In [ ]:

```
# Kuraloviya K [ Amazon_Sales-Analysis ]
```

In [4]:

```

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

In [5]:

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

In [3]:

```
df.shape
```

Out[3]:

```
(128976, 21)
```

In [4]:

```
df.head()
```

Out[4]:

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

5 rows × 21 columns

In [5]:
```
df.tail()
```

Out[5]:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Cour Stat
128971	128970	406-6001380-7673107	05-31-22	Shipped	Amazon	Amazon.in	Expedited	Shirt	XL	Shipp
128972	128971	402-9551604-7544318	05-31-22	Shipped	Amazon	Amazon.in	Expedited	T-shirt	M	Shipp
128973	128972	407-9547469-3152358	05-31-22	Shipped	Amazon	Amazon.in	Expedited	Blazzer	XXL	Shipp
128974	128973	402-6184140-0545956	05-31-22	Shipped	Amazon	Amazon.in	Expedited	T-shirt	XS	Shipp
128975	128974	408-7436540-8728312	05-31-22	Shipped	Amazon	Amazon.in	Expedited	T-shirt	S	Shipp

5 rows × 21 columns

In [6]:
```
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 [16]:

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

In [8]:

```
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 [9]:
```
pd.isnull(df)
# checking null value
```

Out[9]:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	Q
0	False	False	False	False	False	False	False	False	False	False	Fal
1	False	False	False	False	False	False	False	False	False	False	Fal
2	False	False	False	False	False	False	False	False	False	False	Fal
3	False	False	False	False	False	False	False	False	False	False	Fal
4	False	False	False	False	False	False	False	False	False	False	Fal
...	...	...	...	...	...	...	...	...	...	...	
128971	False	False	False	False	False	False	False	False	False	False	Fal
128972	False	False	False	False	False	False	False	False	False	False	Fal
128973	False	False	False	False	False	False	False	False	False	False	Fal
128974	False	False	False	False	False	False	False	False	False	False	Fal
128975	False	False	False	False	False	False	False	False	False	False	Fal

128976 rows × 19 columns

In [10]:
```
pd.isnull(df).sum()
# sum will give total values of null values
```

Out[10]:
```
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 [11]:
```
#drop null values
df.dropna(inplace=True)
```

In [12]:
```
df.shape
```

Out[12]:
```
(37514, 19)
```

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

Out[17]:
```
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 [ ]:
```

```

In [20]:
```
#checking whether the  data type change or not
df['ship-postal-code'].dtype
```

Out[20]:
```
dtype('float64')
```

In [6]:
```
#rename Columns
df.rename(columns={'Qty':'Quantity'})
```

Out[6]:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Cou St
0	0	405-8078784-5731545	04-30-22	Cancelled	Merchant	Amazon.in	Standard	T-shirt	S	Or
1	1	171-9198151-1101146	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	3XL	Ship
2	2	404-0687676-7273146	04-30-22	Shipped	Amazon	Amazon.in	Expedited	Shirt	XL	Ship
3	3	403-9615377-8133951	04-30-22	Cancelled	Merchant	Amazon.in	Standard	Blazzer	L	Or
4	4	407-1069790-7240320	04-30-22	Shipped	Amazon	Amazon.in	Expedited	Trousers	3XL	Ship
...	...	...	...	...	...	...	...	...	...	
128971	128970	406-6001380-7673107	05-31-22	Shipped	Amazon	Amazon.in	Expedited	Shirt	XL	Ship
128972	128971	402-9551604-7544318	05-31-22	Shipped	Amazon	Amazon.in	Expedited	T-shirt	M	Ship
128973	128972	407-9547469-3152358	05-31-22	Shipped	Amazon	Amazon.in	Expedited	Blazzer	XXL	Ship
128974	128973	402-6184140-0545956	05-31-22	Shipped	Amazon	Amazon.in	Expedited	T-shirt	XS	Ship
128975	128974	408-7436540-8728312	05-31-22	Shipped	Amazon	Amazon.in	Expedited	T-shirt	S	Ship

128976 rows × 21 columns

In [25]:
```
df.describe(include='object')
```

Out[25]:

	Order ID	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	currency
count	128976	128976	128976	128976	128976	128976	128976	128976	121176
unique	120229	13	2	2	2	9	11	4	1
top	403-4984515-8861958	Shipped	Amazon	Amazon.in	Expedited	T-shirt	M	Shipped	INR
freq	12	77815	89713	128852	88630	50292	22373	109486	121176

In [26]:
```
df[['Qty','Amount']].describe()
```

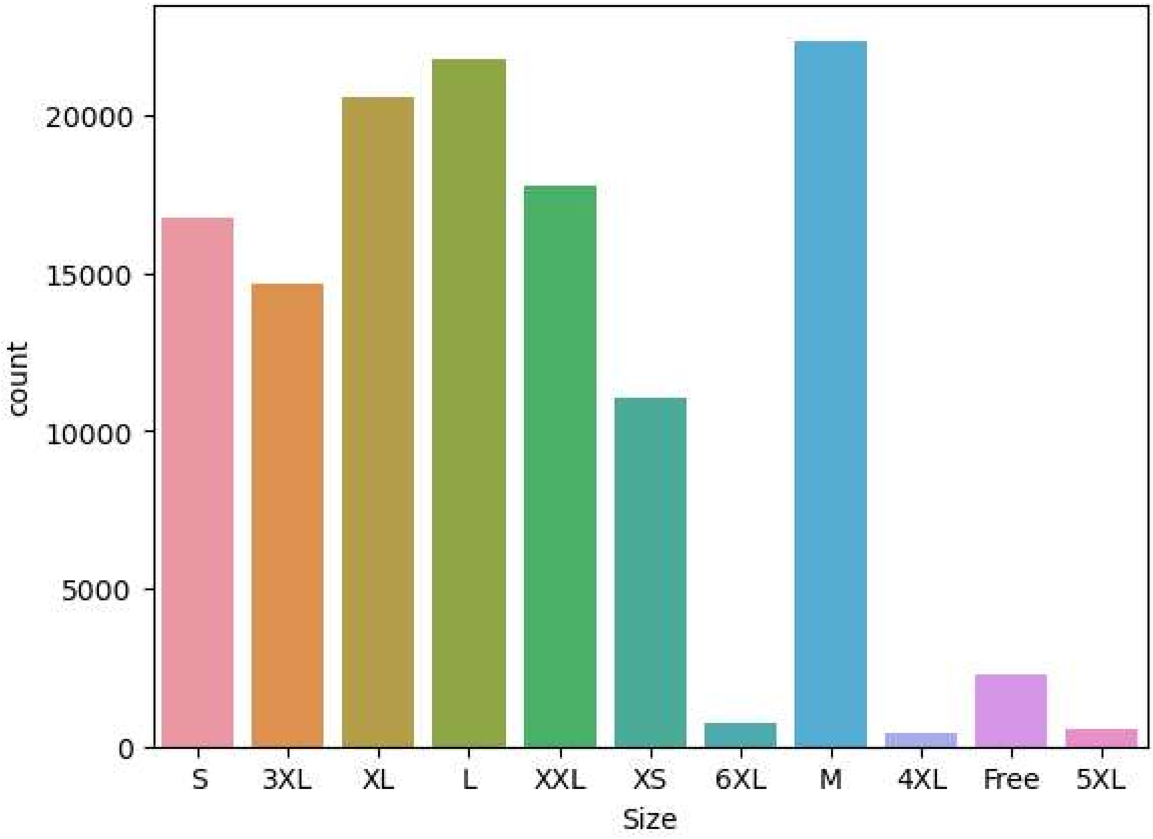
Out[26]:

	Qty	Amount
count	128976.000000	121176.000000
mean	0.904401	648.562176
std	0.313368	281.185041
min	0.000000	0.000000
25%	1.000000	449.000000
50%	1.000000	605.000000
75%	1.000000	788.000000
max	15.000000	5584.000000

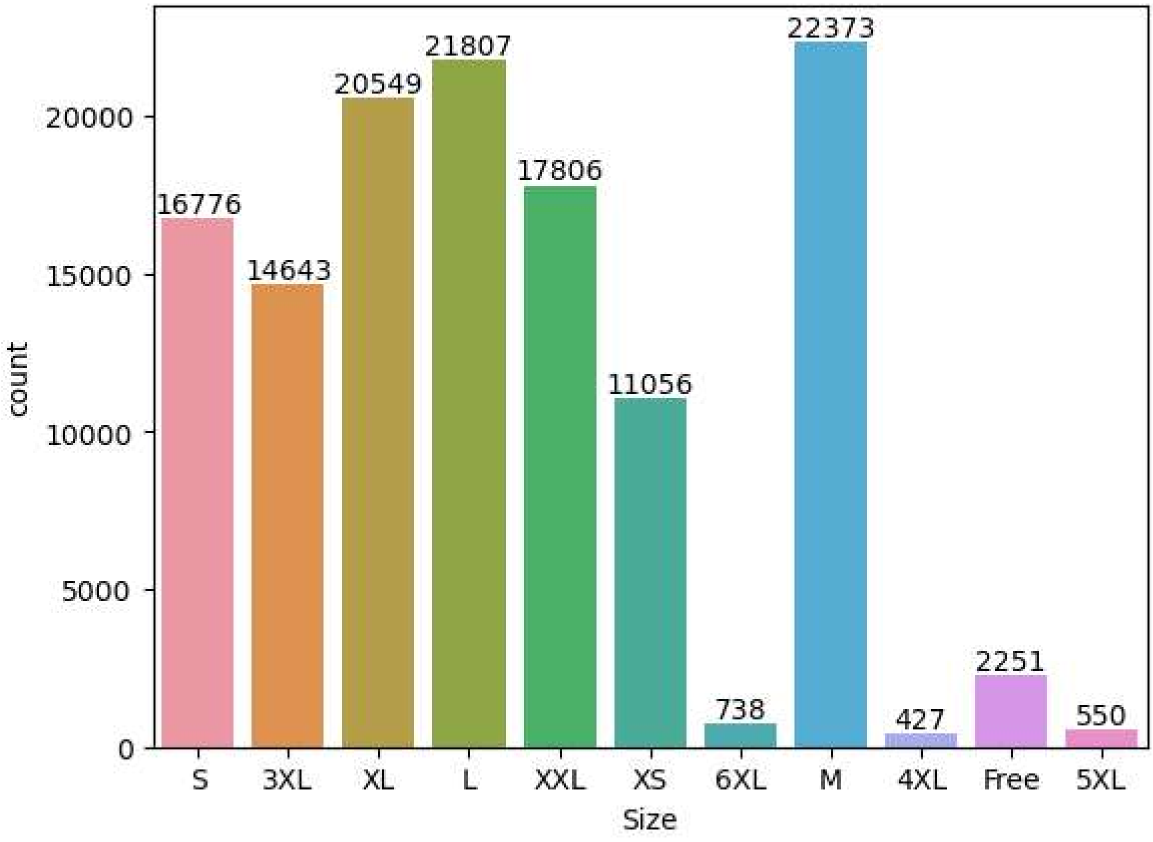
In [27]:
```
df.columns
```

Out[27]:
```
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 [28]: ax=sns.countplot(x='Size' ,data=df)
```



In [29]:
```
ax=sns.countplot(x='Size' ,data=df)

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



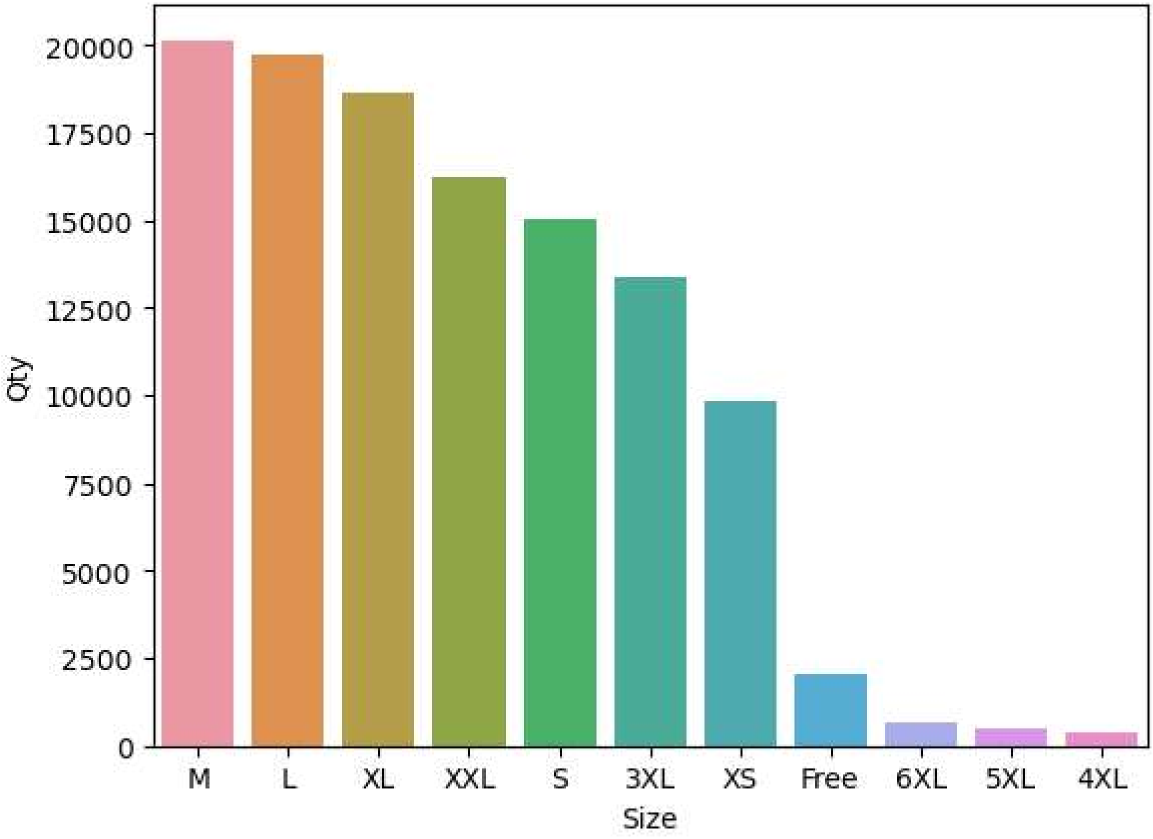
In [31]:
```
df.groupby(['Size'], as_index=False)['Qty'].sum().sort_values(by='Qty',ascendi
```

Out[31]:

	Size	Qty
6	M	20138
5	L	19706
8	XL	18636
10	XXL	16246
7	S	15041
0	3XL	13360
9	XS	9850
4	Free	2070
3	6XL	688
2	5XL	513
1	4XL	398

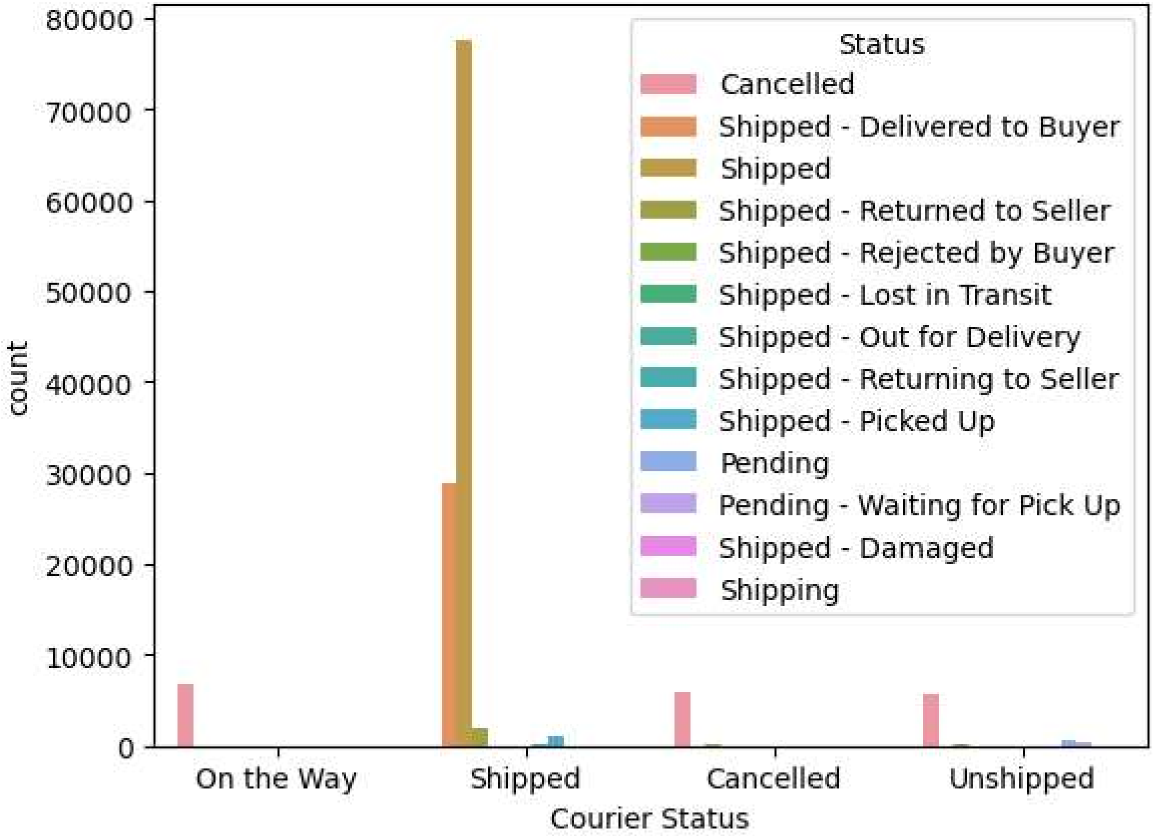
In [33]:
```
Size_Qty=df.groupby(['Size'], as_index=False)['Qty'].sum().sort_values(by='Qty

sns.barplot(x='Size',y='Qty', data=Size_Qty)
```

Out[33]:
```
<Axes: xlabel='Size', ylabel='Qty'>
```

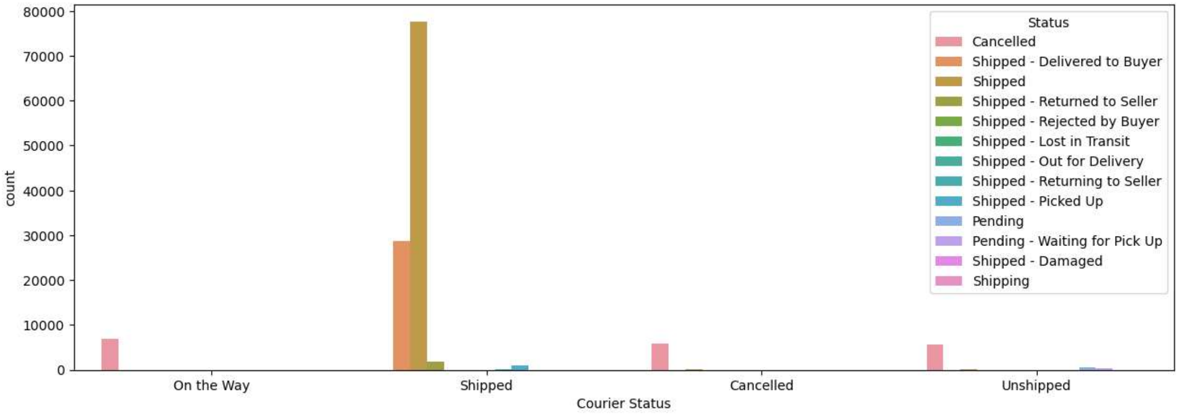


In [34]:
```
sns.countplot(data=df, x='Courier Status',hue= 'Status')
```

Out[34]:
```
<Axes: xlabel='Courier Status', ylabel='count'>
```



In [37]:
```
plt.figure(figsize=(15,5))

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

plt.show()
```

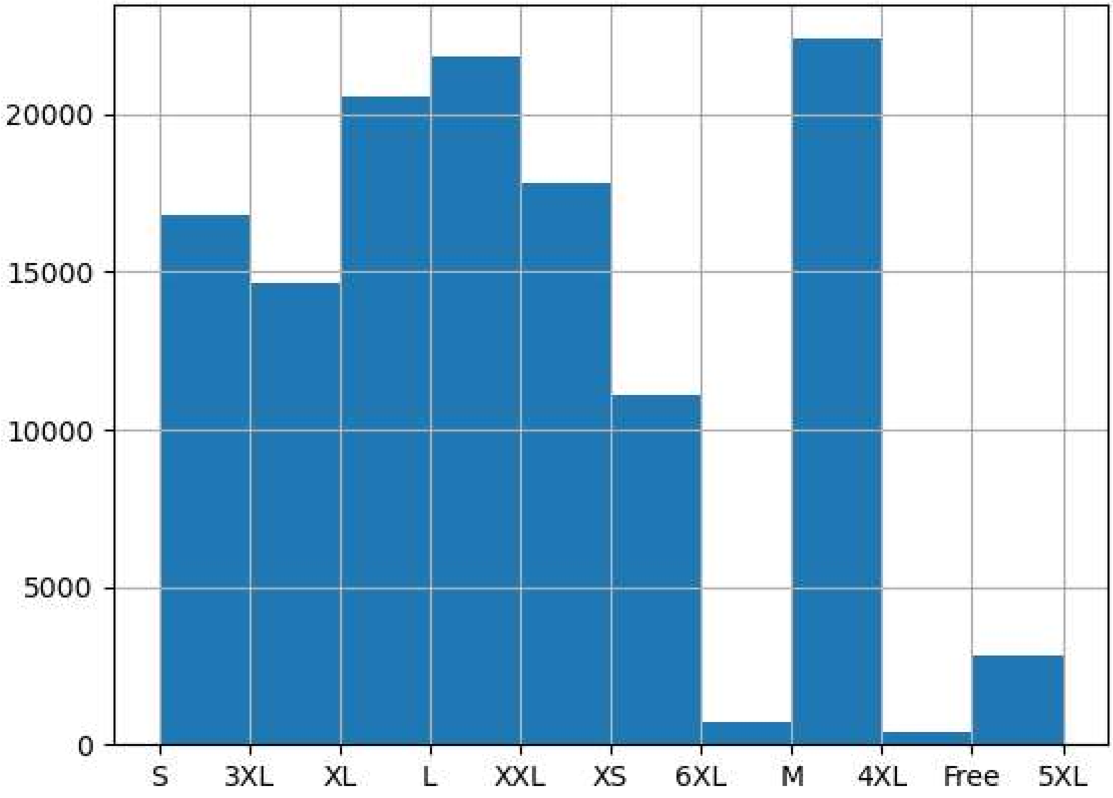


In [7]:

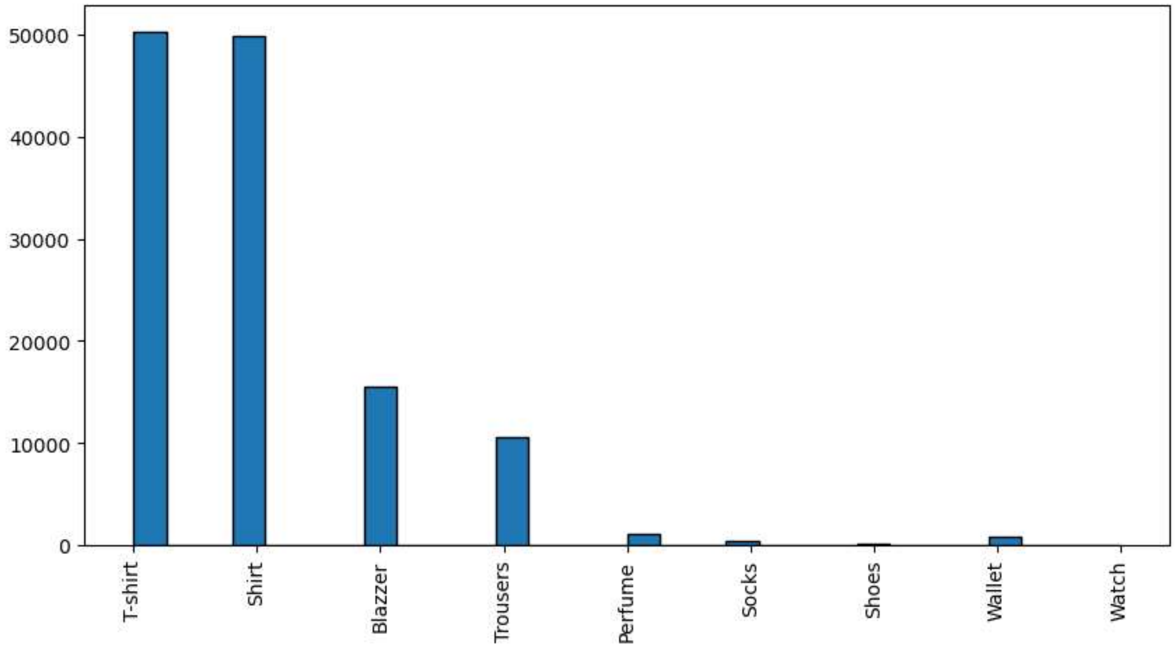
```
#histogram
df['Size'].hist()
```

Out[7]:

```
<Axes: >
```

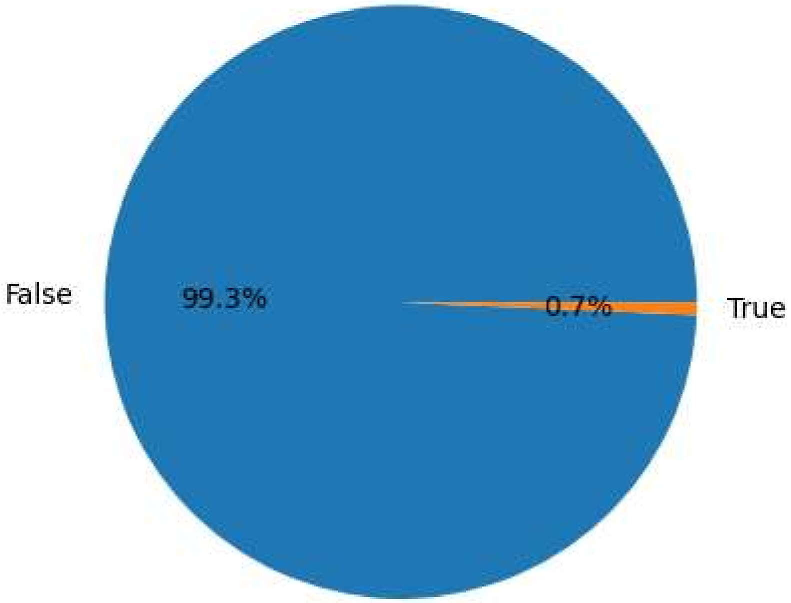


In [9]:
```
df['Category'] = df['Category'].astype(str)
column_data = df['Category']
plt.figure(figsize=(10, 5))
plt.hist(column_data, bins=30, edgecolor='Black')
plt.xticks(rotation=90)
plt.show()
```



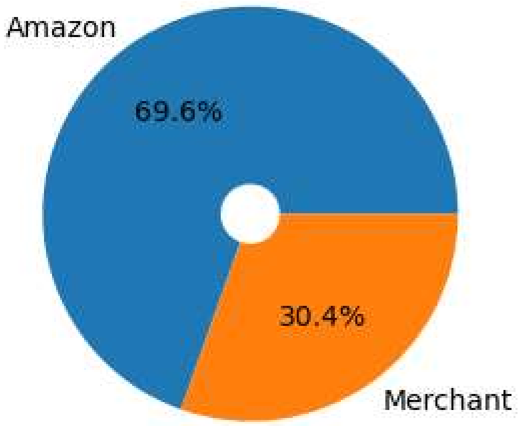
In [42]:

```
# Checking B2B Data  by using pie chart
B2B_Check = df['B2B'].value_counts()

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



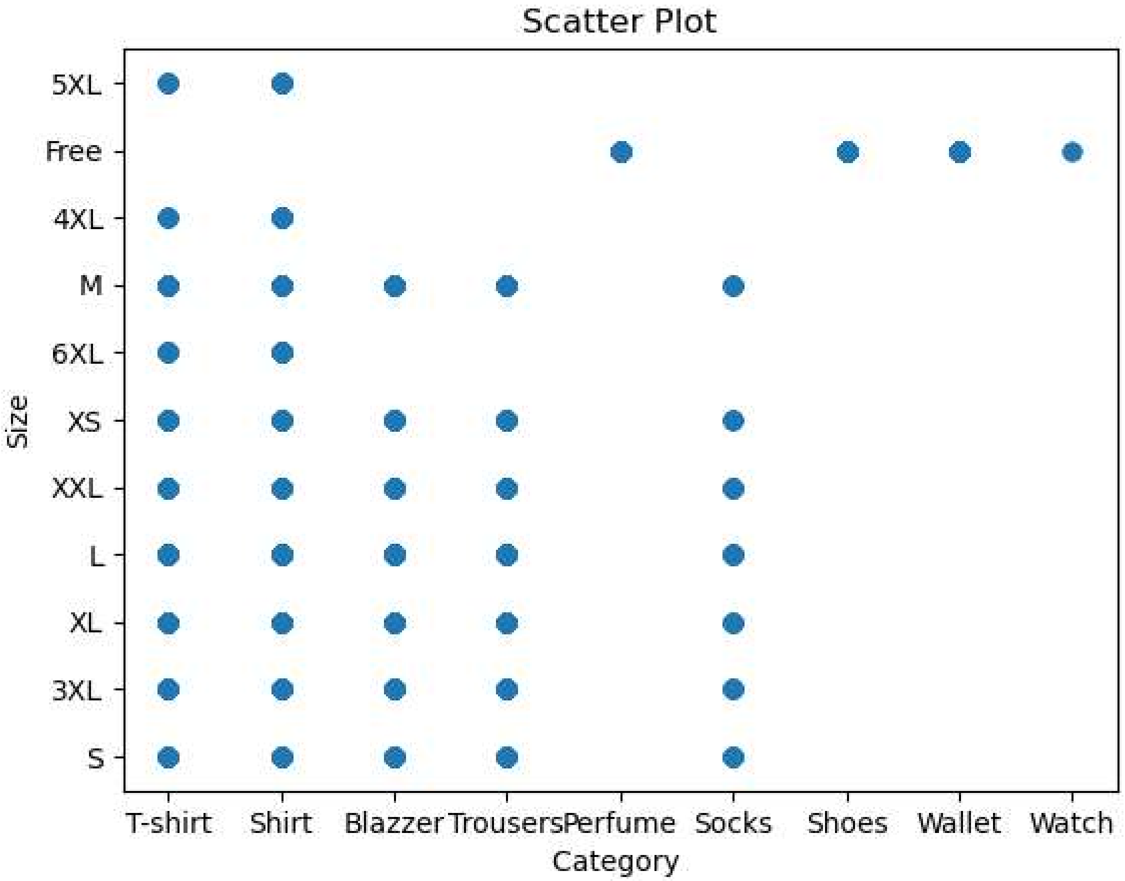
In [43]:

```
#  Prepare data for pie chart
a1 = df['Fulfilment'].value_counts()

# Step 4: Plot the pie chart
fig, ax = plt.subplots()

ax.pie(a1, labels=a1.index, autopct='%1.1f%%', radius=0.7, wedgeprops=dict(wid
ax.set(aspect="equal")

plt.show()
```

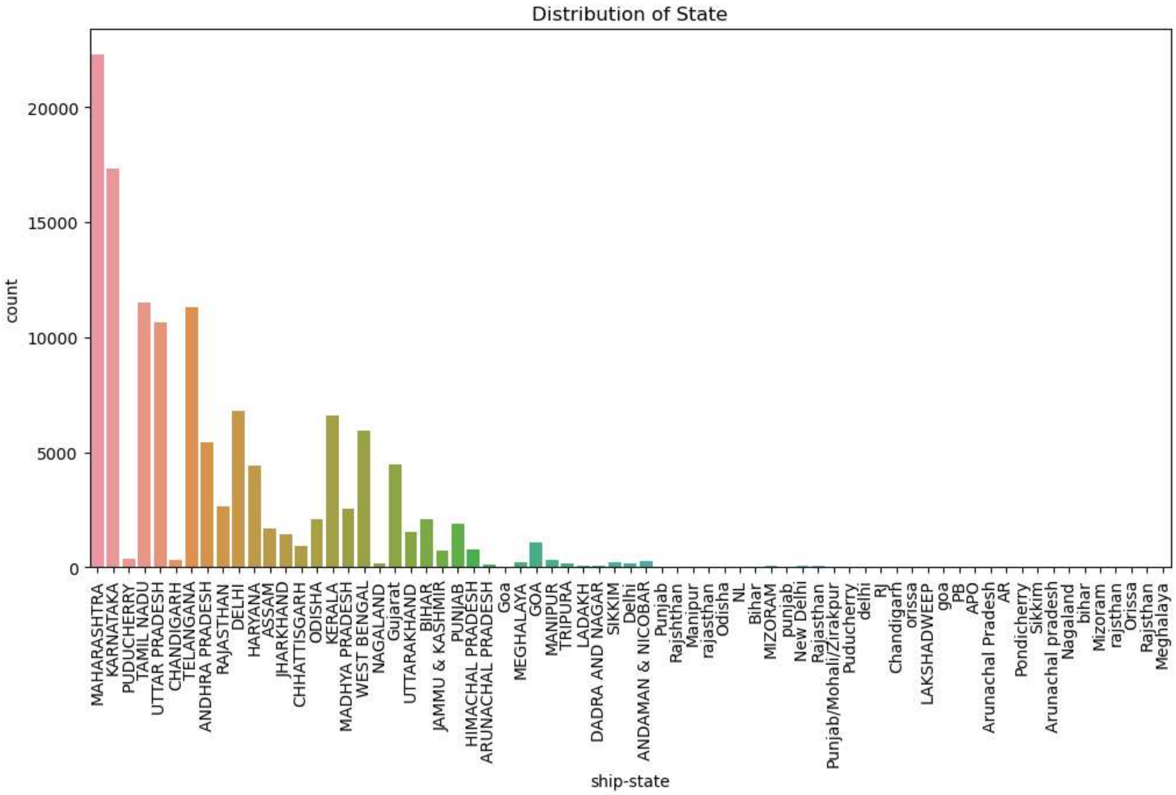


In [46]:

```
x_data = df['Category']
y_data = df['Size']

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

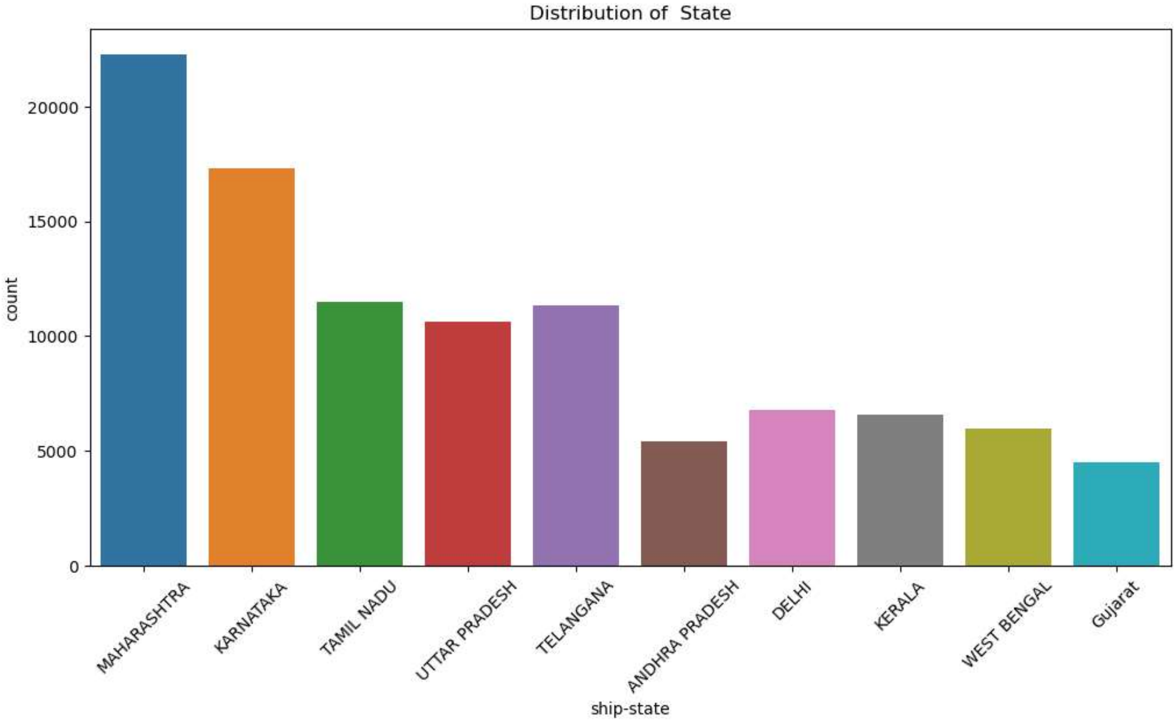


In [47]:
```
plt.figure(figsize=(12, 6))
sns.countplot(data=df, x='ship-state')
plt.xlabel('ship-state')
plt.ylabel('count')
plt.title('Distribution of State')
plt.xticks(rotation=90)
plt.show()
```



In [48]:

```
# top_10_States
top_10_state = df['ship-state'].value_counts().head(10)
# Plot count of cities by state
plt.figure(figsize=(12, 6))
sns.countplot(data=df[df['ship-state'].isin(top_10_state.index)], x='ship-stat
plt.xlabel('ship-state')
plt.ylabel('count')
plt.title('Distribution of  State')
plt.xticks(rotation=45)
plt.show()
```



In [ ]: