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

Shopping Trend Analysis

```
df2=pd.read_csv('/content/shopping_trends.csv')
df2.head()
```

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season	Review Rating	Subscription Status	Payment Method	Shipping Type	D
0	1	55	Male	Blouse	Clothing	53	Kentucky	L	Gray	Winter	3.1	Yes	Credit Card	Express	
1	2	19	Male	Sweater	Clothing	64	Maine	L	Maroon	Winter	3.1	Yes	Bank Transfer	Express	

Next steps: Generate code with df2 View recommended plots

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3900 entries, 0 to 3899
Data columns (total 19 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  Payment Method            3900 non-null   object
 13  Shipping Type             3900 non-null   object
 14  Discount Applied          3900 non-null   object
 15  Promo Code Used           3900 non-null   object
 16  Previous Purchases        3900 non-null   int64
 17  Preferred Payment Method  3900 non-null   object
 18  Frequency of Purchases    3900 non-null   object
dtypes: float64(1), int64(4), object(14)
memory usage: 579.0+ KB
```

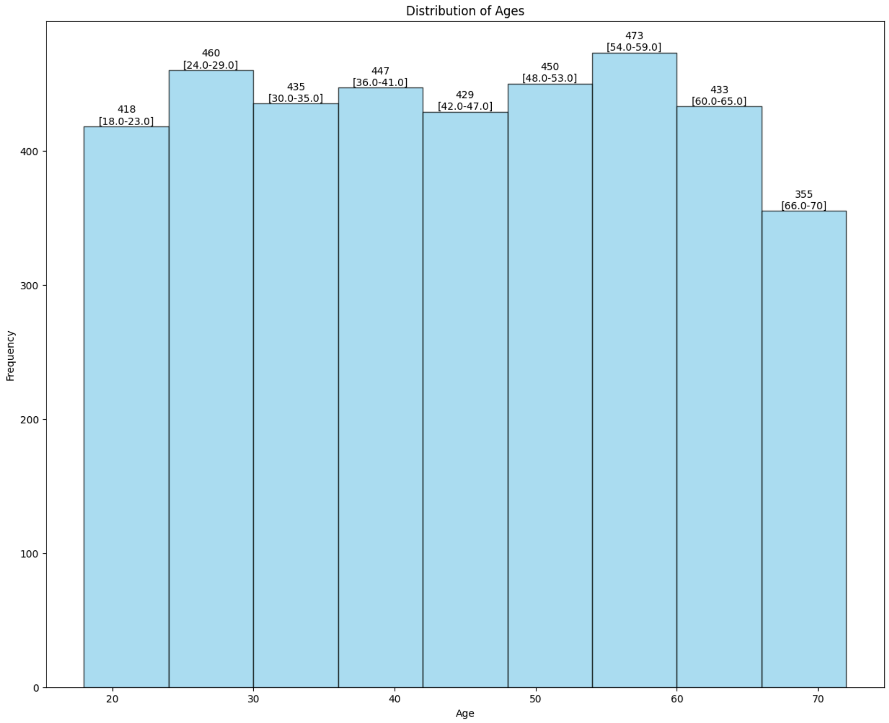
```
df2.describe()
```

	Customer ID	Age	Purchase Amount (USD)	Review Rating	Previous Purchases
count	3900.000000	3900.000000	3900.000000	3900.000000	3900.000000
mean	1950.500000	44.068462	59.764359	3.749949	25.351538
std	1125.977353	15.207589	23.685392	0.716223	14.447125
min	1.000000	18.000000	20.000000	2.500000	1.000000
25%	975.750000	31.000000	39.000000	3.100000	13.000000
50%	1950.500000	44.000000	60.000000	3.700000	25.000000
75%	2925.250000	57.000000	81.000000	4.400000	38.000000
max	3900.000000	70.000000	100.000000	5.000000	50.000000

```
df2['Age'].describe()
```

```
count    3900.000000
mean       44.068462
std        15.207589
min        18.000000
25%        31.000000
50%        44.000000
75%        57.000000
max        70.000000
Name: Age, dtype: float64
```

```
# Determine the range of ages and calculate the bin width
min_age = df2['Age'].min()
max_age = df2['Age'].max()
bin_width = int((max_age - min_age) / 10) + 1  # Assuming 10 bins

# Create bins with whole numbers and equal interval
bins = range(min_age, max_age + bin_width, bin_width)

# Create a histogram of ages
plt.figure(figsize=(15, 12))
counts, bins, _ = plt.hist(df2['Age'], bins=bins, color='skyblue', edgecolor='black', alpha=0.7, density=False)
plt.title('Distribution of Ages')
plt.xlabel('Age')
plt.ylabel('Frequency')

# Add text labels for each bin
for i in range(len(bins) - 1):
    bin_start = bins[i]
    bin_end = bins[i + 1] - 1 if i < len(bins) - 2 else max_age
    plt.text(bins[i] + (bins[i + 1] - bins[i]) / 2, counts[i], f"{int(counts[i])}\n[{bin_start}-{bin_end}]", ha='center', va='bottom')

plt.show()
```



```
df2['Location'].value_counts()
```

```
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
```

```
# Group the data by location and payment method
location_payment_group = df2.groupby(['Location', 'Payment Method']).size().reset_index(name='Frequency')

# Find the preferred payment method for each location
preferred_payment_by_location = location_payment_group.loc[location_payment_group.groupby('Location')['Frequency'].idxmax()]

# Display the results
print("Preferred payment method based on location:")
print(preferred_payment_by_location)
```

```
Preferred payment method based on location:
           Location Payment Method  Frequency
0           Alabama  Bank Transfer         22
11           Alaska          Venmo         14
15          Arizona     Debit Card         13
21         Arkansas     Debit Card         16
24       California  Bank Transfer         18
31         Colorado           Cash         17
41      Connecticut          Venmo         17
47         Delaware          Venmo         20
53          Florida          Venmo         15
59          Georgia          Venmo         20
64           Hawaii         PayPal         14
70            Idaho         PayPal         19
74         Illinois    Credit Card         19
82          Indiana         PayPal         18
86             Iowa    Credit Card         15
94           Kansas         PayPal         15
98         Kentucky    Credit Card         17
105       Louisiana     Debit Card         16
112           Maine         PayPal         15
115        Maryland           Cash         23
121   Massachusetts           Cash         17
126        Michigan  Bank Transfer         15
133       Minnesota           Cash         18
140     Mississippi    Credit Card         17
148        Missouri         PayPal         19
153         Montana     Debit Card         20
161        Nebraska          Venmo         18
162          Nevada  Bank Transfer         18
170   New Hampshire    Credit Card         18
175      New Jersey           Cash         14
181      New Mexico           Cash         22
187        New York           Cash         18
195  North Carolina     Debit Card         20
200    North Dakota    Credit Card         17
204            Ohio  Bank Transfer         21
210        Oklahoma  Bank Transfer         17
218          Oregon    Credit Card         19
227    Pennsylvania          Venmo         17
228    Rhode Island  Bank Transfer         16
237  South Carolina     Debit Card         15
```

```
244    South Dakota         PayPal         16
248       Tennessee    Credit Card         20
252           Texas  Bank Transfer         16
260            Utah    Credit Card         20
269         Vermont          Venmo         18
274        Virginia         PayPal         20
278      Washington    Credit Card         17
283   West Virginia           Cash         17
288       Wisconsin  Bank Transfer         18
299         Wyoming          Venmo         17
```

```
payment_counts = df2['Payment Method'].value_counts()
payment_counts
```

```
Payment Method
Credit Card      696
Venmo            653
Cash             648
PayPal           638
Debit Card       633
Bank Transfer    632
Name: count, dtype: int64
```

```
gender=df2['Gender'].value_counts()
gender
```

```
Gender
Male      2652
Female    1248
Name: count, dtype: int64
```

```
proportion_of_gender=gender/df2['Gender'].count()
proportion_of_gender
```

```
Gender
Male      0.68
Female    0.32
Name: count, dtype: float64
```

```
# Grouping the data by 'Item Purchased' and calculating the total number of purchases for each item
item_purchases = df2.groupby('Item Purchased').size().reset_index(name='Total Purchases')

# Sorting the items by total purchases in descending order
most_purchased_items = item_purchases.sort_values(by='Total Purchases', ascending=False)

# Displaying the top most purchased items
print("Most purchased items:")
print(most_purchased_items.head())
```

```
Most purchased items:
   Item Purchased  Total Purchases
12        Jewelry              171
2          Blouse              171
13          Pants              171
16          Shirt              169
5           Dress              166
```

```
jewlery_prices=df2[df2["Item Purchased"]=="Jewelry"]["Purchase Amount (USD)"]
jewlery_prices.sum()
```

```
10010
```

```
'''
# Filter the DataFrame to include only rows where the item is a scarf
scarf_prices = df2[df2['Item Purchased'] == 'Scarf']['Purchase Amount (USD)']

# Display the purchase prices for all scarfs
print("Purchase prices for all scarfs:")
print(scarf_prices)

'''
```

```
blouse_prices=df2[df2["Item Purchased"]=="Blouse"]["Purchase Amount (USD)"]
blouse_prices.sum()
```

```
10410
```

```
pants_prices=df2[df2["Item Purchased"]=="Pants"]["Purchase Amount (USD)"]
pants_prices.sum()
```

```
10090
```

```
shirt_prices=df2[df2["Item Purchased"]=="Shirt"]["Purchase Amount (USD)"]
shirt_prices.sum()
```

```
10332
```

```
dress_prices=df2[df2["Item Purchased"]=="Dress"]["Purchase Amount (USD)"]
dress_prices.sum()
```

```
10320
```

```
sum(df2["Purchase Amount (USD)"])
```

```
233081
```

```
blouse_prices.sum()/233081
```

```
0.044662585109897415
```

```
df2['Category'].value_counts()
```

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

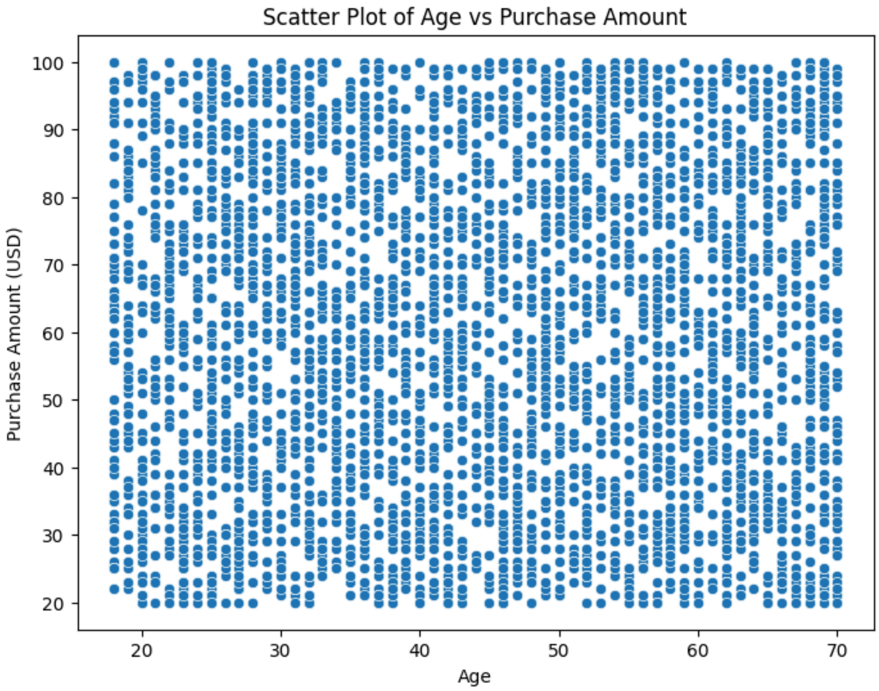
```
preferred_payment=df2['Preferred Payment Method'].value_counts()
preferred_payment
```

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

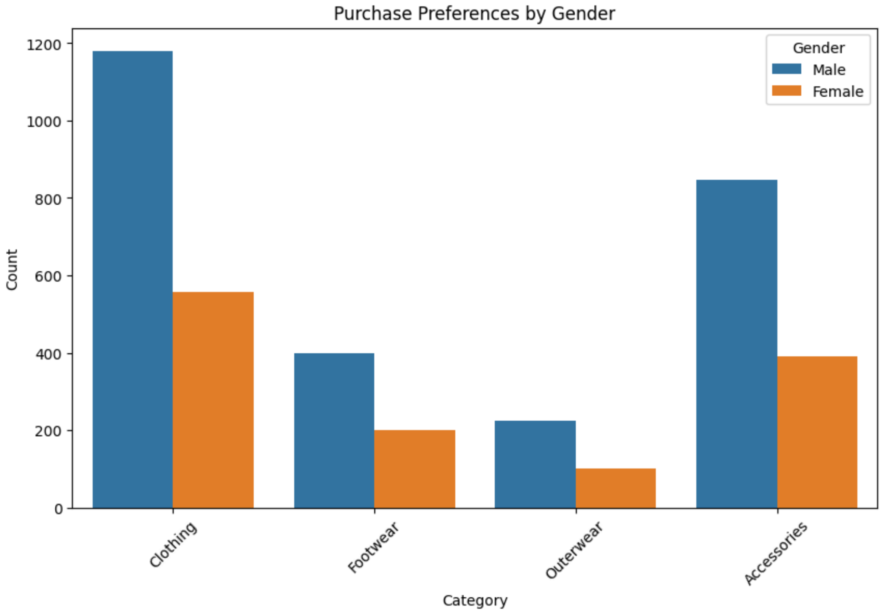
```
correlation = df2['Age'].corr(df2['Purchase Amount (USD)'])
correlation
```

```
-0.01042364737868652
```

```
plt.figure(figsize=(8, 6))
sns.scatterplot(x='Age', y='Purchase Amount (USD)', data=df2)
plt.title('Scatter Plot of Age vs Purchase Amount')
plt.xlabel('Age')
plt.ylabel('Purchase Amount (USD)')
plt.show()

#The scatter plot shows
```



```
plt.figure(figsize=(10, 6))
sns.countplot(x='Category', hue='Gender', data=df2)
plt.title('Purchase Preferences by Gender')
plt.xlabel('Category')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.legend(title='Gender')
plt.show()
```



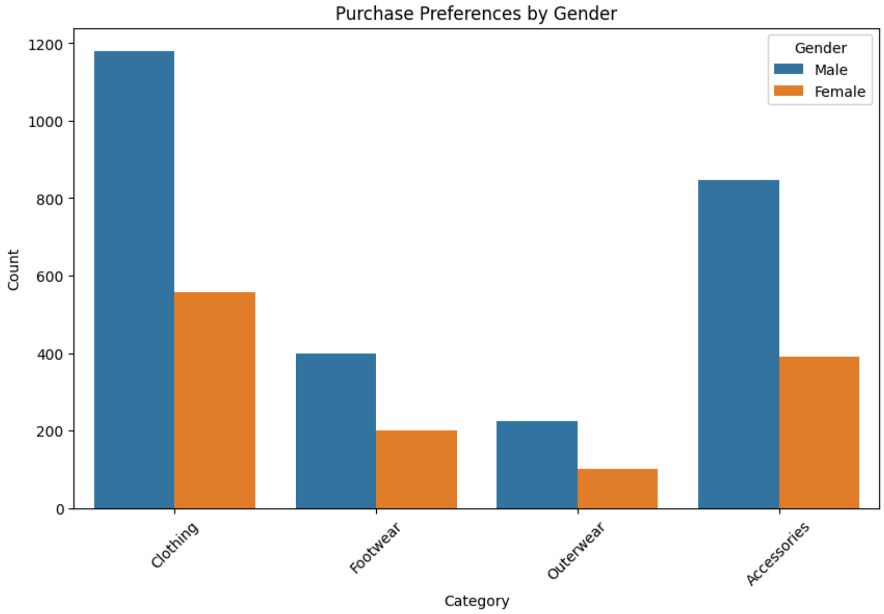
```
season_counts=df2['Season'].value_counts()
season_counts
```

```
Season
Spring    999
Fall      975
Winter    971
Summer    955
Name: count, dtype: int64
```

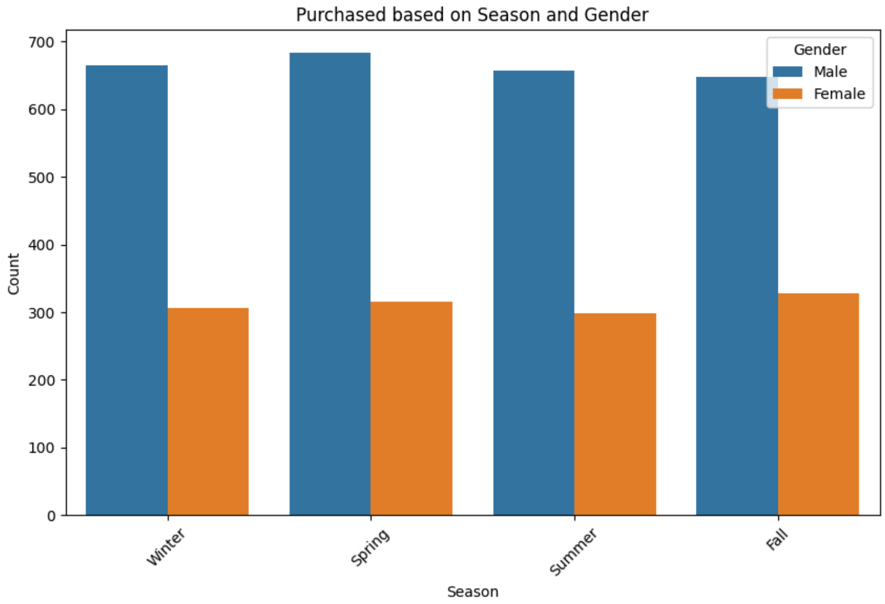
```
# Counting the number of purchases based on gender and season
purchase_counts = df2.groupby(['Gender', 'Season']).size()

# Summing the purchase amounts based on gender and season
purchase_amounts = df2.groupby(['Gender', 'Season'])['Purchase Amount (USD)'].sum()

purchase_amounts
```

```
Gender  Season
Female  Fall      20193
        Spring    18578
        Summer    18060
        Winter    18360
Male    Fall      39825
        Spring    40101
        Summer    37717
        Winter    40247
Name: Purchase Amount (USD), dtype: int64
```

```
plt.figure(figsize=(10, 6))
sns.countplot(x='Category', hue='Gender', data=df2)
plt.title('Purchase Preferences by Gender')
plt.xlabel('Category')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.legend(title='Gender')
plt.show()
```



```
plt.figure(figsize=(10,6))
sns.countplot(x='Season', hue='Gender', data=df2)
plt.title('Purchased based on Season and Gender')
plt.xlabel('Season')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.legend(title='Gender')
plt.show()
```



```
category_revenue=df2.groupby('Category')['Purchase Amount (USD)'].sum()
category_revenue
```

```
Category
Accessories     74200
Clothing       104264
Footwear        36093
Outerwear       18524
Name: Purchase Amount (USD), dtype: int64
```

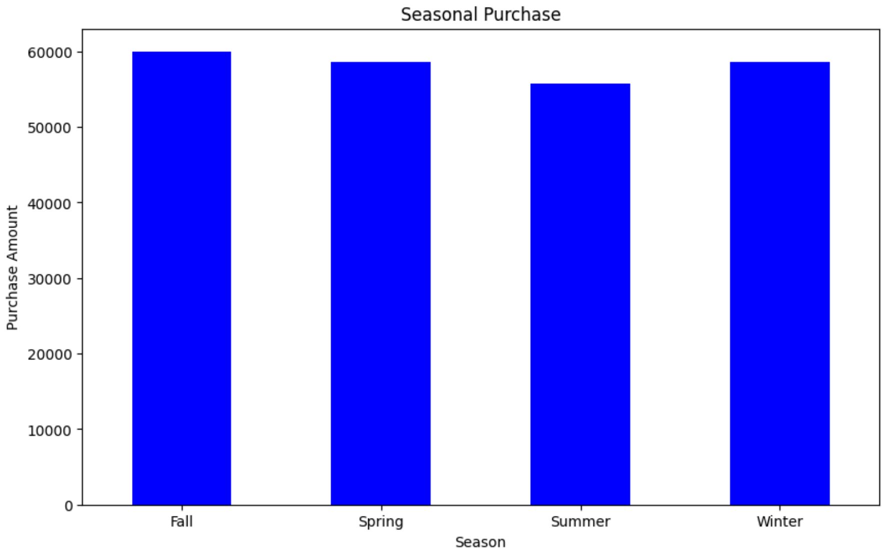
```
average_category_purchase_amount=df2.groupby('Category')['Purchase Amount (USD)'].mean()
average_category_purchase_amount
```

```
Category
Accessories    59.838710
Clothing       60.025331
Footwear       60.255426
Outerwear      57.172840
Name: Purchase Amount (USD), dtype: float64
```

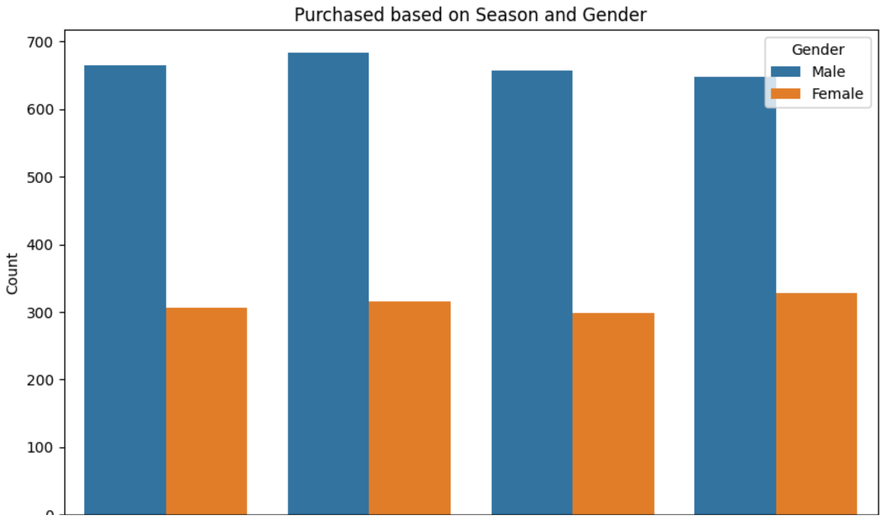
```
seasonal_purchase=df2.groupby('Season')['Purchase Amount (USD)'].sum()
seasonal_purchase
```

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

```
plt.figure(figsize=(10,6))
seasonal_purchase.plot(kind='bar', color='blue')
plt.title("Seasonal Purchase")
plt.xlabel("Season")
plt.ylabel("Purchase Amount")
plt.xticks(rotation=0)
plt.show()
```



```
plt.figure(figsize=(10,6))
sns.countplot(x='Season', hue='Gender', data=df2)
plt.title('Purchased based on Season and Gender')
plt.xlabel('Season')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.legend(title='Gender')
plt.show()
```


Purchased based on Season and Gender
Count
Gender
🟦 Male
🟧 Female