```
In [2]:  import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         %matplotlib inline
         import seaborn as sns
```

```
In [3]:  df=pd.read_csv('Amazon Sale Report.csv',encoding= 'unicode_escape')
```

```
In [4]:  df.shape
```

Out[4]: (128976, 21)

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

Out[5]:

	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 [8]:  df.tail()
```

Out[8]:

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

In [12]: ```python
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 [13]: ```python
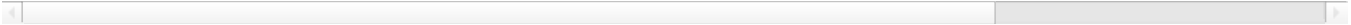pd.isnull(df)
checking null value
```

| | index | Order ID | Date | Status | Fulfilment | Sales Channel | ship-service-level | Category | Size | Courier Status | Qty | currency | Amount | ship-city | ship-state |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False |
| **1** | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False |
| **2** | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False |
| **3** | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False |
| **4** | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **128971** | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False |
| **128972** | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False |
| **128973** | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False |
| **128974** | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False |
| **128975** | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False |

128976 rows × 19 columns

```python
pd.isnull(df).sum()
# sum will give total values of null values
```

```
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 [16]:
```python
df.shape
```

Out[16]: `(128976, 19)`

In [17]:
```python
#drop null values
df.dropna(inplace=True)
```

In [18]:
```python
df.shape
```

Out[18]: `(37514, 19)`

In [19]:
```python
df.columns
```

Out[19]:
```
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 [20]:
```python
# change data type
df['ship-postal-code']=df['ship-postal-code'].astype('int')
```

In [21]:
```python
#checking whether the  data type change or not
df['ship-postal-code'].dtype
```

Out[21]: `dtype('int32')`

In [25]:
```python
df['Date']=pd.to_datetime (df['Date'])
```

In [27]:
```python
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]: #rename Columns
         df.rename(columns={'Qty':'Quantity'})
```

Out[28]:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	Quantity	currency	Amount
0	0	405-8078784-5731545	2022-04-30	Cancelled	Merchant	Amazon.in	Standard	T-shirt	S	On the Way	0	INR	647.62
1	1	171-9198151-1101146	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	3XL	Shipped	1	INR	406.00
3	3	403-9615377-8133951	2022-04-30	Cancelled	Merchant	Amazon.in	Standard	Blazzer	L	On the Way	0	INR	753.33
7	7	406-7807733-3785945	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	S	Shipped	1	INR	399.00
12	12	405-5513694-8146768	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	XS	Shipped	1	INR	399.00
...	...	...	...	...	...	...	...	...	...	...	...	...	...
128875	128874	405-4724097-1016369	2022-06-01	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	T-shirt	S	Shipped	1	INR	854.00
128876	128875	403-9524128-9243508	2022-06-01	Cancelled	Merchant	Amazon.in	Standard	Blazzer	XL	On the Way	0	INR	734.29
128888	128887	405-6493630-8542756	2022-05-31	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Trousers	M	Shipped	1	INR	518.00
128891	128890	407-0116398-1810752	2022-05-31	Cancelled	Merchant	Amazon.in	Standard	Wallet	Free	On the Way	0	INR	398.10
128892	128891	403-0317423-9322704	2022-05-31	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Blazzer	M	Shipped	1	INR	721.00

37514 rows × 19 columns
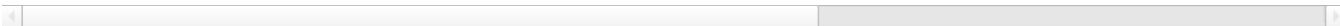
```
In [29]: #describe() method return description of the data in the DataFrame(i.e count,mean,std,min..etc)
         df.describe()
```

Out[29]:

	index	Date	Qty	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 [30]: df.describe(include='object')
```

	Order ID	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	currency	ship-city	ship-state	cc
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	6892	31859	37514	2839	6236	

```
In [31]: #use describe() for specific columns
         df[['Qty','Amount']].describe()
```

Out[31]:

	Qty	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 [32]: df.columns
```

```
Out[32]: 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')
```

size

```
In [33]: ax=sns.countplot(x='Size' ,data=df)
```
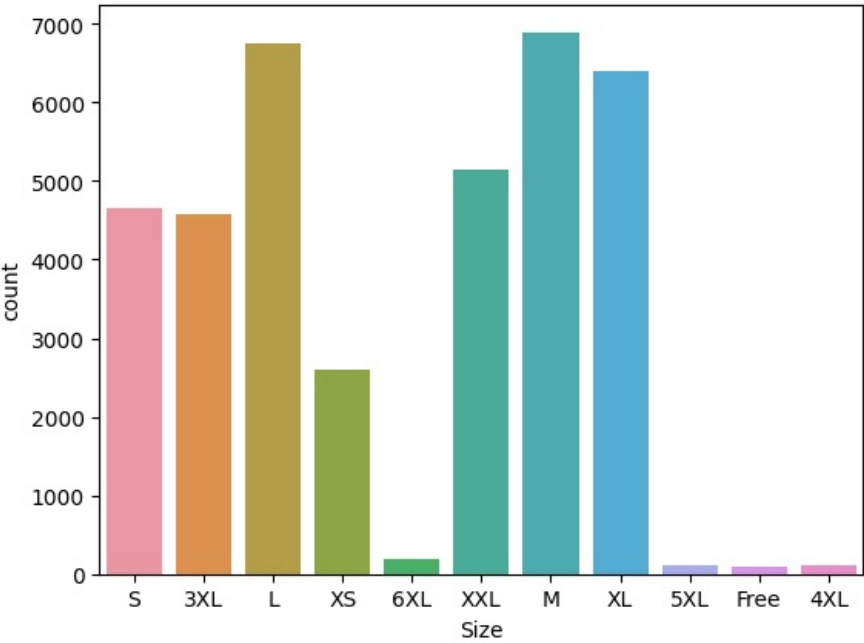


```
In [35]: ax=sns.countplot(x='Size' ,data=df)

         for bars in ax.containers:
             ax.bar_label(bars)
```
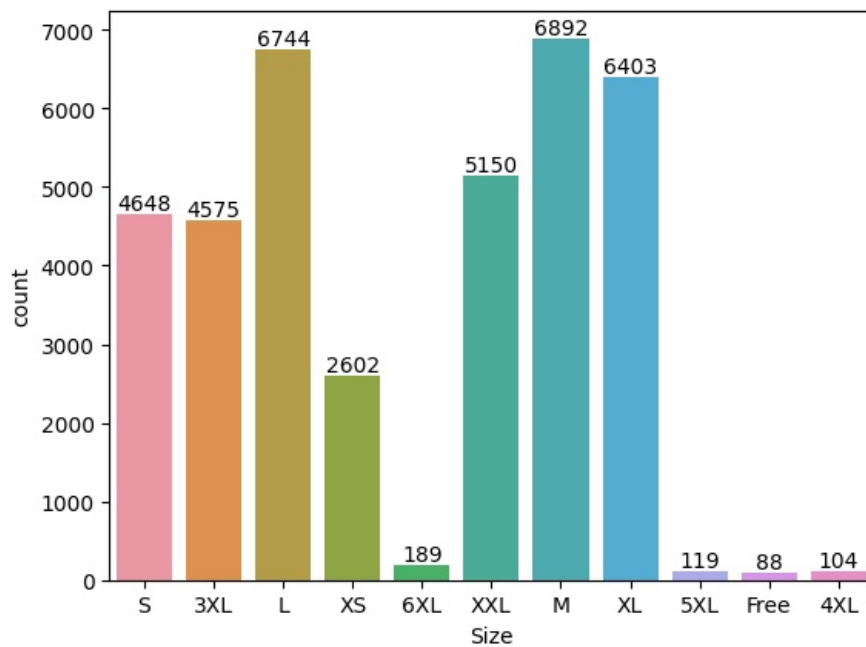
Note: From above Graph you can see that most of the people buys M-Size

Group By

The groupby() function in pandas is used to group data based on one or more columns in a DataFrame

```
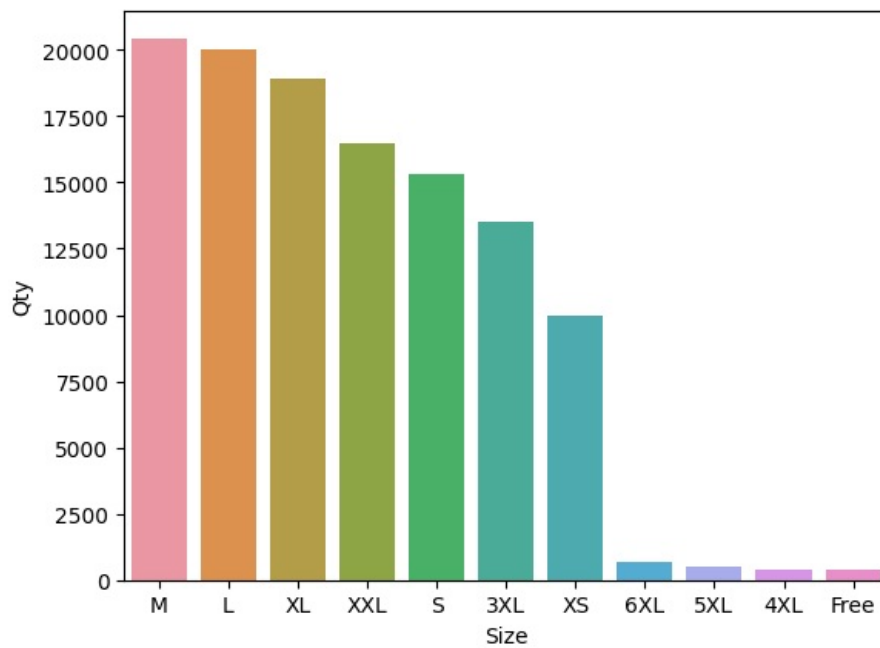In [38]: df.groupby(['Size'], as_index=False)['Qty'].sum().sort_values(by='Qty',ascending=False)
```

Out[38]:

	Size	Qty
6	M	5978
5	L	5875
8	XL	5543
10	XXL	4518
0	3XL	4008
7	S	3973
9	XS	2204
3	6XL	170
2	5XL	104
1	4XL	93
4	Free	73

```
In [95]: S_Qty=df.groupby(['Size'], as_index=False)['Qty'].sum().sort_values(by='Qty',ascending=False)

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

Out[95]: <Axes: xlabel='Size', ylabel='Qty'>

Note: From above Graph you can see that most of the Qty buys M-Size in the sales

Courier Status

```
In [101... sns.countplot(data=df, x='Courier Status',hue= 'Status')
```

```
Out[101... <Axes: xlabel='Courier Status', ylabel='count'>
```
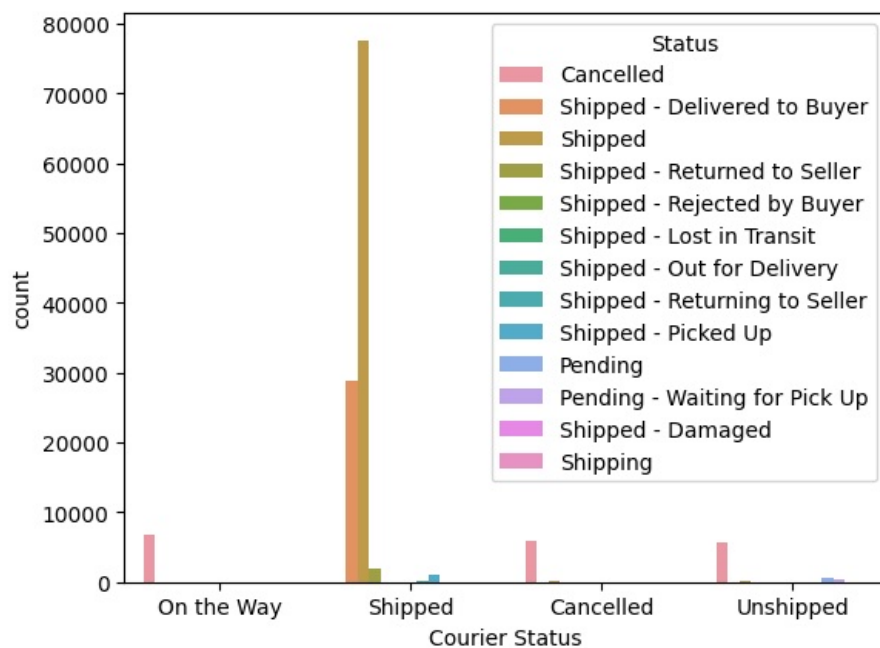


```
In [59]: plt.figure(figsize=(10,5))
```

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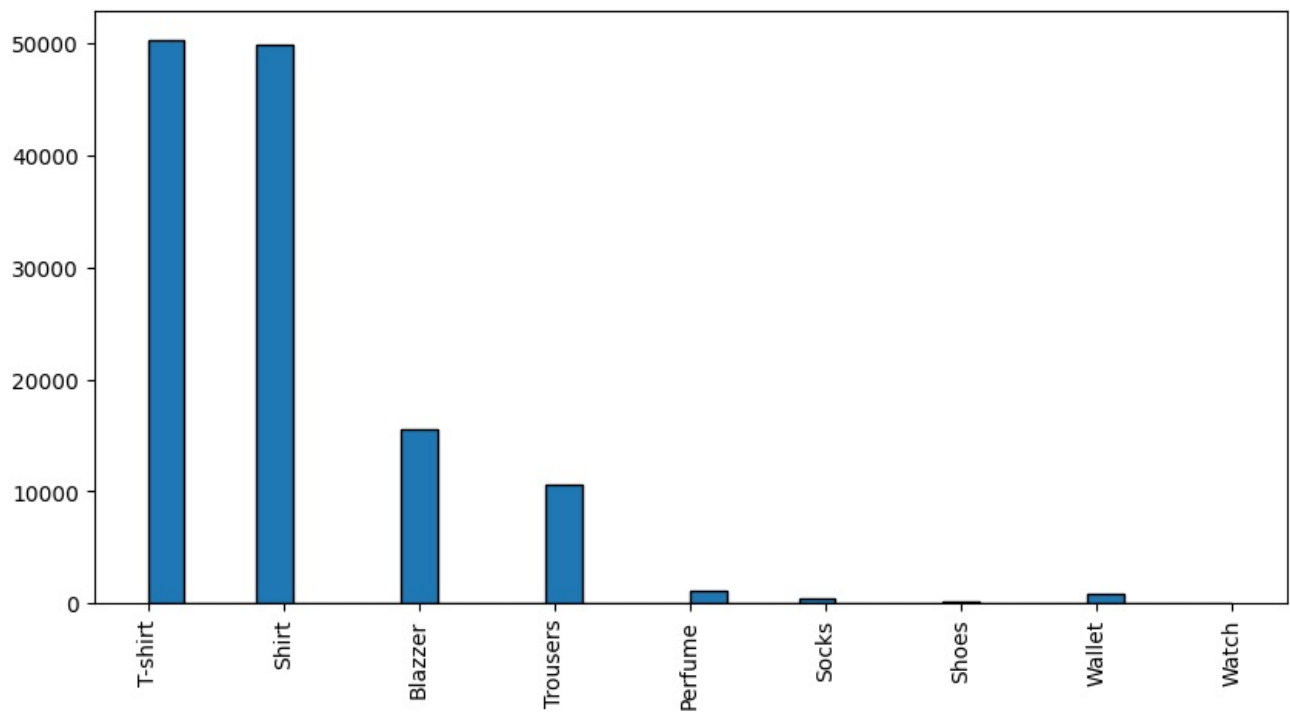ax=sns.countplot(data=df, x='Courier Status',hue= 'Status')

plt.show()
```
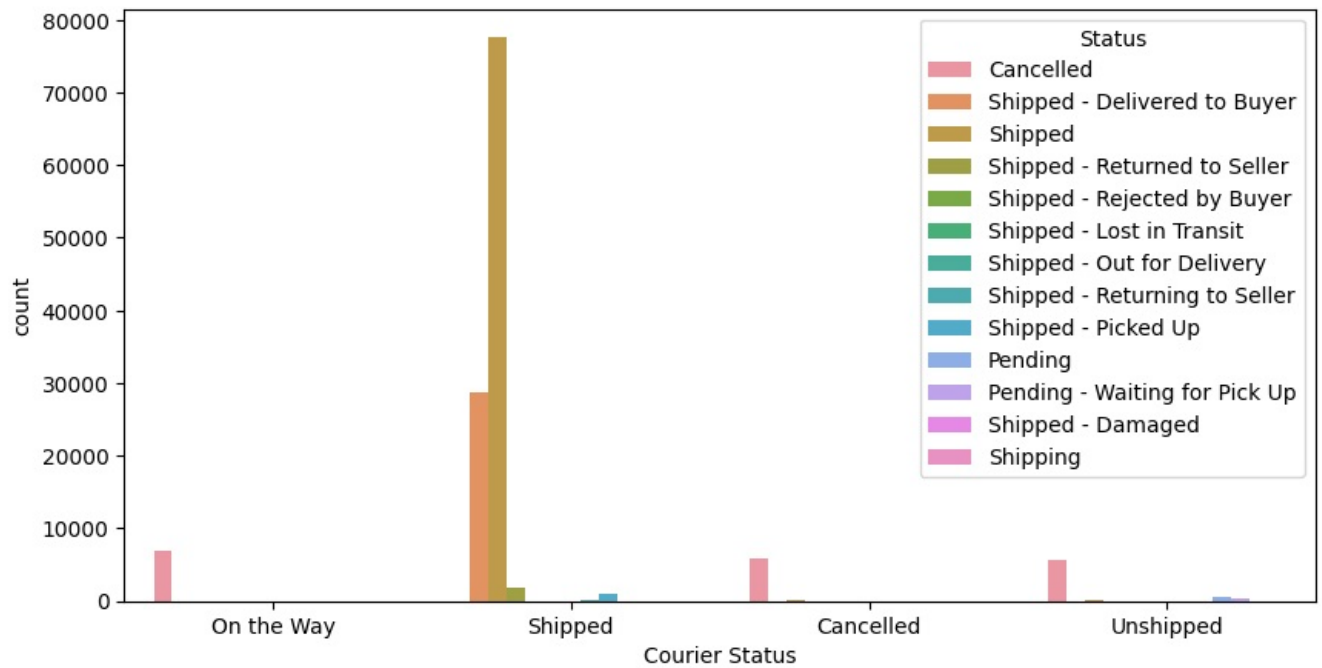

Note: From above Graph the majority of the orders are shipped through the courier.

In [81]:
```
#histogram
df['Size'].hist()
```

Out[81]: <Axes: >
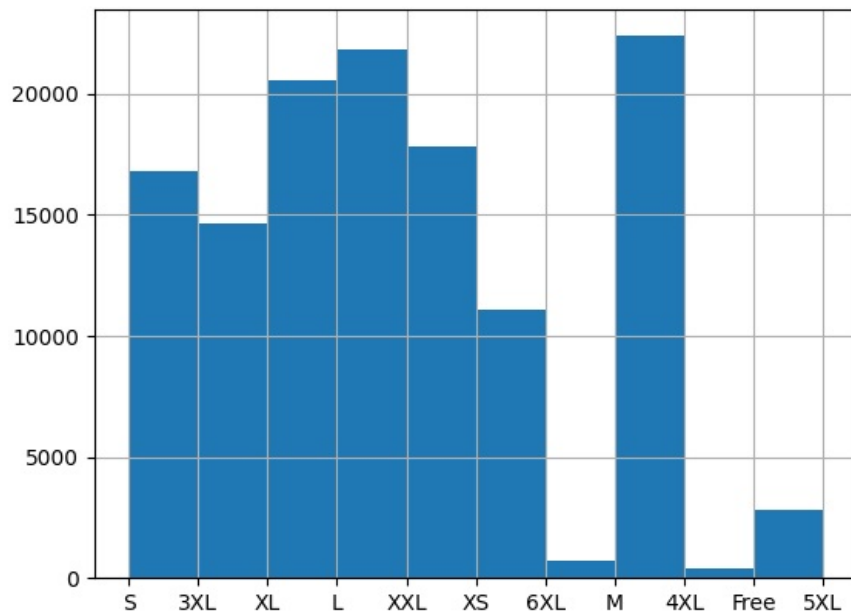
In [82]:
```
df['Category'] = df['Category'].astype(str)
column_data = df['Category']
plt.figure(figsize=(10, 5))
plt.hist(column_data, bins=30, edgecolor='Black')
plt.xticks(rotation=90)
plt.show()
```

Note: From above Graph you can see that most of the buyers are T-shirt

```
In [112... # Checking B2B Data  by using pie chart
         B2B_Check = df['B2B'].value_counts()

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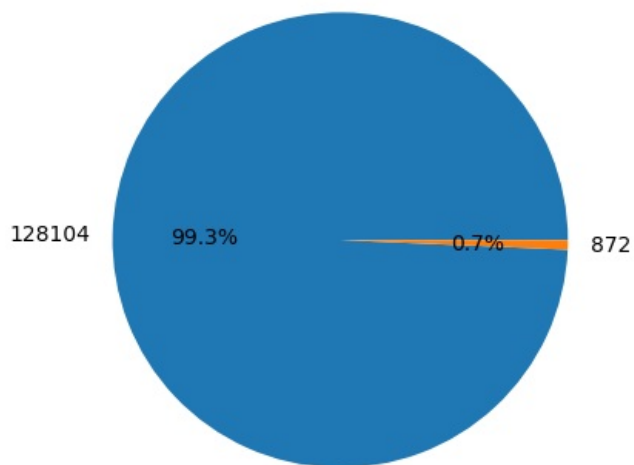


```
In [113... # Checking B2B Data  by using pie chart
         B2B_Check = df['B2B'].value_counts()

         #  Plot the pie chart
         plt.pie(B2B_Check, labels=B2B_Check.index, autopct='%1.1f%%')
         #plt.axis('equal')
         plt.show()
```
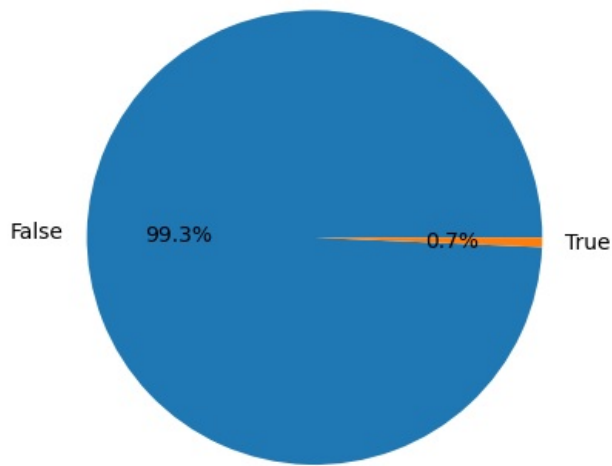
Note : From above chart we can see that maximum i.e. 99.3% of buyers are retailers and 0.7% are B2B buyers

```python
#  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(width=0.6))
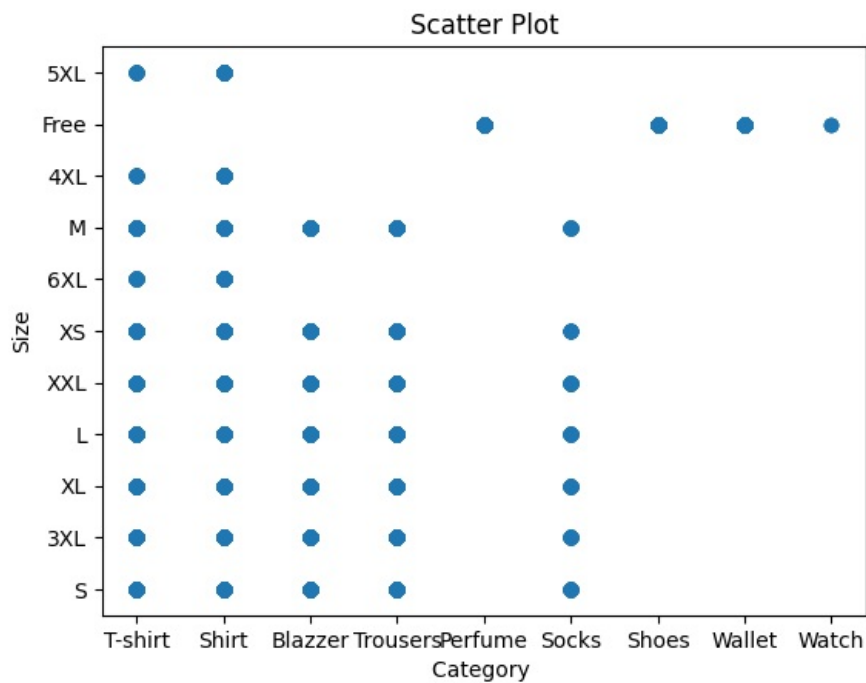ax.set(aspect="equal")

plt.show()
```
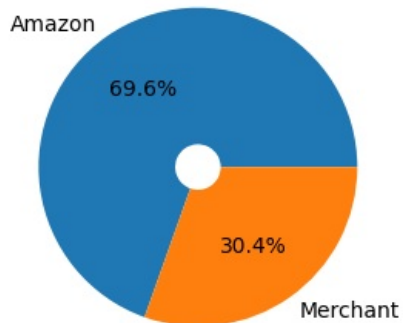

Note: From above chart you can see that most of the Fulfilment are amazon

```python
# Prepare data for scatter plot
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()
```

Scatter Plot

```python
# Plot count of cities by state
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()
```

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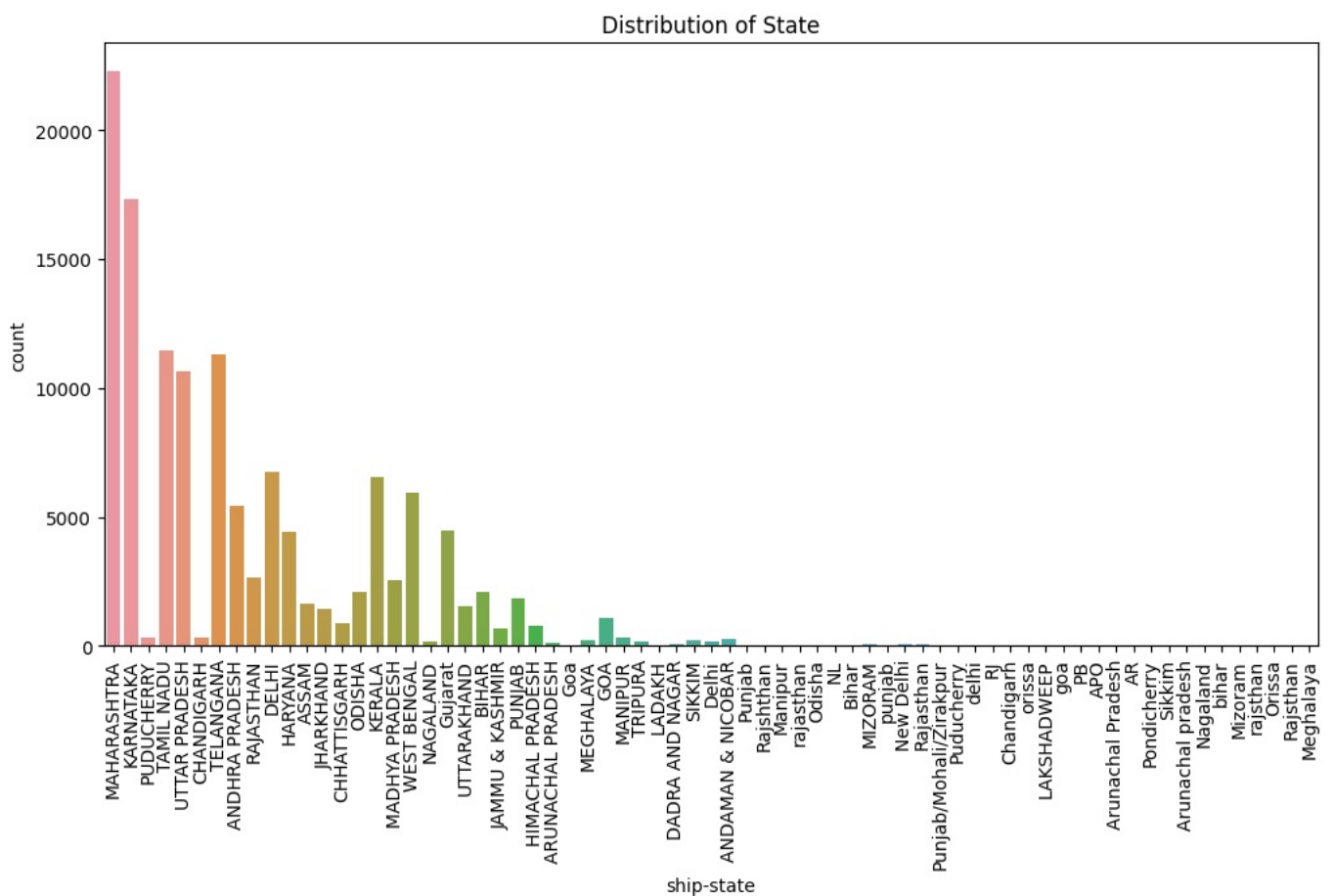
```python
# top_10_States
top_10_state = df['ship-state'].value_counts().head(10)
# Plot count of cities by state
plt.figure(figsize=(12, 6))
sns.countplot(data=df[df['ship-state'].isin(top_10_state.index)], x='ship-state')
plt.xlabel('ship-state')
plt.ylabel('count')
plt.title('Distribution of  State')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

Distribution of State
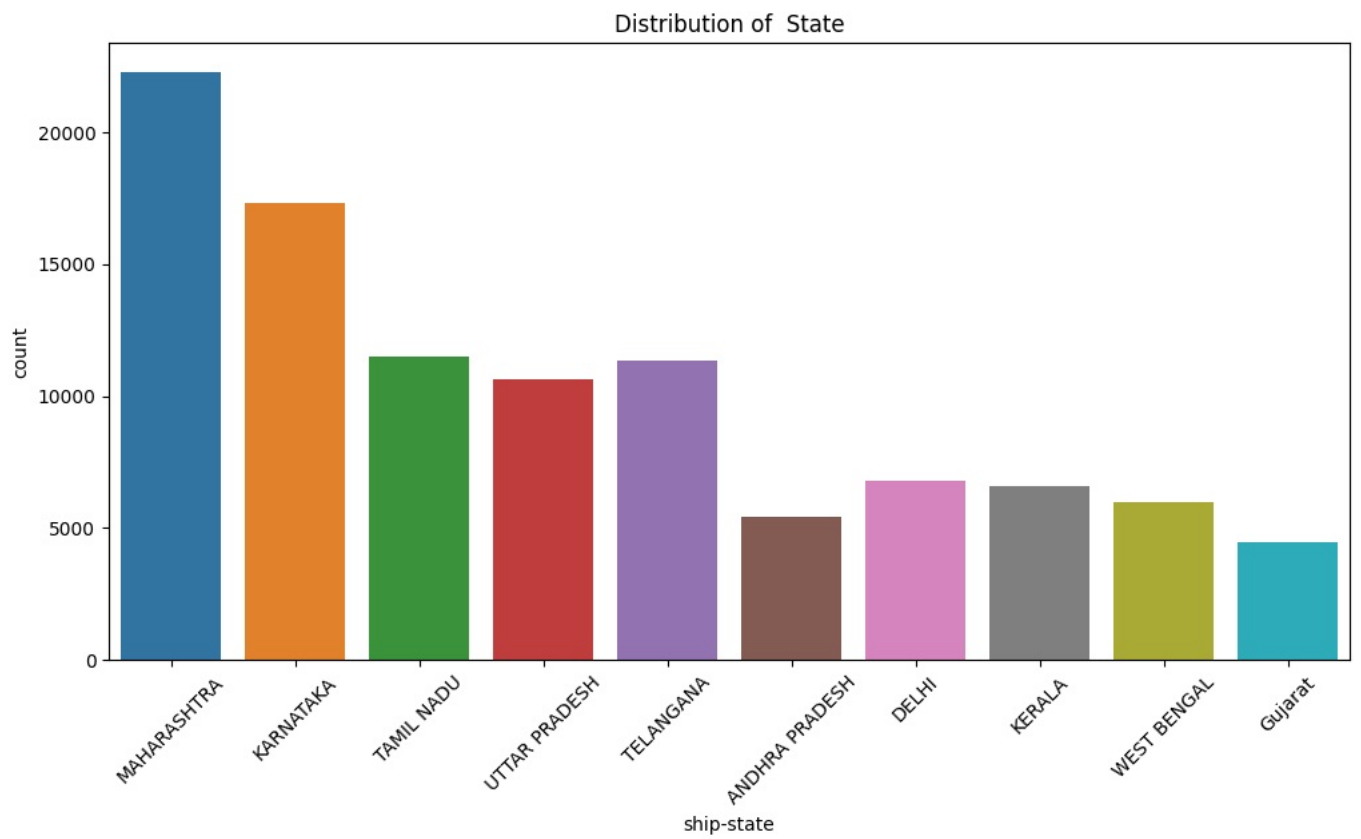
Note: From above Graph you can see that most of the buyers are Maharashtra state

Conclusion

The data analysis reveals that the business has a significant customer base in Maharashtra state, mainly serves retailers, fulfills orders through Amazon, experiences high demand for T-shirts, and sees M-Size as the preferred choice among buyers.

In [ ]:

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