In [ ]:

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

In [6]:

```
df = pd.read_csv('Amazon Sale Report.csv', encoding= 'unicode escape')
```

In [7]:

```
df.shape
```

Out[7]:

```
(128976, 21)
```

In [8]:

```
df.head()
```

Out[8]:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	...	currency	Amount	ship-ci
0	0	405-8078784-5731545	04-30-22	Cancelled	Merchant	Amazon.in	Standard	T-shirt	S	On the Way	...	INR	647.62	MUMB
1	1	171-9198151-1101146	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	3XL	Shipped	...	INR	406.00	BENGALUR
2	2	404-0687676-7273146	04-30-22	Shipped	Amazon	Amazon.in	Expedited	Shirt	XL	Shipped	...	INR	329.00	NAVI MUMB
3	3	403-9615377-8133951	04-30-22	Cancelled	Merchant	Amazon.in	Standard	Blazzer	L	On the Way	...	INR	753.33	PUDUCHERR
4	4	407-1069790-7240320	04-30-22	Shipped	Amazon	Amazon.in	Expedited	Trousers	3XL	Shipped	...	INR	574.00	CHENN

5 rows × 21 columns

In [9]:

```
df.tail()
```

Out[9]:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	...	currency	Amount	shi
128971	128970	406-6001380-7673107	05-31-22	Shipped	Amazon	Amazon.in	Expedited	Shirt	XL	Shipped	...	INR	517.0	HYDER
128972	128971	402-9551604-7544318	05-31-22	Shipped	Amazon	Amazon.in	Expedited	T-shirt	M	Shipped	...	INR	999.0	GURU
128973	128972	407-9547469-3152358	05-31-22	Shipped	Amazon	Amazon.in	Expedited	Blazzer	XXL	Shipped	...	INR	690.0	HYDER
128974	128973	402-6184140-0545956	05-31-22	Shipped	Amazon	Amazon.in	Expedited	T-shirt	XS	Shipped	...	INR	1199.0	
128975	128974	408-7436540-8728312	05-31-22	Shipped	Amazon	Amazon.in	Expedited	T-shirt	S	Shipped	...	INR	696.0	

5 rows × 21 columns

In [10]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 128976 entries, 0 to 128975
Data columns (total 21 columns):
 #   Column              Non-Null Count   Dtype  
---  ------              --------------   -----  
 0   index               128976 non-null  int64  
 1   Order ID            128976 non-null  object 
 2   Date                128976 non-null  object 
 3   Status              128976 non-null  object 
 4   Fulfilment          128976 non-null  object 
 5   Sales Channel       128976 non-null  object 
 6   ship-service-level  128976 non-null  object 
 7   Category            128976 non-null  object 
 8   Size                128976 non-null  object 
 9   Courier Status      128976 non-null  object 
 10  Qty                 128976 non-null  int64  
 11  currency            121176 non-null  object 
 12  Amount              121176 non-null  float64
 13  ship-city           128941 non-null  object 
 14  ship-state          128941 non-null  object 
 15  ship-postal-code    128941 non-null  float64
 16  ship-country        128941 non-null  object 
 17  B2B                 128976 non-null  bool   
 18  fulfilled-by        39263 non-null   object 
 19  New                 0 non-null       float64
 20  PendingS            0 non-null       float64
dtypes: bool(1), float64(4), int64(2), object(14)
memory usage: 19.8+ MB
```

In [12]:

```
#Removing blank columns
df.drop(['New','PendingS'], axis = 1, inplace = True)
```

In [13]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 128976 entries, 0 to 128975
Data columns (total 19 columns):
 #   Column              Non-Null Count   Dtype  
---  ------              --------------   -----  
 0   index               128976 non-null  int64  
 1   Order ID            128976 non-null  object 
 2   Date                128976 non-null  object 
 3   Status              128976 non-null  object 
 4   Fulfilment          128976 non-null  object 
 5   Sales Channel       128976 non-null  object 
 6   ship-service-level  128976 non-null  object 
 7   Category            128976 non-null  object 
 8   Size                128976 non-null  object 
 9   Courier Status      128976 non-null  object 
 10  Qty                 128976 non-null  int64  
 11  currency            121176 non-null  object 
 12  Amount              121176 non-null  float64
 13  ship-city           128941 non-null  object 
 14  ship-state          128941 non-null  object 
 15  ship-postal-code    128941 non-null  float64
 16  ship-country        128941 non-null  object 
 17  B2B                 128976 non-null  bool   
 18  fulfilled-by        39263 non-null   object 
dtypes: bool(1), float64(2), int64(2), object(14)
memory usage: 17.8+ MB
```

In [15]:

```
#checking null values
pd.isnull(df).sum()
```

Out[15]:

```
index                     0
Order ID                  0
Date                      0
Status                    0
Fulfilment                0
Sales Channel             0
ship-service-level        0
Category                  0
Size                      0
Courier Status            0
Qty                       0
currency               7800
Amount                 7800
ship-city                35
ship-state               35
ship-postal-code         35
ship-country             35
B2B                       0
fulfilled-by          89713
dtype: int64
```

In [17]:
```
df.shape
```

Out[17]:
```
(128976, 19)
```

In [18]:
```
#dropping null values
df.dropna(inplace = True)
```

In [19]:
```
df.shape
```

Out[19]:
```
(37514, 19)
```

In [21]:
```
df.columns
```

Out[21]:
```
Index(['index', 'Order ID', 'Date', 'Status', 'Fulfilment', 'Sales Channel',
       'ship-service-level', 'Category', 'Size', 'Courier Status', 'Qty',
       'currency', 'Amount', 'ship-city', 'ship-state', 'ship-postal-code',
       'ship-country', 'B2B', 'fulfilled-by'],
      dtype='object')
```

In [38]:
```
#changing data type
df['ship-postal-code'] = df['ship-postal-code'].astype('int')
```

In [39]:
```
df['ship-postal-code'].dtype
```

Out[39]:
```
dtype('int32')
```

In [40]:
```
df['Date'] = pd.to_datetime (df['Date'])
```

In [41]:
```
df.columns
```

Out[41]:
```
Index(['index', 'Order ID', 'Date', 'Status', 'Fulfilment', 'Sales Channel',
       'ship-service-level', 'Category', 'Size', 'Courier Status', 'Quantity',
       'currency', 'Amount', 'ship-city', 'ship-state', 'ship-postal-code',
       'ship-country', 'B2B', 'fulfilled-by'],
      dtype='object')
```

In [52]:
```
#renaming the columns
df.rename(columns = {'Qty': 'Quantity'}, inplace = True)
```

In [53]:
```
df.describe()
```

Out[53]:

	index	Date	Quantity	Amount	ship-postal-code
count	37514.000000	37514	37514.000000	37514.000000	37514.000000
mean	60953.809858	2022-05-11 07:56:47.303939840	0.867383	646.553960	463291.552754
min	0.000000	2022-03-31 00:00:00	0.000000	0.000000	110001.000000
25%	27235.250000	2022-04-20 00:00:00	1.000000	458.000000	370465.000000
50%	63470.500000	2022-05-09 00:00:00	1.000000	629.000000	500019.000000
75%	91790.750000	2022-06-01 00:00:00	1.000000	771.000000	600042.000000
max	128891.000000	2022-06-29 00:00:00	5.000000	5495.000000	989898.000000
std	36844.853039	NaN	0.354160	279.952414	194550.425637

In [54]:
```
df.describe(include = 'object')
```

Out[54]:

	Order ID	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	currency	ship-city	ship-state
count	37514	37514	37514	37514	37514	37514	37514	37514	37514	37514	37514
unique	34664	11	1	1	1	8	11	3	1	4698	58
top	171-5057375-2831560	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	T-shirt	M	Shipped	INR	BENGALURU	MAHARASHTRA
freq	12	28741	37514	37514	37514	14062	6806	31859	37514	2839	6236

In [55]:
```
#usinf describe for specific columns
df[['Quantity','Amount']].describe()
```

Out[55]:

	Quantity	Amount
count	37514.000000	37514.000000
mean	0.867383	646.553960
std	0.354160	279.952414
min	0.000000	0.000000
25%	1.000000	458.000000
50%	1.000000	629.000000
75%	1.000000	771.000000
max	5.000000	5495.000000

Exploratory Data Analysis

In [56]:

```
df.columns
```

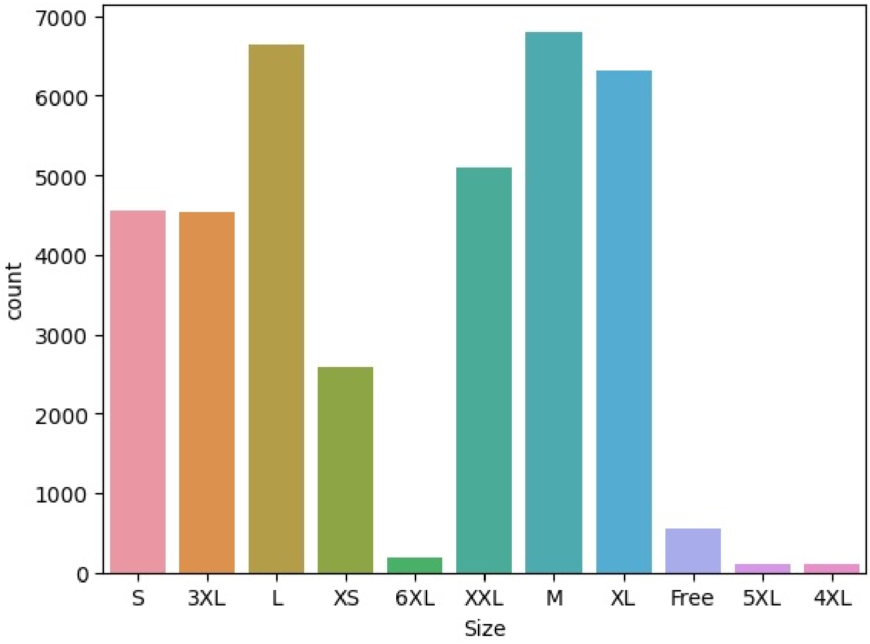
Out[56]:

```
Index(['index', 'Order ID', 'Date', 'Status', 'Fulfilment', 'Sales Channel',
       'ship-service-level', 'Category', 'Size', 'Courier Status', 'Quantity',
       'currency', 'Amount', 'ship-city', 'ship-state', 'ship-postal-code',
       'ship-country', 'B2B', 'fulfilled-by'],
      dtype='object')
```

Size

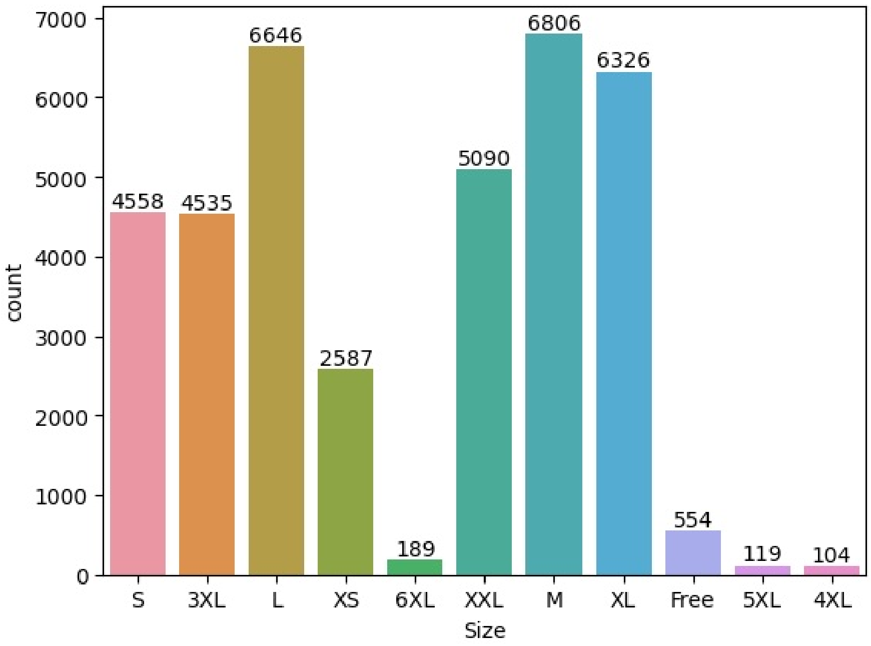
In [57]:

```
ax = sns.countplot(x = 'Size', data=df)
```



In [58]:

```
ax = sns.countplot(x = 'Size', data=df)

for bars in ax.containers:
    ax.bar_label(bars)
```



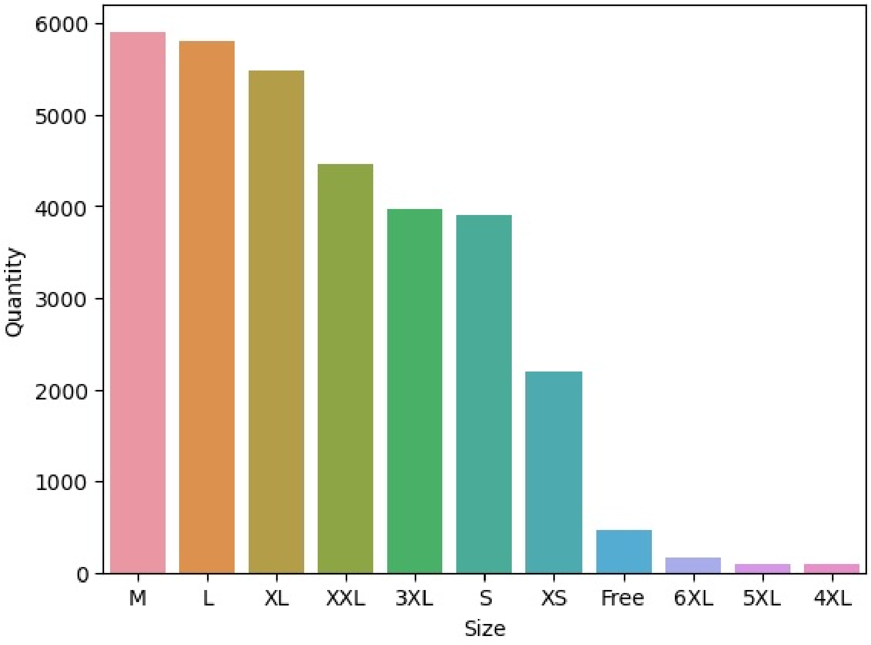
In [59]:
```
df.groupby(['Size'], as_index = False)['Quantity'].sum().sort_values(by = 'Quantity',ascending = False)
```

Out[59]:

	Size	Quantity
6	M	5905
5	L	5795
8	XL	5481
10	XXL	4465
0	3XL	3972
7	S	3896
9	XS	2191
4	Free	467
3	6XL	170
2	5XL	104
1	4XL	93

In [62]:
```
S_Quantity = df.groupby(['Size'], as_index = False)['Quantity'].sum().sort_values(by = 'Quantity',ascending = Fa

sns.barplot(x = 'Size', y ='Quantity', data = S_Quantity )
```

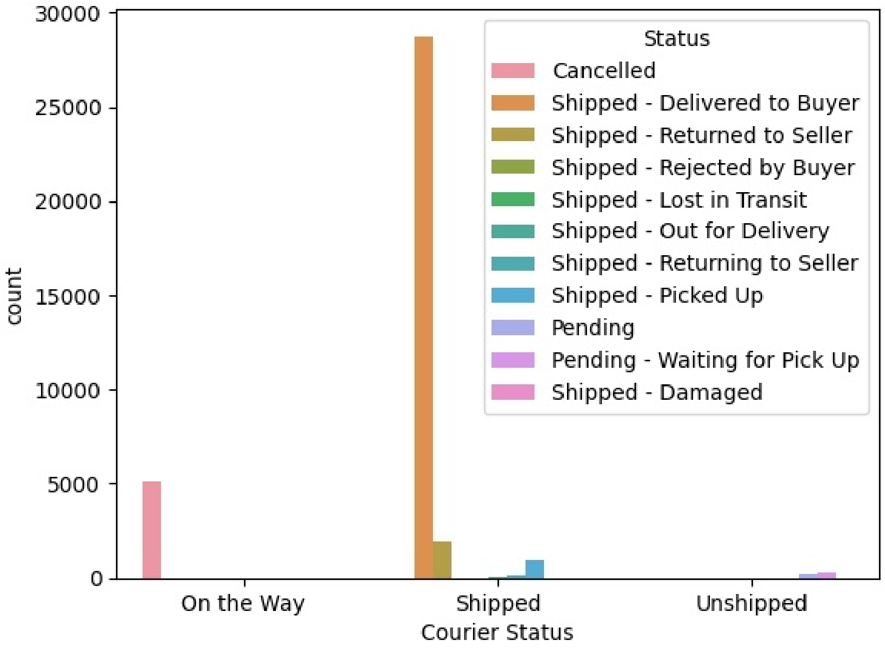
Out[62]:
```
<Axes: xlabel='Size', ylabel='Quantity'>
```



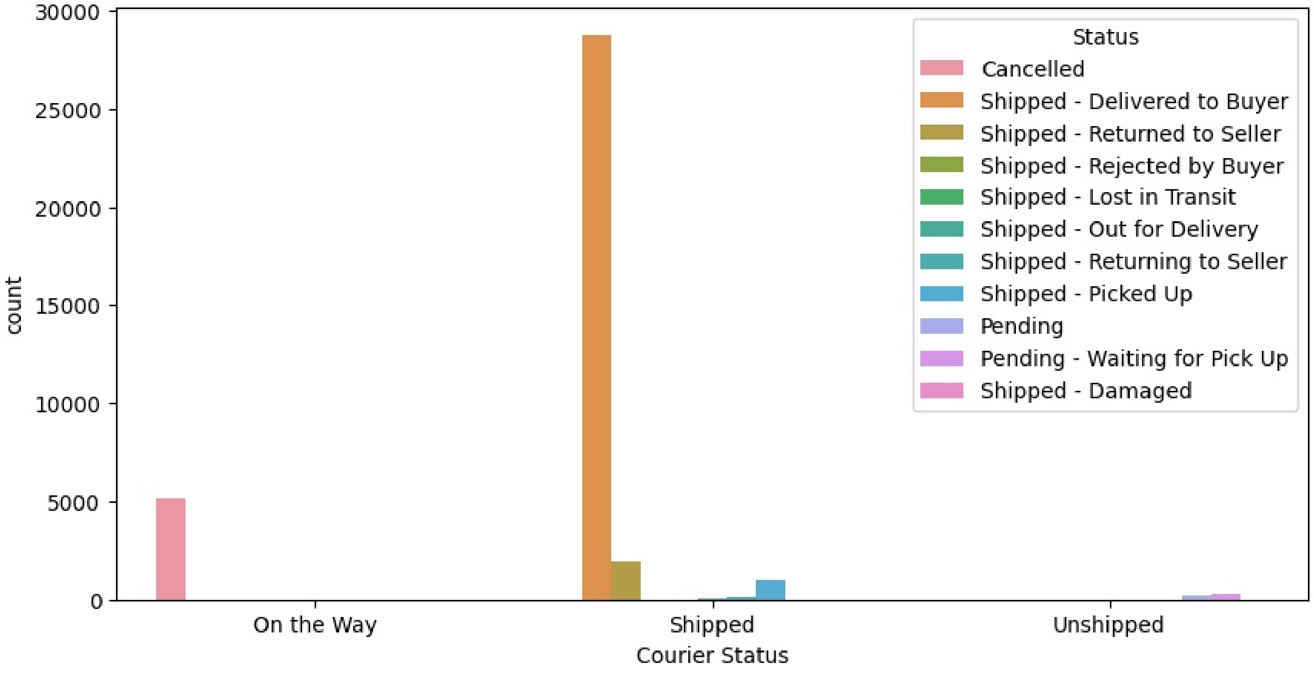
Courier Status

```
In [63]: sns.countplot(data = df, x = 'Courier Status',hue = 'Status')
```

```
Out[63]: <Axes: xlabel='Courier Status', ylabel='count'>
```

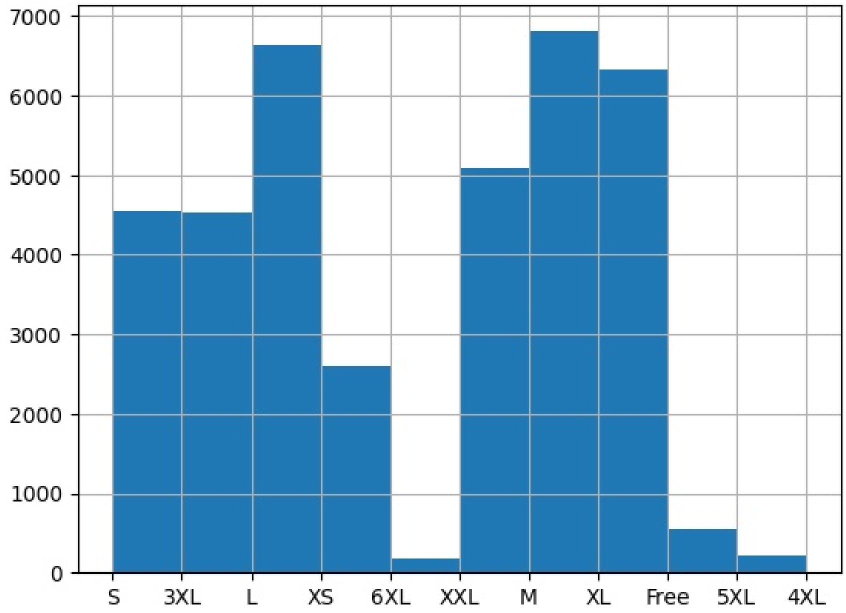


```
In [66]: plt.figure(figsize = (10,5))

         ax = sns.countplot(data = df, x = 'Courier Status', hue ='Status')
         plt.show()
```

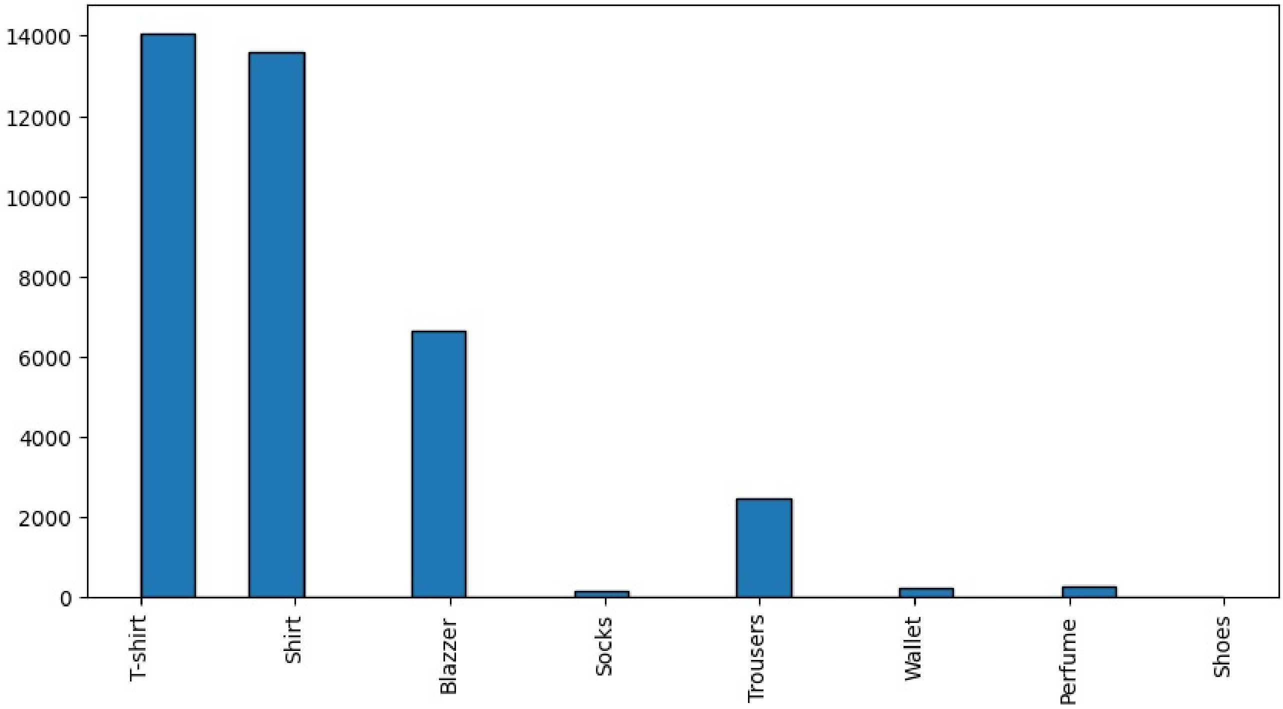


```
In [67]: df['Size'].hist()
```

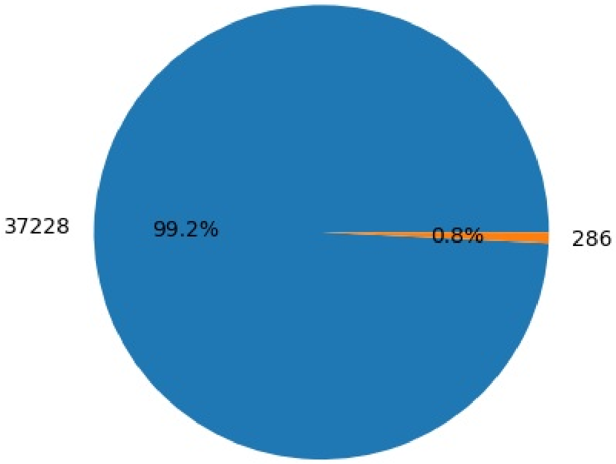
```
Out[67]: <Axes: >
```



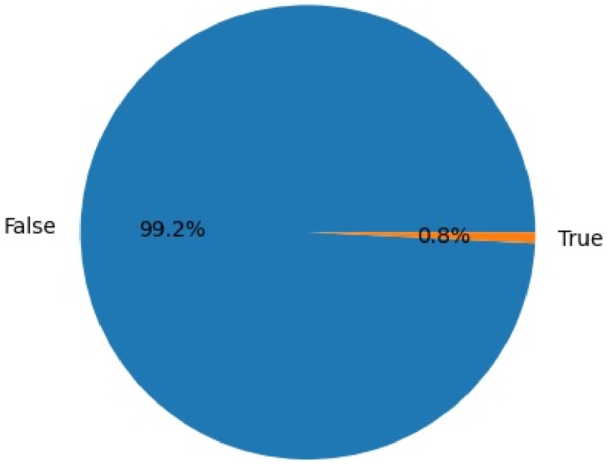
In [69]:
```
df['Category'] = df['Category'].astype(str)
column_data = df['Category']
plt.figure(figsize = (10,5))
plt.hist(column_data, bins = 20, edgecolor = 'Black')
plt.xticks(rotation = 90)
plt.show()
```



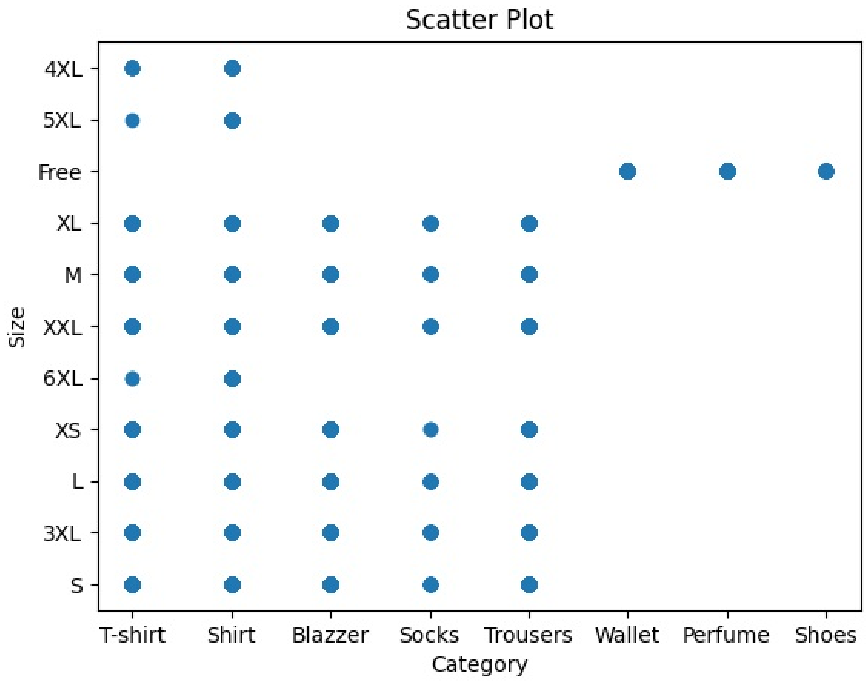
In [70]:
```
#checking B2B Data by using pie chart
B2B_Check = df['B2B'].value_counts()

#plotting pie chart
plt.pie(B2B_Check, labels = B2B_Check, autopct = '%1.1f%%')
plt.show()
```



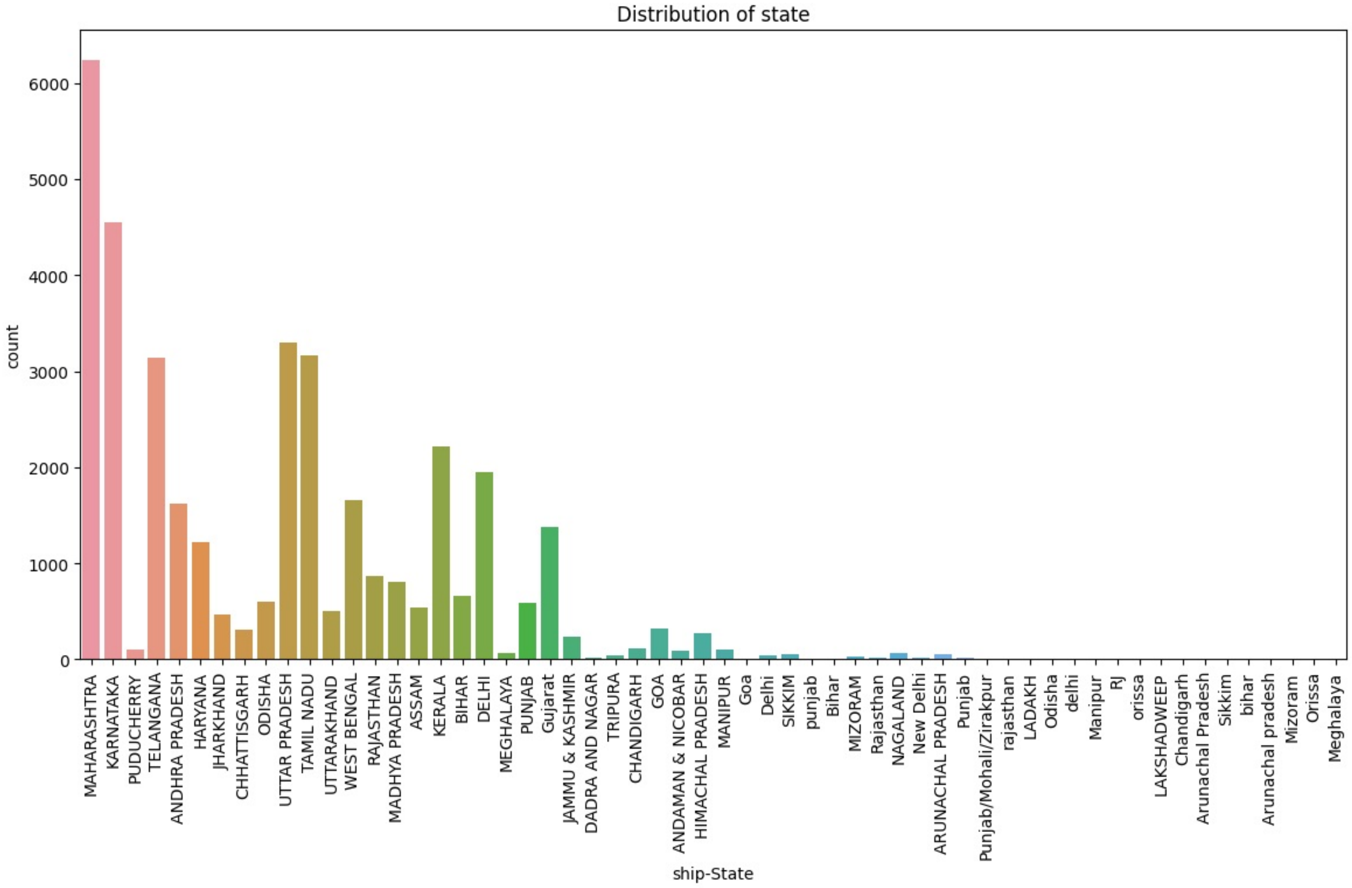
In [71]:
```
B2B_Check = df['B2B'].value_counts()
plt.pie(B2B_Check, labels = B2B_Check.index, autopct = '%1.1f%%')
plt.show()
```



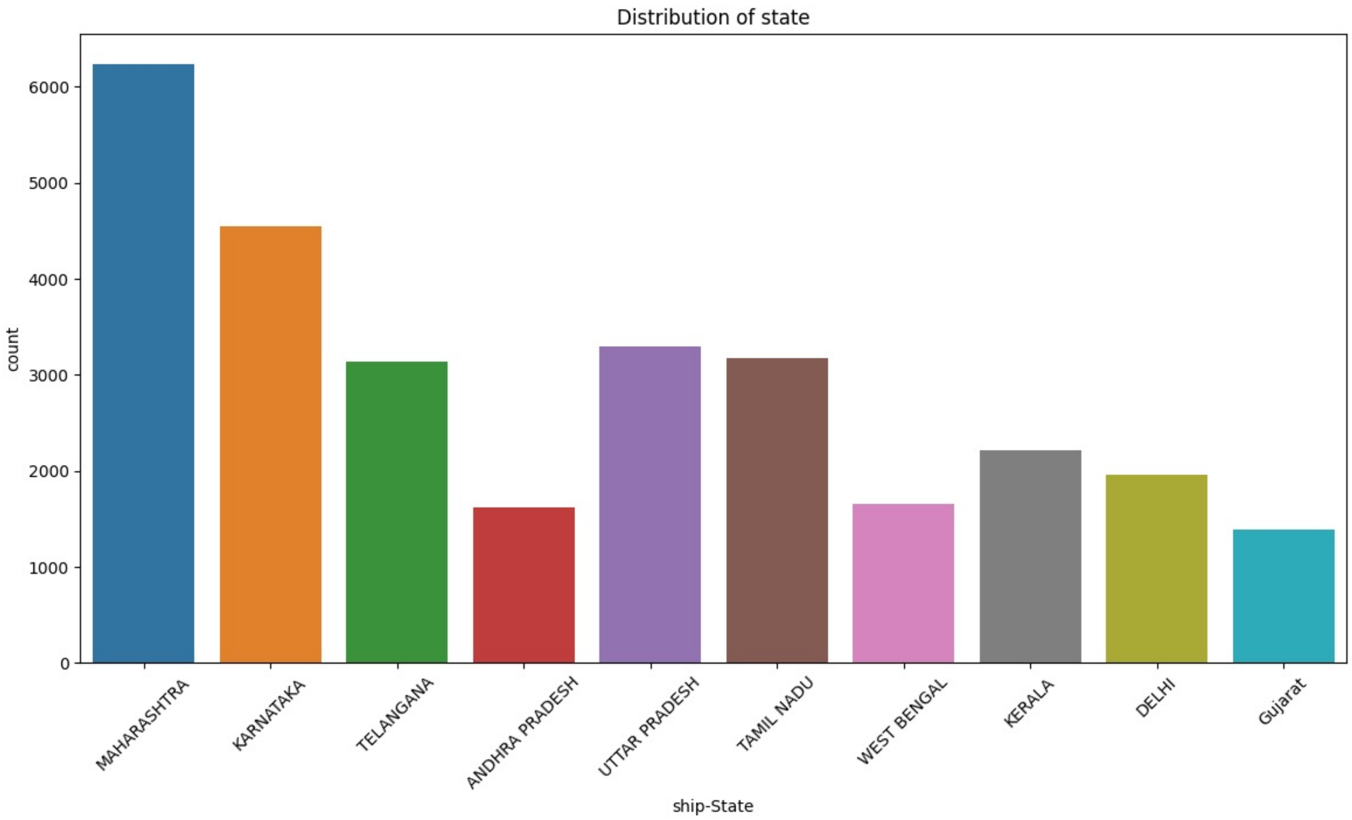
In [72]:
```
#preparing data for scatter plot
x_data = df['Category']
y_data = df['Size']

#plotting the scatter plot
plt.scatter(x_data, y_data)
plt.xlabel('Category')
plt.ylabel('Size')
plt.title('Scatter Plot')
plt.show()
```



```
In [77]: #plotting the count of cities by state
         plt.figure(figsize =(14,7))
         sns.countplot(data = df, x = 'ship-state')
         plt.xlabel('ship-State')
         plt.ylabel('count')
         plt.title('Distribution of state')
         plt.xticks(rotation = 90)
         plt.show()
```



```
In [81]: Top_10_State = df['ship-state'].value_counts().head(10)
         plt.figure(figsize =(14,7))
         sns.countplot(data = df[df['ship-state'].isin(Top_10_State.index)], x = 'ship-state')
         plt.xlabel('ship-State')
         plt.ylabel('count')
         plt.title('Distribution of state')
         plt.xticks(rotation = 45)
         plt.show()
```



Insights

Most of the People Buys M-size

Most of the Quantity Buys M-size in sales

Majority of the Orders are shipped through courier

Most of the People buys T-shirt

Maximum(99.2%) of the buyers are retailers and 0.8% are B2B buyers

Most of the buyers are from Maharashtra State

Conclusion

The data analysis reveals that the buisness has a significant customer base in Maharashtra state, mainly serves retailers, Fulfil orders through Amazon, experinces high demand for T-shirts and sees M-size as the preffered choice among buyers.

```
In [ ]:
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js