In [1]:
```
## import library
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
## load data set
ST = pd.read_csv('shopping_trends.csv')
ST
```

Out[2]:

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color
0	1	55	Male	Blouse	Clothing	53	Kentucky	L	Gray
1	2	19	Male	Sweater	Clothing	64	Maine	L	Maroon
2	3	50	Male	Jeans	Clothing	73	Massachusetts	S	Maroon
3	4	21	Male	Sandals	Footwear	90	Rhode Island	M	Maroon
4	5	45	Male	Blouse	Clothing	49	Oregon	M	Turquoise
...	...	...	...	...	...	...	...	...	...
3895	3896	40	Female	Hoodie	Clothing	28	Virginia	L	Turquoise
3896	3897	52	Female	Backpack	Accessories	49	Iowa	L	White
3897	3898	46	Female	Belt	Accessories	33	New Jersey	L	Green
3898	3899	44	Female	Shoes	Footwear	77	Minnesota	S	Brown
3899	3900	52	Female	Handbag	Accessories	81	California	M	Beige

3900 rows × 19 columns

In [3]:
```
##Review info for the csv
ST.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3900 entries, 0 to 3899
Data columns (total 19 columns):
Customer ID                 3900 non-null int64
Age                         3900 non-null int64
Gender                      3900 non-null object
Item Purchased              3900 non-null object
Category                    3900 non-null object
Purchase Amount (USD)       3900 non-null int64
Location                    3900 non-null object
Size                        3900 non-null object
Color                       3900 non-null object
Season                      3900 non-null object
Review Rating               3900 non-null float64
Subscription Status         3900 non-null object
Payment Method              3900 non-null object
Shipping Type               3900 non-null object
Discount Applied            3900 non-null object
Promo Code Used             3900 non-null object
Previous Purchases          3900 non-null int64
Preferred Payment Method    3900 non-null object
Frequency of Purchases      3900 non-null object
dtypes: float64(1), int64(4), object(14)
memory usage: 579.0+ KB
```

In [4]:
```
##Count NaN values of whole DataFrame
ST.isnull().sum()
```

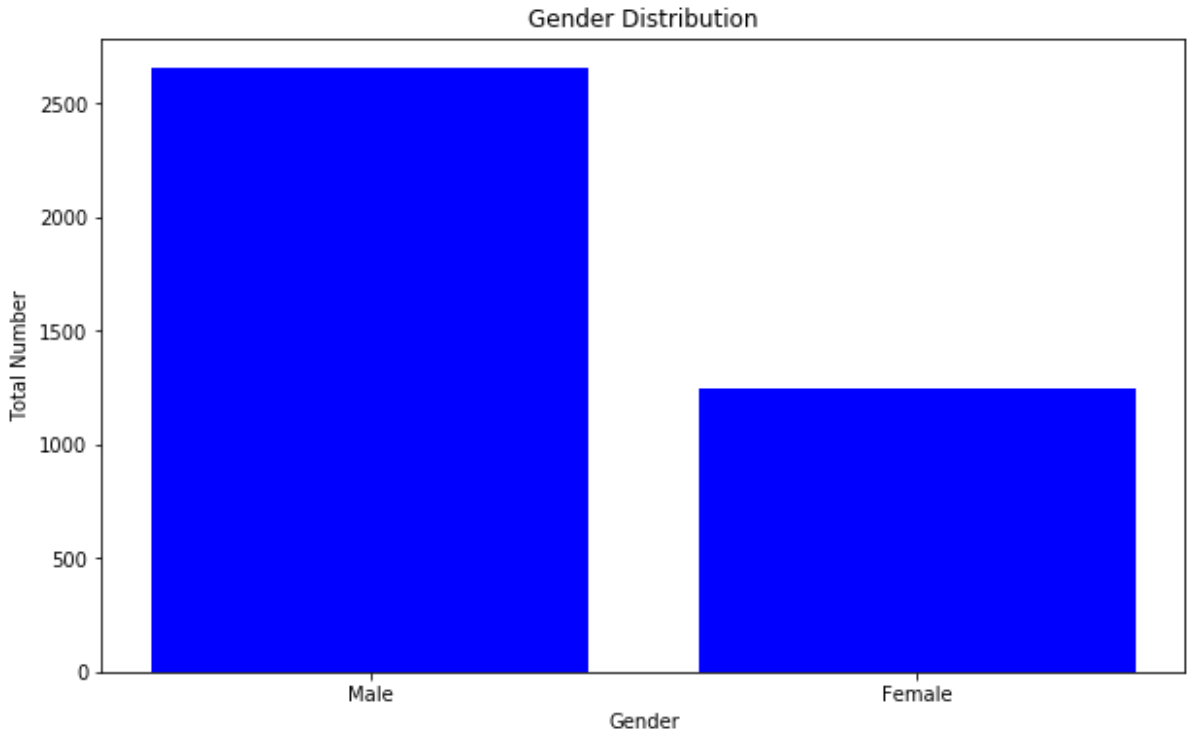
Out[4]:
```
Customer ID                 0
Age                         0
Gender                      0
Item Purchased              0
Category                    0
Purchase Amount (USD)       0
Location                    0
Size                        0
Color                       0
Season                      0
Review Rating               0
Subscription Status         0
Payment Method              0
Shipping Type               0
Discount Applied            0
Promo Code Used             0
Previous Purchases          0
Preferred Payment Method    0
Frequency of Purchases      0
dtype: int64
```

In [5]:
```
ST.describe()
```

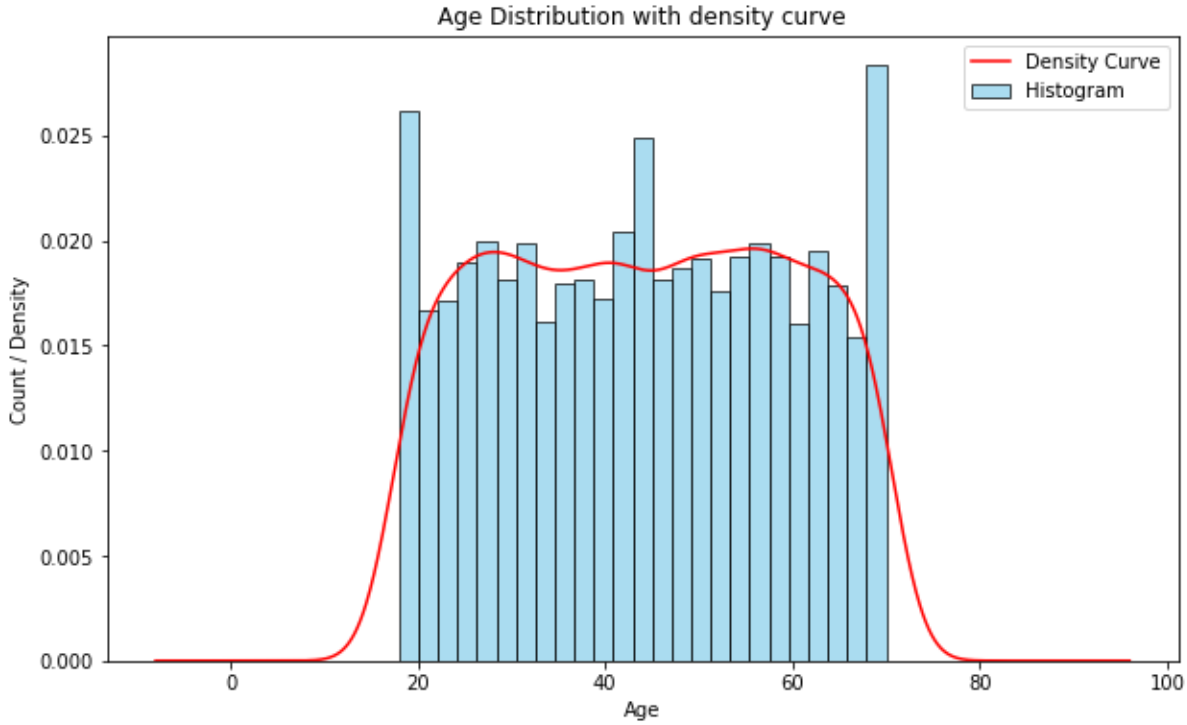
Out[5]:

	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

In [6]:
```
plt.figure(figsize=(10,6))
A = ST['Gender'].value_counts()
bars = plt.bar(A.index, A.values, color='Blue')

plt.xlabel('Gender')
plt.ylabel('Total Number')
plt.title('Gender Distribution')

plt.show()
```



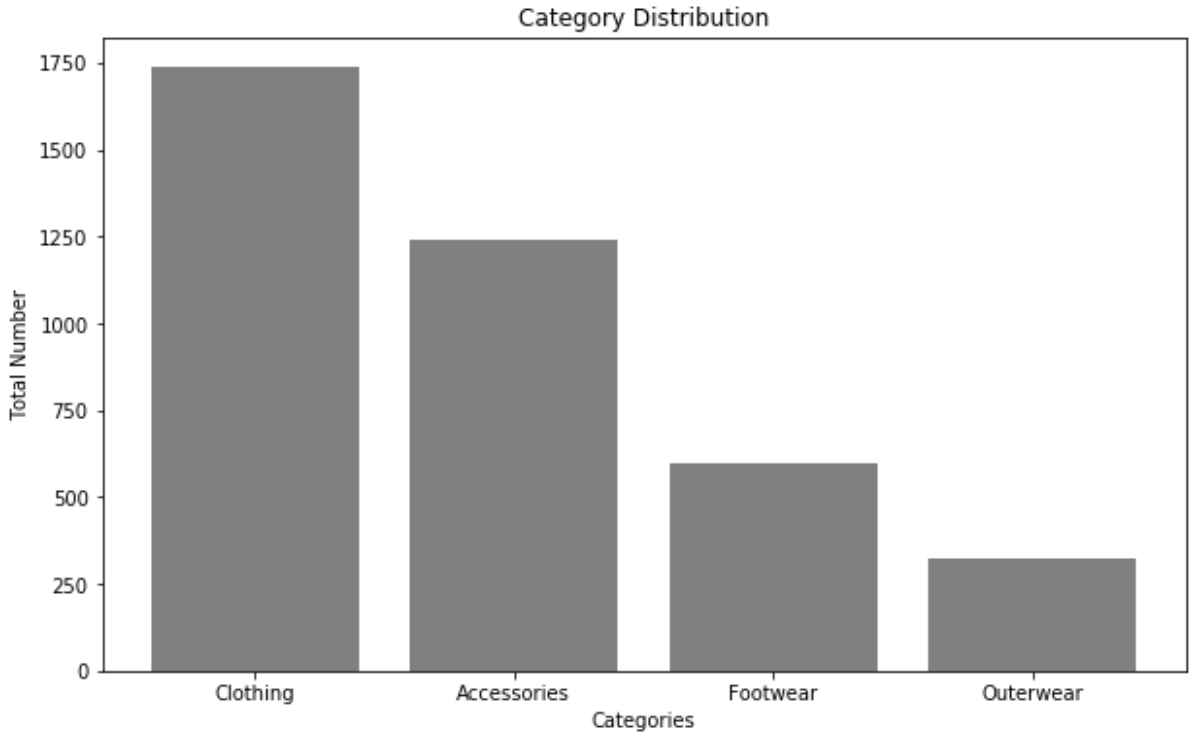
In [7]:
```
##kernel density estimate (KDE) plot is a method for visualizing the distribut
ion of observations in a dataset, analogous to a histogram
plt.figure(figsize=(10, 6))
plt.hist(ST['Age'],edgecolor = 'black',alpha=0.7,bins=25,color = 'skyblue',den
sity=True)
ST['Age'].plot(kind='kde', color = 'red')

plt.xlabel('Age')
plt.ylabel('Count / Density')
plt.title('Age Distribution with density curve')
plt.legend(['Density Curve', 'Histogram'])

plt.show()
```



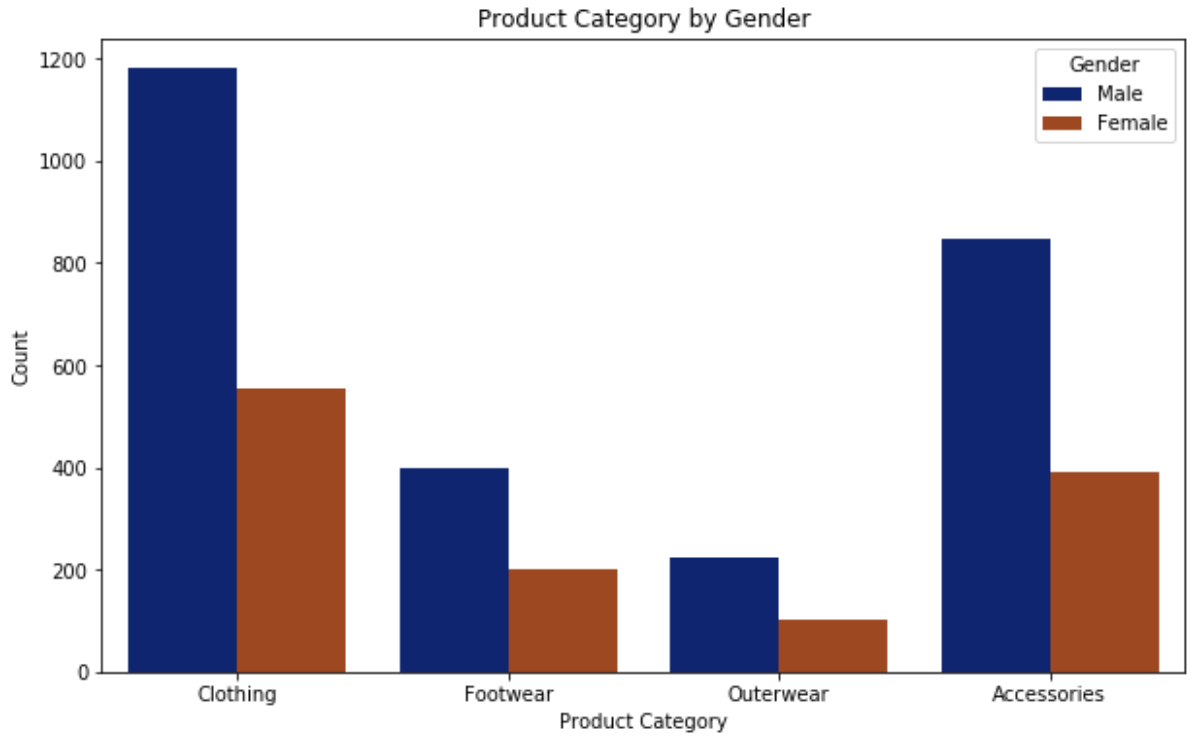
In [8]:
```
# Category
ST["Category"].value_counts()
```

Out[8]:
```
Clothing       1737
Accessories    1240
Footwear        599
Outerwear       324
Name: Category, dtype: int64
```

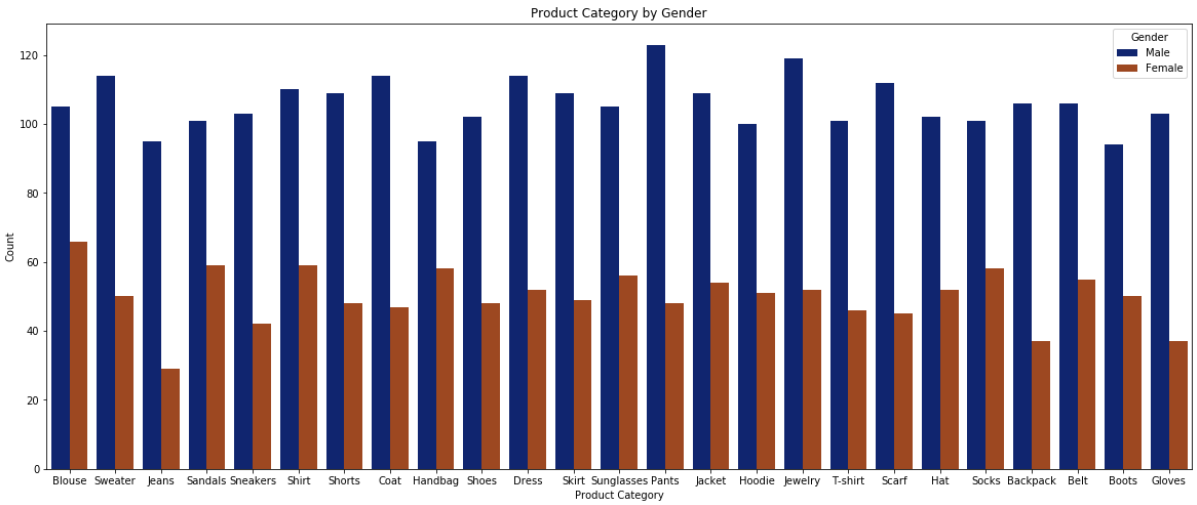
In [9]:
```
plt.figure(figsize=(10, 6))
c = ST['Category'].value_counts()
bar=plt.bar(c.index,c.values,color= 'gray')

plt.xlabel('Categories')
plt.ylabel('Total Number')
plt.title('Category Distribution')

plt.show()
```



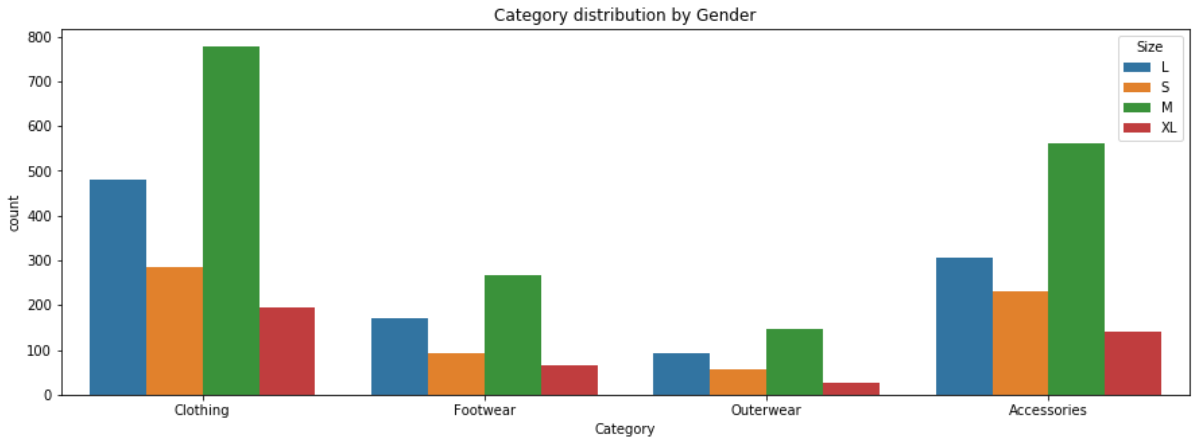
In [10]:
```
# Create a count plot to visualize the relationship
plt.figure(figsize=(10, 6))
sns.countplot(data=ST, x='Category', hue='Gender',palette='dark')
plt.title('Product Category by Gender')
plt.xlabel('Product Category')
plt.ylabel('Count')
plt.legend(title='Gender', labels=ST['Gender'].unique())

# Show the plot
plt.show()
```



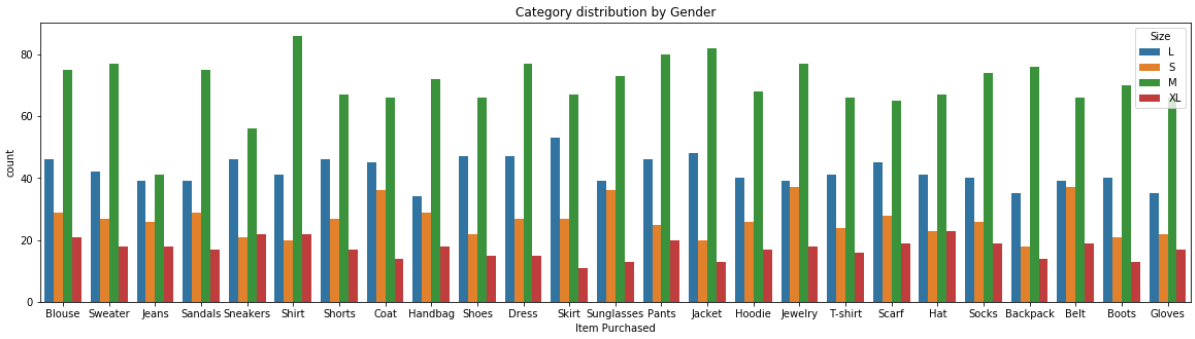
In [21]:
```
# Create a count plot to visualize the relationship
plt.figure(figsize=(20, 8))
sns.countplot(data=ST, x='Item Purchased', hue='Gender',palette='dark')
plt.title('Product Category by Gender')
plt.xlabel('Product Category')
plt.ylabel('Count')
plt.legend(title='Gender', labels=ST['Gender'].unique())

# Show the plot
plt.show()
```



In [22]:
```
plt.figure(figsize=(15,5))
sns.countplot(data=ST,x='Category' ,hue='Size')
plt.title('Category distribution by Gender')
plt.show()
```

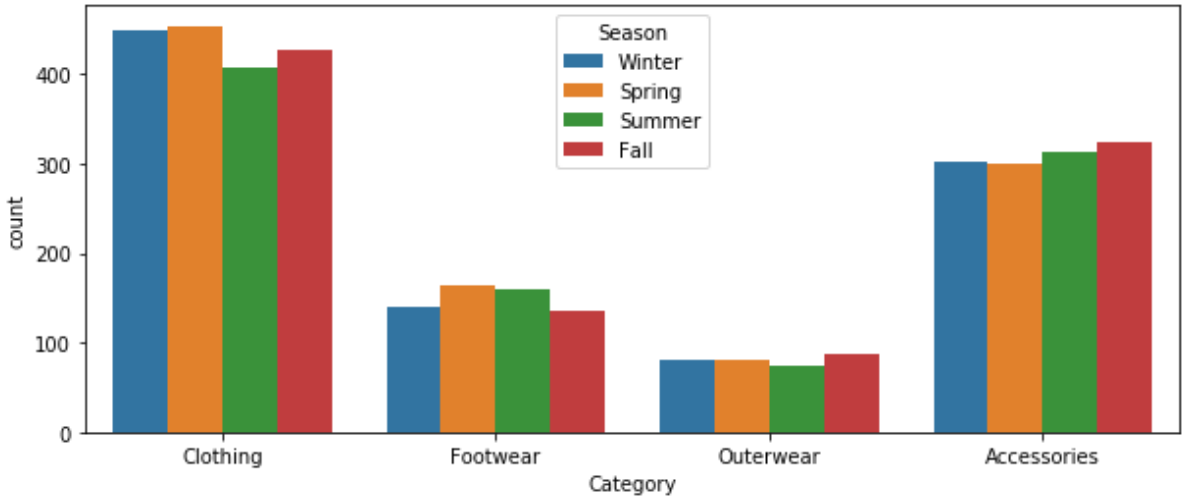


In [24]:
```
plt.figure(figsize=(20,5))
sns.countplot(data=ST,x='Item Purchased' ,hue='Size')
plt.title('Category distribution by Gender')
plt.show()
```

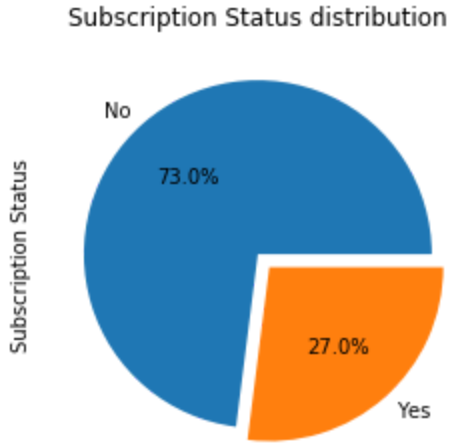


In [25]:
```
plt.figure(figsize=(10,4))
sns.countplot(data=ST,x='Category',hue='Season')
```

Out[25]: <matplotlib.axes._subplots.AxesSubplot at 0x1bf9b151c88>

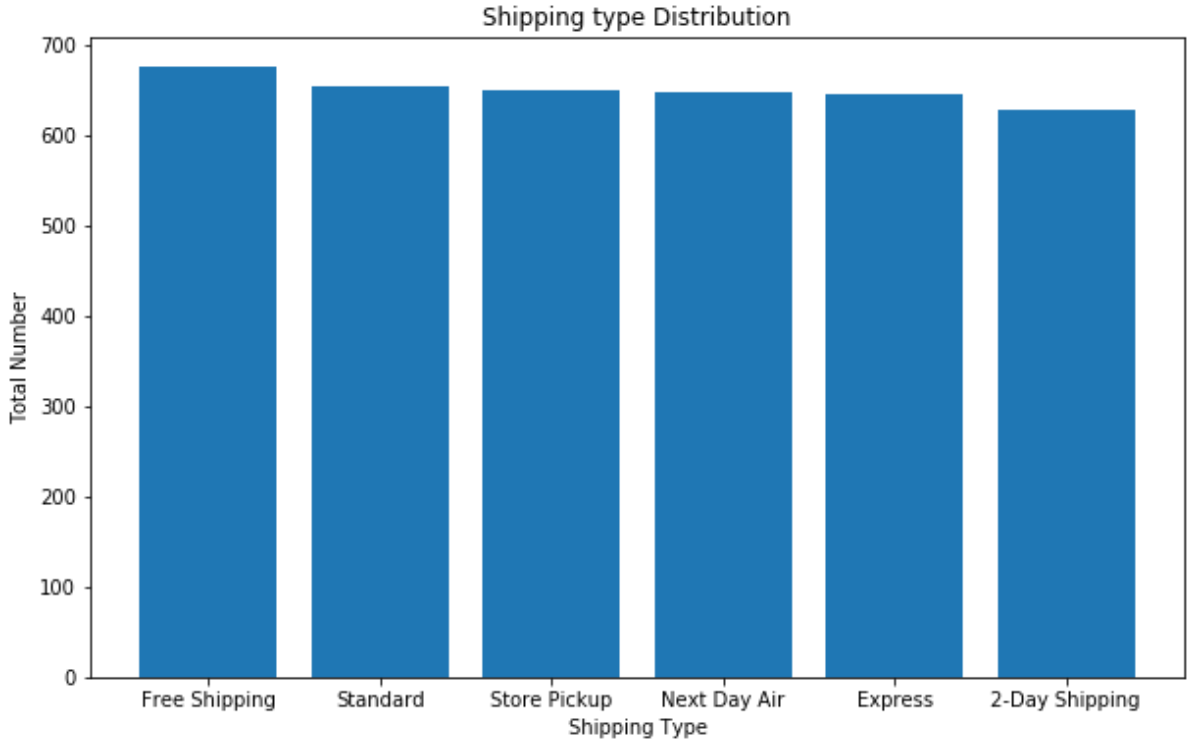


In [30]:
```
count=ST['Subscription Status'].value_counts()
count.plot(kind='pie', explode=(0,0.1),autopct='%1.1f%%')
plt.title('Subscription Status distribution')
plt.show()
```



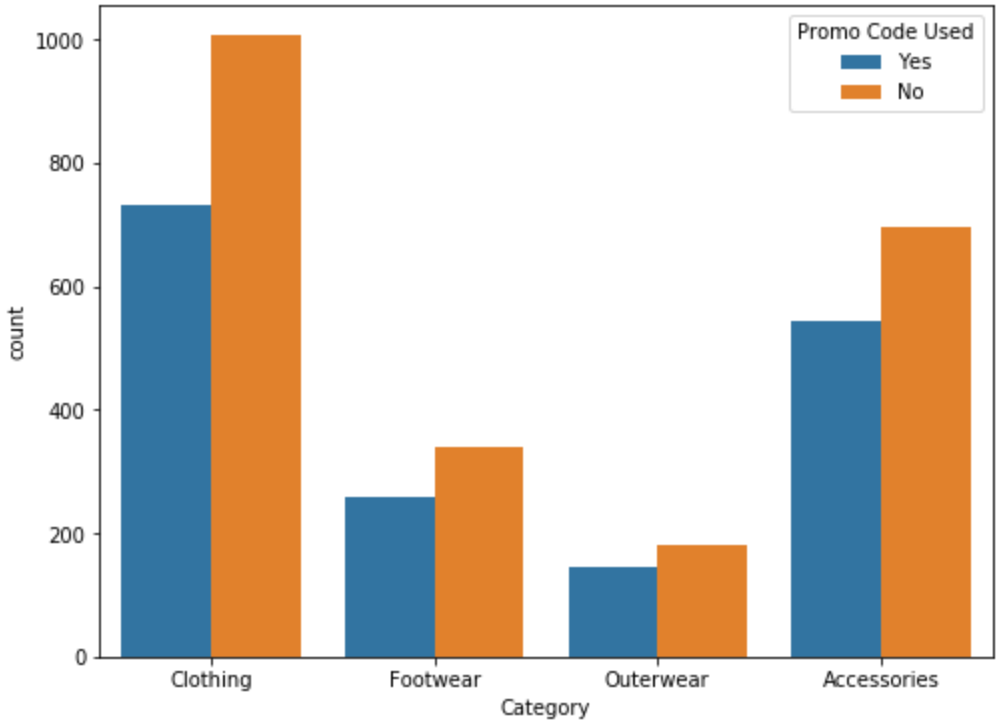
In [34]:
```
plt.figure(figsize=(10, 6))
c=ST['Shipping Type'].value_counts()
bar=plt.bar(c.index,c.values)

plt.xlabel('Shipping Type')
plt.ylabel('Total Number')
plt.title('Shipping type Distribution')
```

Out[34]:
```
Text(0.5, 1.0, 'Shipping type Distribution')
```



In [39]:
```
plt.figure(figsize=(8,6))
sns.countplot(data=ST,x='Category',hue='Promo Code Used')
```

Out[39]: <matplotlib.axes._subplots.AxesSubplot at 0x1bf9b141f48>



In [43]:
```
#Average age of the customer
average_age = ST['Age'].mean()
print("Average Age:", average_age)
```

```
Average Age: 44.06846153846154
```

In [42]:
```
#Total purchase amount for each category

TotalPur = ST.groupby('Category')['Purchase Amount (USD)'].sum()
print("Total purchases for each category:", TotalPur)
```

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

In [45]:
```
#The most common payment method used by customers
most_common_payment_method = ST['Payment Method'].mode()[0]
print("Most Common Payment Method:", most_common_payment_method)
```

```
Most Common Payment Method: Credit Card
```

In [47]:
```
# How many customers have opted for the Subscription
subscription_count = ST[ST['Subscription Status'] == 'Yes']['Customer ID'].cou
nt()
print("Number of Customers with Subscription: ", subscription_count)
```

```
Number of Customers with Subscription:  1053
```

In [49]:
```
# What is the most common season for purchases?
most_common_season = ST['Season'].mode()[0]
print("Most Common Season for Purchases:", most_common_season)
```

```
Most Common Season for Purchases: Spring
```

In [51]:
```
#How many customers used a promo code for their purchase
promo_code_count = ST[ST['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
```

In [52]:
```
#What is the maximum and minimum review rating in the dataset?
max_review_rating = ST['Review Rating'].max()
min_review_rating = ST['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
```

In [53]:
```
# What is the most common shipping type for customers with a review rating abo
ve 4 ?
common_shipping_high_rating = ST[ST['Review Rating'] > 4]['Shipping Type'].mod
e()[0]
print("Most Common Shipping Type for High Review Ratings:", common_shipping_hi
gh_rating)
```

```
Most Common Shipping Type for High Review Ratings: Standard
```

In [54]:
```
#How many customers have made more than 30 previous purchases ?
customers_above_30_previous_purchases = ST[ST['Previous Purchases'] > 30]['Cus
tomer ID'].count()
print("Number of Customers with more than 30 Previous Purchases:", customers_a
bove_30_previous_purchases)
```

```
Number of Customers with more than 30 Previous Purchases: 1549
```