

# Amazon Sales Analysis

```
In [6]: # Import Libraries  
# Library used to perform data manipulation and cleaning  
import pandas as pd  
# Library used for Working with arrays  
import numpy as np  
# Library used for basic graphs  
import matplotlib.pyplot as plt  
# Library used for patterns and plots for data visualization.  
import seaborn as sns
```

```
In [7]: df=pd.read_csv('Amazon Sale Report.csv')
```

```
In [8]: df.shape
```

```
Out[8]: (128976, 21)
```

```
In [9]: df.head(5)
```

```
Out[9]:
```

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

5 rows × 21 columns

```
In [11]: df.tail(5)
```

Out[11]:

		index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Couri 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 [12]: `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 [13]: `#drop unrelated/black columns  
df.drop(['New', 'PendingS'], axis=1, inplace=True)`In [15]: `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 [17]: `# Checking null value  
pd.isnull(df)`

Out[17]:

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

128976 rows × 19 columns

In [18]: `pd.isnull(df).sum()`

```
Out[18]: 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 [19]: df.shape
```

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

```
In [21]: #drop null values  
df.dropna(inplace=True)
```

```
In [23]: df.shape
```

```
Out[23]: (37514, 19)
```

```
In [24]: df.columns
```

```
Out[24]: 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 [25]: # Change data type  
df['ship-postal-code']=df['ship-postal-code'].astype('int')
```

```
In [27]: # Checking whether the data type change or not  
df['ship-postal-code'].dtype
```

```
Out[27]: dtype('int32')
```

```
In [28]: # Change date data type  
df['Date']=pd.to_datetime(df['Date'])
```

```
In [30]: # Renames Columns  
df.rename(columns={'Qty' : 'Quantity'})
```

Out[30]:

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

37514 rows × 19 columns

In [31]: # Count, Mean, Std, Min, Max only for numerical.  
df.describe()

Out[31]:

	index	Qty	Amount	ship-postal-code
<b>count</b>	37514.000000	37514.000000	37514.000000	37514.000000
<b>mean</b>	60953.809858	0.867383	646.553960	463291.552754
<b>std</b>	36844.853039	0.354160	279.952414	194550.425637
<b>min</b>	0.000000	0.000000	0.000000	110001.000000
<b>25%</b>	27235.250000	1.000000	458.000000	370465.000000
<b>50%</b>	63470.500000	1.000000	629.000000	500019.000000
<b>75%</b>	91790.750000	1.000000	771.000000	600042.000000
<b>max</b>	128891.000000	5.000000	5495.000000	989898.000000

In [32]:

`df.describe(include='object')`

Out[32]:

	Order ID	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	currency
<b>count</b>	37514	37514	37514	37514	37514	37514	37514	37514	37514
<b>unique</b>	34664	11	1	1	1	8	11	3	1
<b>top</b>	5057375-2831560	171-Shipped Delivered to Buyer	Merchant	Amazon.in	Standard	T-shirt	M	Shipped	INR
<b>freq</b>	12	28741	37514	37514	37514	14062	6806	31859	37514

In [33]:

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

Out[33]:

	Qty	Amount
<b>count</b>	37514.000000	37514.000000
<b>mean</b>	0.867383	646.553960
<b>std</b>	0.354160	279.952414
<b>min</b>	0.000000	0.000000
<b>25%</b>	1.000000	458.000000
<b>50%</b>	1.000000	629.000000
<b>75%</b>	1.000000	771.000000
<b>max</b>	5.000000	5495.000000

## Exploratory Data Analysis

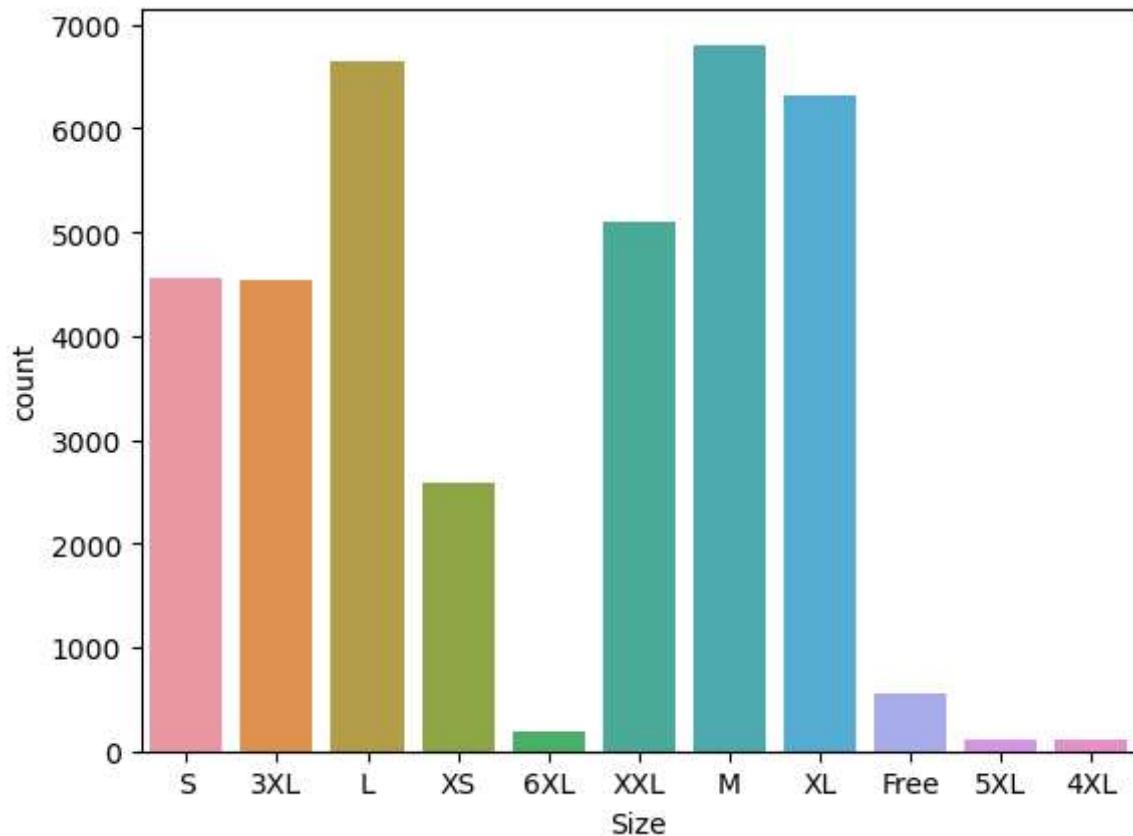
In [35]:

`df.columns`

```
Out[35]: 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')
```

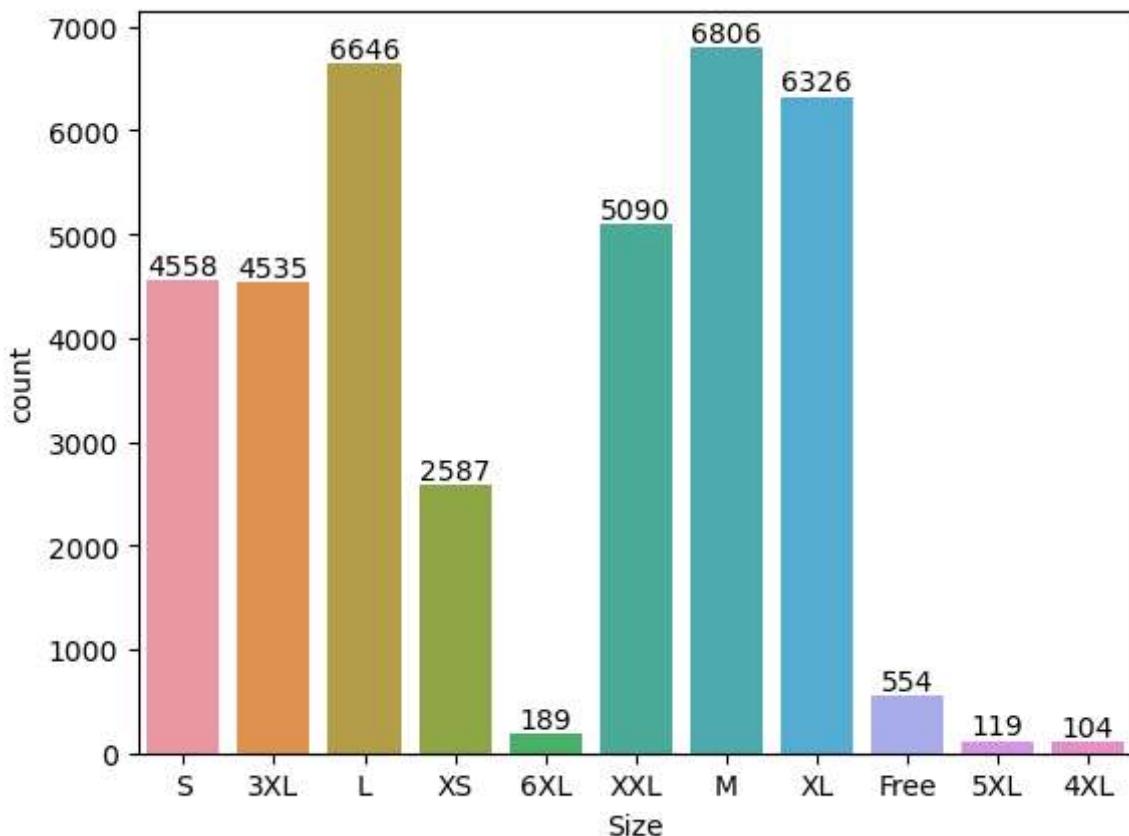
## Size

```
In [37]: ax=sns.countplot(x='Size' ,data=df)
```



```
In [38]: 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

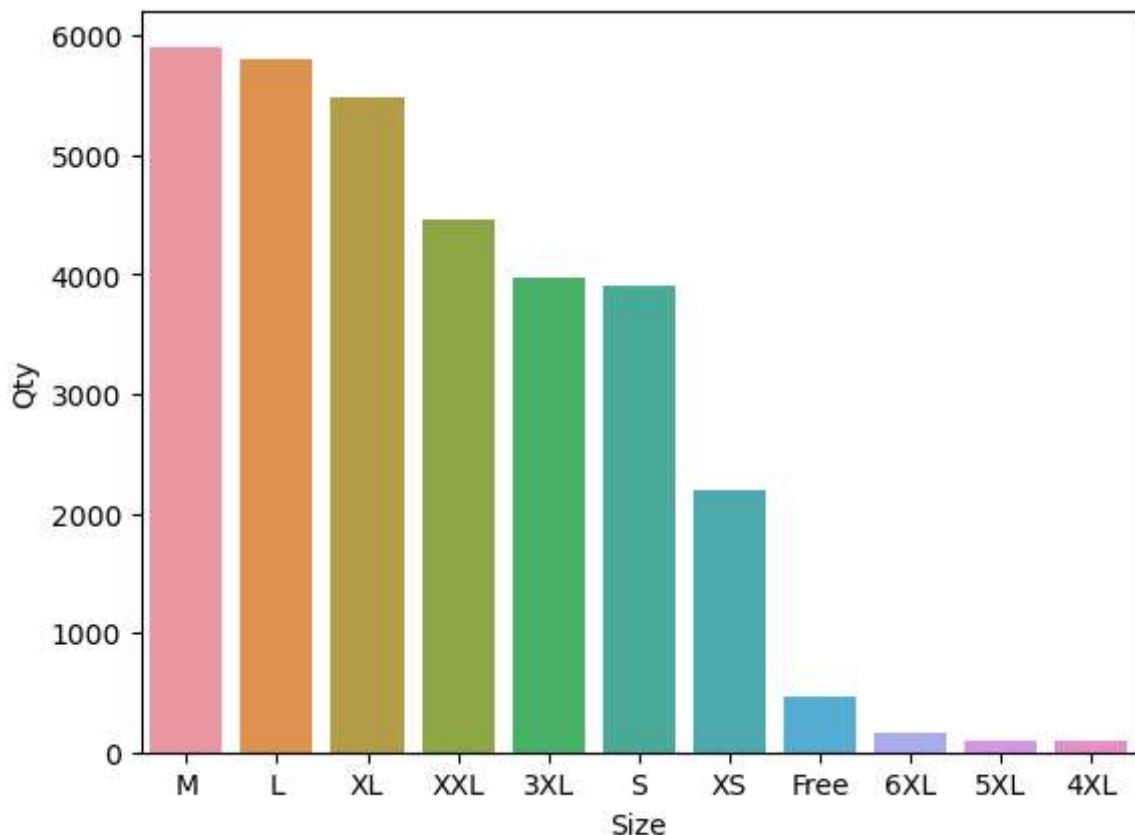
```
In [39]: df.groupby(['Size'], as_index=False)['Qty'].sum().sort_values(by='Qty', ascending=False)
```

```
Out[39]:
```

	Size	Qty
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 [41]: 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[41]: &lt;AxesSubplot:xlabel='Size', ylabel='Qty'&gt;

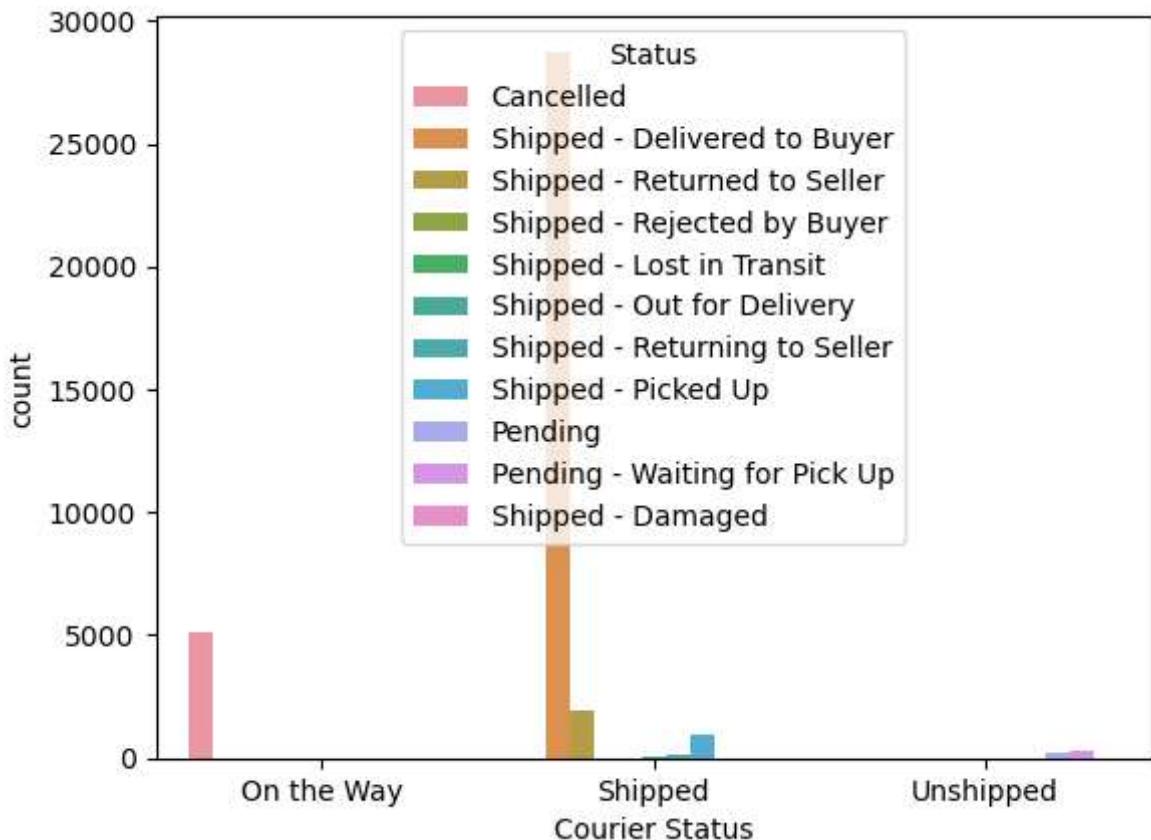


Note: From above Graph you can see that most of the Qty buys M-Size in the sales.

## Courier Status

In [51]: `sns.countplot(data=df, x='Courier Status', hue= 'Status')`

Out[51]: &lt;AxesSubplot:xlabel='Courier Status', ylabel='count'&gt;



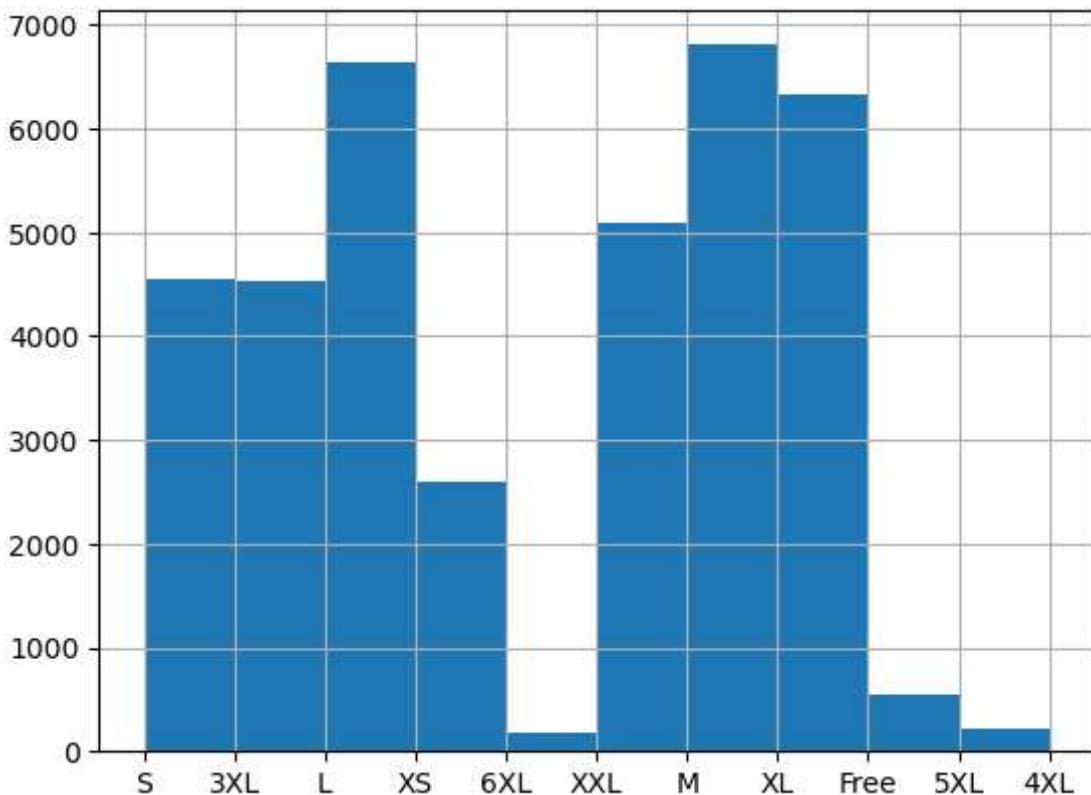
```
In [53]: plt.figure(figsize=(10,5))
sns.countplot(data=df,x='Courier Status',hue= 'Status')
plt.show()
```



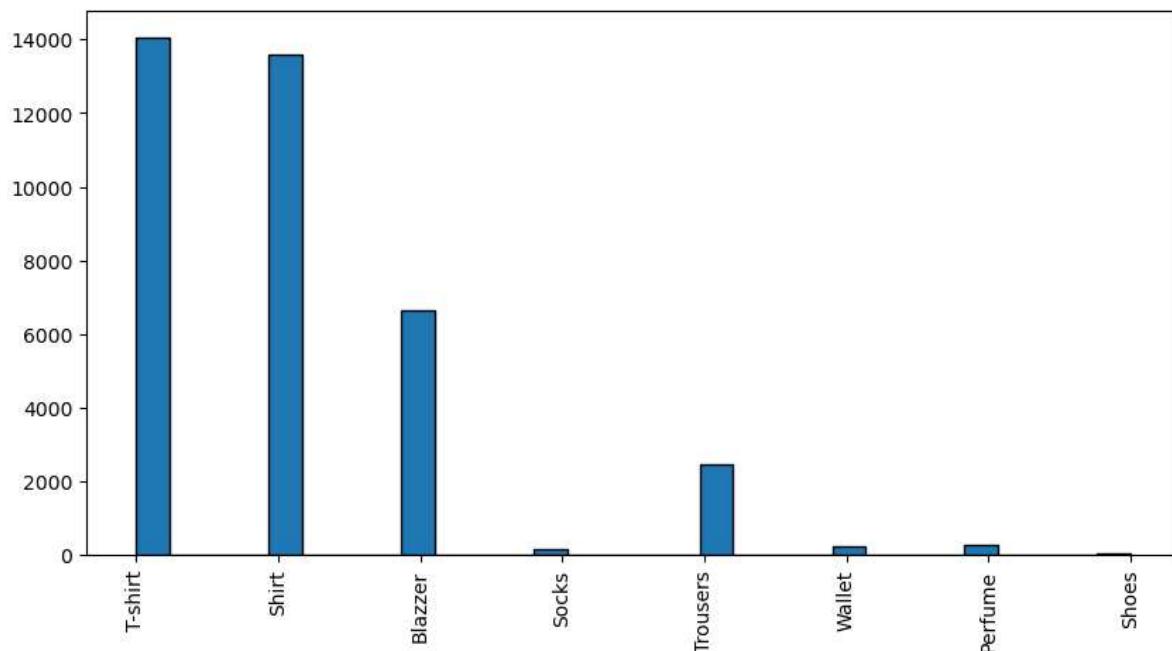
**Note:** From above Graph the majority of the orders are shipped through the counter

```
In [57]: #Histogram
df['Size'].hist()
```

```
Out[57]: <AxesSubplot:>
```



```
In [60]: 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()
```

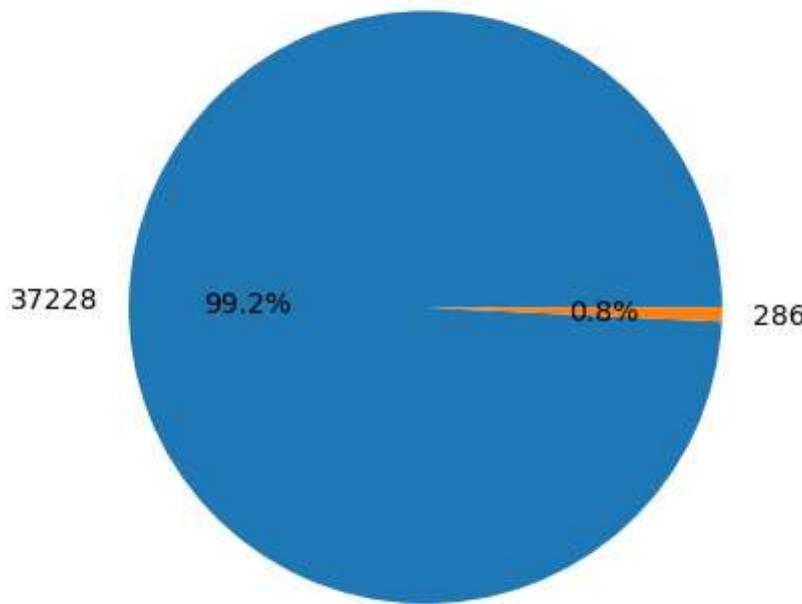


Note: From above Graph you can see that most of the buyers are T-shirt

```
In [62]: # Checking B2B Data by using pie chart
B2B_Check = df['B2B'].value_counts()

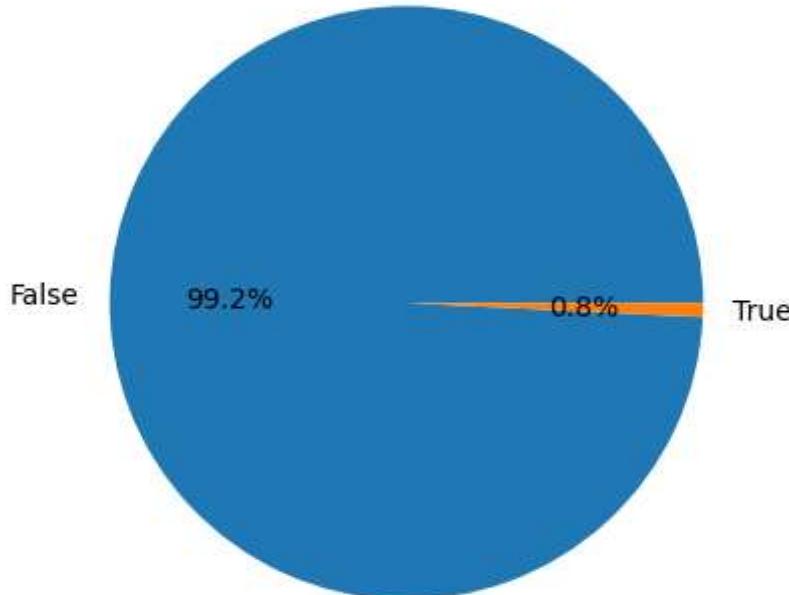
# Plot the pie chart
plt.pie(B2B_Check, labels=B2B_Check, autopct='%1.1f%%')
```

```
# Plt.axis('equal')
plt.show()
```



```
In [63]: # Checking B2B Data by using pie chart
B2B_Check = df['B2B'].value_counts()

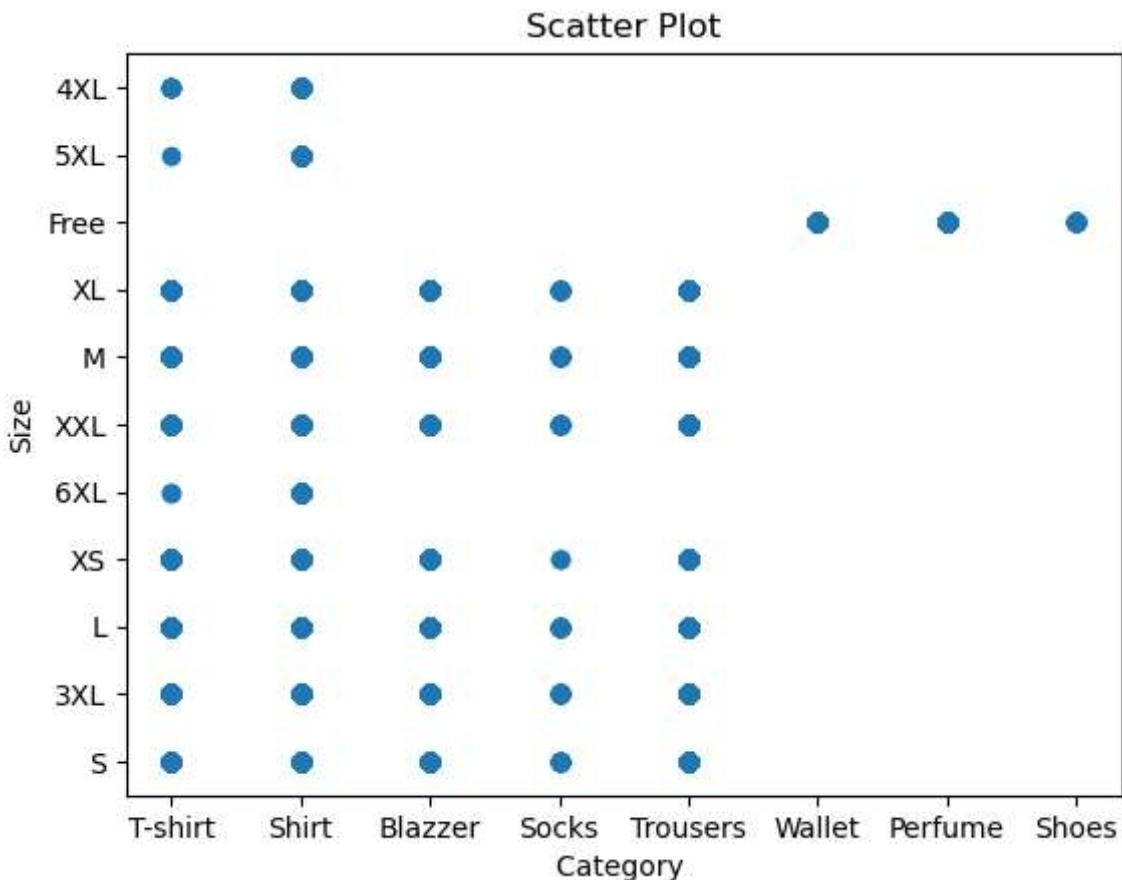
# Plot the pie chart
plt.pie(B2B_Check, labels=B2B_Check.index, autopct='%1.1f%%')
# Plt.axis('equal')
plt.show()
```



Note : From above chart we can see that maximum i.e. 99.3% of buyers are retailers and 0.7% are B2B buyers

```
In [64]: # 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()
```

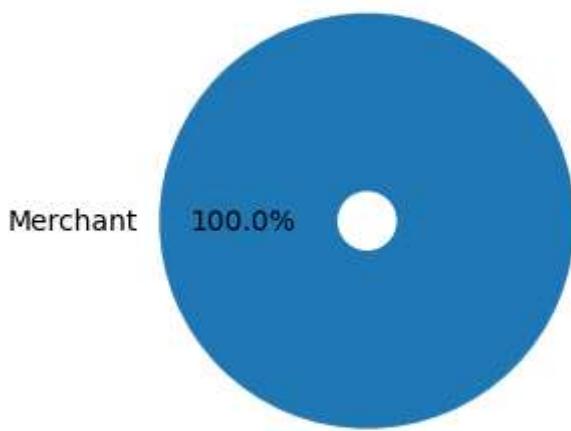


```
In [65]: # Prepare data for pie chart
a1 = df['Fulfilment'].value_counts()

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

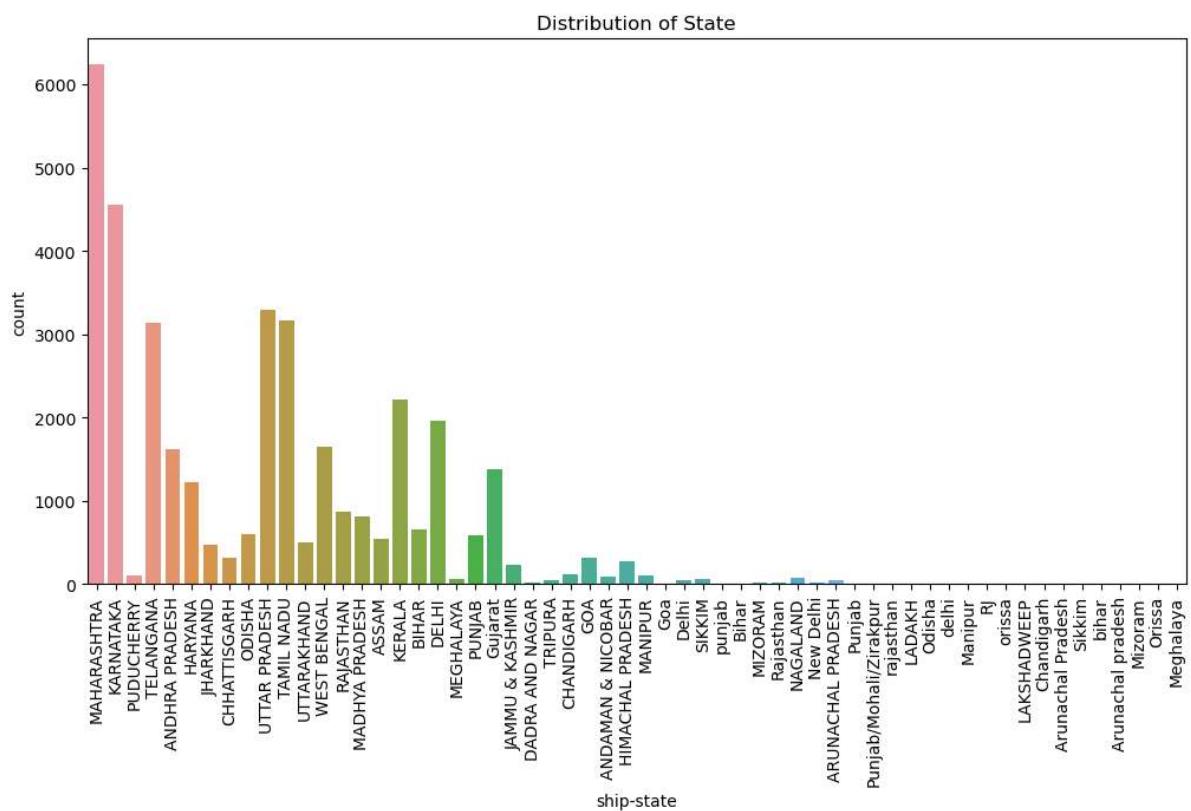
ax.pie(a1, labels=a1.index, autopct='%1.1f%%', radius=0.7, wedgeprops=dict(width=0.
ax.set(aspect="equal")

plt.show()
```

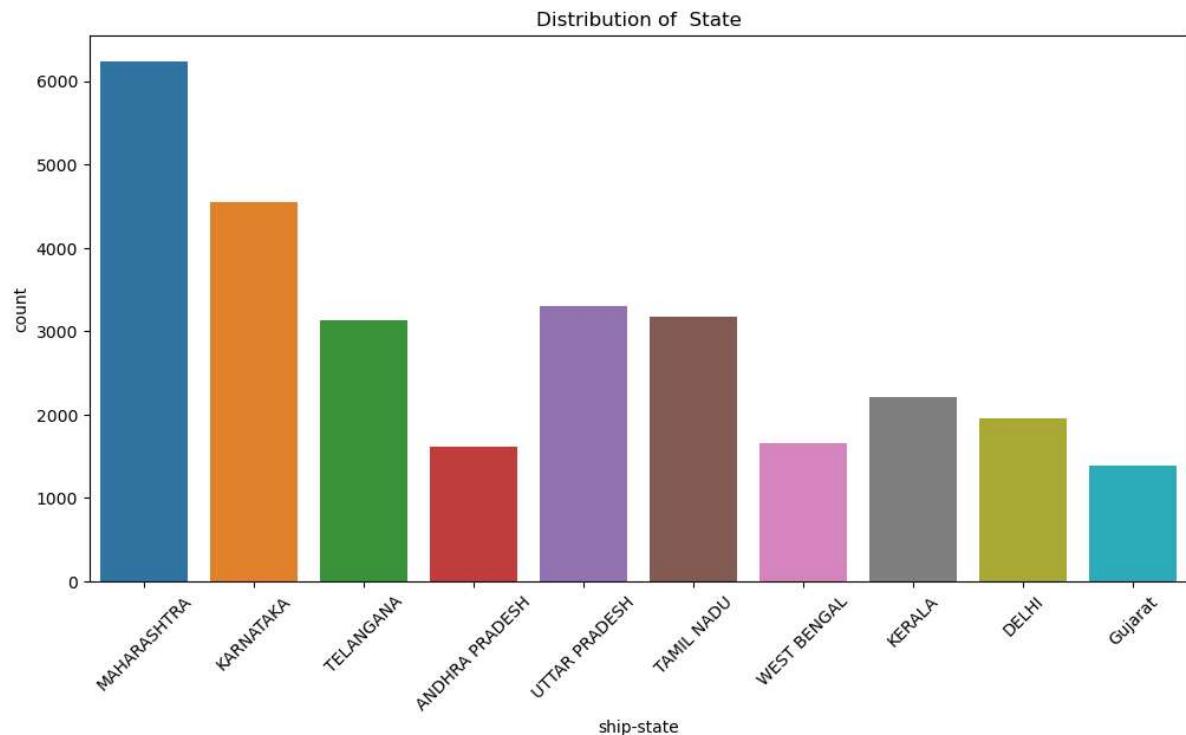


Note: From above chart you can see that most of the Fulfilment are amazon

```
In [66]: # 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()
```



```
In [67]: # 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()
```



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 [ ]:
```