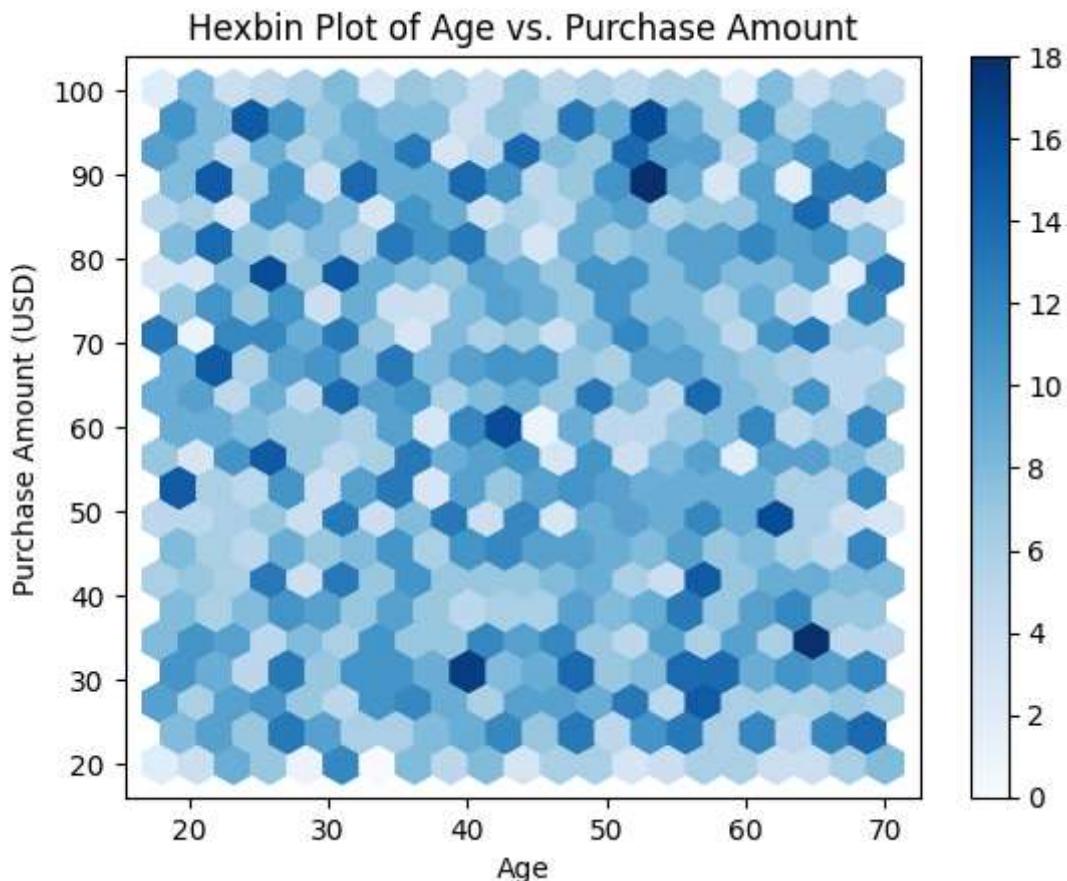
Area Plot of Total Purchase Amount by Category

```
In [263...]  
plt.figure(figsize=(20, 6))  
category_purchase_total = df.groupby('Category')['Purchase Amount (USD)'].sum()  
  
category_purchase_total.plot(kind='area')  
plt.title('Total Purchase Amount by Category')  
plt.xlabel('Category')  
plt.ylabel('Total Purchase Amount (USD)')  
plt.xticks(rotation=45)  
plt.show()
```



Hexbin Plot of Age vs. Purchase Amount

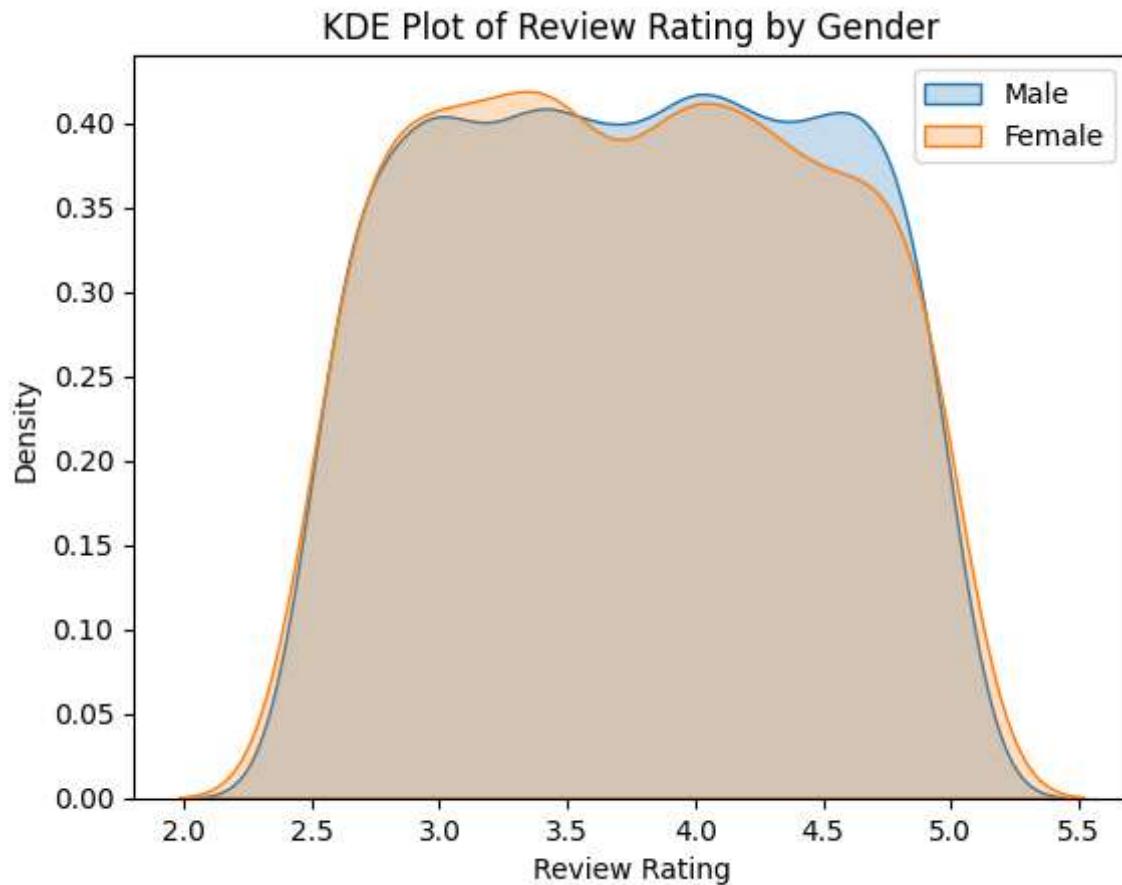
```
In [264]: plt.hexbin(df['Age'], df['Purchase Amount (USD)'], gridsize=20, cmap='Blues')
plt.title('Hexbin Plot of Age vs. Purchase Amount')
plt.xlabel('Age')
plt.ylabel('Purchase Amount (USD)')
plt.colorbar()
plt.show()
```



KDE Plot of Review Rating by Gender

```
In [265]: sns.kdeplot(df[df['Gender'] == 'Male']['Review Rating'], label='Male', shade=True)
sns.kdeplot(df[df['Gender'] == 'Female']['Review Rating'], label='Female', shade=True)
plt.title('KDE Plot of Review Rating by Gender')
plt.xlabel('Review Rating')
plt.ylabel('Density')
```

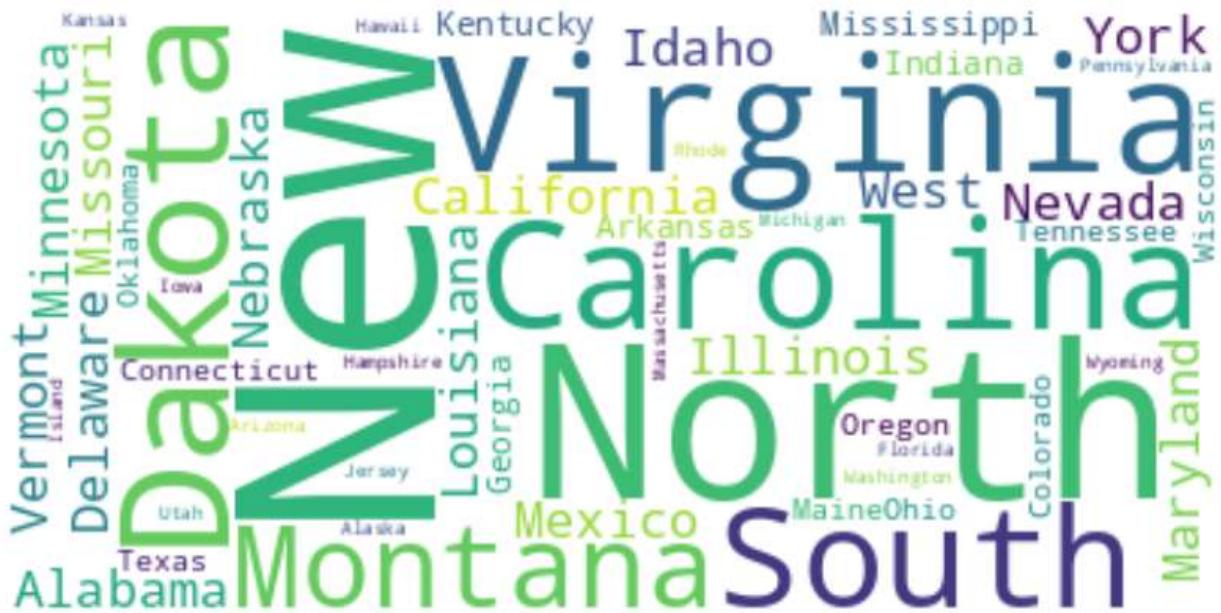
```
plt.legend()  
plt.show()
```



In [266]:

```
text = " ".join(title for title in df["Location"])  
word_cloud = WordCloud(collocations = False, background_color = 'white').generate(text)  
plt.figure(figsize = (20, 5))  
plt.imshow(word_cloud, interpolation = 'bilinear')  
plt.axis("off")  
plt.title('WordCloud for Location\n', fontsize = 12, color = "#FF0000")  
plt.show()
```

WorldCloud for Location



```
In [267...]: text = " ".join(title for title in df["Payment Method"])
word_cloud = WordCloud(collocations = False, background_color = 'white').generate(text)
plt.figure(figsize = (20, 5))
plt.imshow(word_cloud, interpolation = 'bilinear')
plt.axis("off")
plt.title('WorldCloud for Payment Method\n', fontsize = 12, color = "#FF0000")
plt.show()
```

## WorldCloud for Payment Method

