Introduction

The "Amazon Sales Analysis" project is a Python-based data analysis endeavor aimed at exploring and understanding sales data obtained from Amazon.

The project employs various Python libraries, including NumPy, Pandas, Matplotlib, and Seaborn, to analyze and visualize the dataset.

Data Loading and Inspection

The first step involves loading the sales data from a CSV file using Pandas.

The dataset contains 128,976 entries with 21 columns, including information such as Order ID, Date, Status, Sales Channel, Quantity, Amount, and more. Initial inspection using methods like `head()`, `info()`, and `shape` provides a quick overview of the data structure.

Task & Goals for entire analysis

Exploratory Data Analysis (EDA)

- Size Analysis
- Grouping by Size
- Courier Status and Order Status
- Category Distribution
- B2B Analysis
- Fulfilment Analysis
- State - Wise Distribution

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

In [2]:
```
df= pd.read_csv("Amazon Sale Report.csv")
```

In [3]:
```
df.head()
```

Out[3]:

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

5 rows × 21 columns

In [4]:
```
df.shape
```
Out[4]:
```
(128976, 21)
```

In [5]:
```
df.size
```
Out[5]:
```
2708496
```

In [6]:
```
df.ndim
```
Out[6]:
```
2
```

In [7]:
```
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 [8]:
```
# Check for Unwanted Columns

df.head(2)
```

Out[8]:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size
0	0	405-8078784-5731545	04-30-22	Cancelled	Merchant	Amazon.in	Standard	T-shirt	S
1	1	171-9198151-1101146	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	3XL

2 rows × 21 columns

In [9]:
```
# We have two unwanted column named as ["New" & "PendingS"]

df.drop(["New", "PendingS"], axis=1, inplace = True)
```

In [10]:
```
df.head(2)
```

Out[10]:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	C
0	0	405-8078784-5731545	04-30-22	Cancelled	Merchant	Amazon.in	Standard	T-shirt	S	
1	1	171-9198151-1101146	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	3XL	S

In [11]:
```
# Check for Null Values
df.isnull() # 1st method
```

Out[11]:

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

128976 rows × 19 columns

In [12]:
```
pd.isnull(df) # Second Method
```

Out[12]:

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

128976 rows × 19 columns

In [13]:
```
# Total null values
df.isnull().sum()
```

Out[13]:
```
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 [14]:
```
# Drop all null values
df.dropna(inplace=True)
```

In [15]:
```
df.shape
```

Out[15]:
```
(37514, 19)
```

In [16]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 37514 entries, 0 to 128892
Data columns (total 19 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   index               37514 non-null  int64  
 1   Order ID            37514 non-null  object 
 2   Date                37514 non-null  object 
 3   Status              37514 non-null  object 
 4   Fulfilment          37514 non-null  object 
 5   Sales Channel       37514 non-null  object 
 6   ship-service-level  37514 non-null  object 
 7   Category            37514 non-null  object 
 8   Size                37514 non-null  object 
 9   Courier Status      37514 non-null  object 
 10  Qty                 37514 non-null  int64  
 11  currency            37514 non-null  object 
 12  Amount              37514 non-null  float64
 13  ship-city           37514 non-null  object 
 14  ship-state          37514 non-null  object 
 15  ship-postal-code    37514 non-null  float64
 16  ship-country        37514 non-null  object 
 17  B2B                 37514 non-null  bool   
 18  fulfilled-by        37514 non-null  object 
dtypes: bool(1), float64(2), int64(2), object(14)
memory usage: 5.5+ MB
```

In [17]:
```
# Change data type

df["ship-postal-code"]=df["ship-postal-code"].astype('int')
```

In [18]:
```
# Check whether the data type change or not

df["ship-postal-code"].dtype
```

Out[18]:
```
dtype('int64')
```

In [19]:
```
# Convert the date object to datetime formate

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

```
C:\Users\sanad\AppData\Local\Temp\ipykernel_15220\1378184051.py:3: UserWarning: C
ould not infer format, so each element will be parsed individually, falling back
to `dateutil`. To ensure parsing is consistent and as-expected, please specify a
format.
  df['Date'] = pd.to_datetime(df['Date'])
```

In [20]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 37514 entries, 0 to 128892
Data columns (total 19 columns):
 #   Column              Non-Null Count  Dtype         
---  ------              --------------  -----         
 0   index               37514 non-null  int64         
 1   Order ID            37514 non-null  object        
 2   Date                37514 non-null  datetime64[ns]
 3   Status              37514 non-null  object        
 4   Fulfilment          37514 non-null  object        
 5   Sales Channel       37514 non-null  object        
 6   ship-service-level  37514 non-null  object        
 7   Category            37514 non-null  object        
 8   Size                37514 non-null  object        
 9   Courier Status      37514 non-null  object        
 10  Qty                 37514 non-null  int64         
 11  currency            37514 non-null  object        
 12  Amount              37514 non-null  float64       
 13  ship-city           37514 non-null  object        
 14  ship-state          37514 non-null  object        
 15  ship-postal-code    37514 non-null  int64         
 16  ship-country        37514 non-null  object        
 17  B2B                 37514 non-null  bool          
 18  fulfilled-by        37514 non-null  object        
dtypes: bool(1), datetime64[ns](1), float64(1), int64(3), object(13)
memory usage: 5.5+ MB
```

In [21]:
```
# Rename the column name

df.rename(columns={'Qty':'Quantity'})
```

Out[21]:

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

37514 rows × 19 columns

In [22]:

```
# we cannot use this for object. only for numbers
# describe() method return description of the data in the DataFrame(i.e count, m
df.describe()
```

Out[22]:

	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 [23]:
```
df.describe(include='object')
```

Out[23]:

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

In [24]:
```
df['Amount'].describe()
```

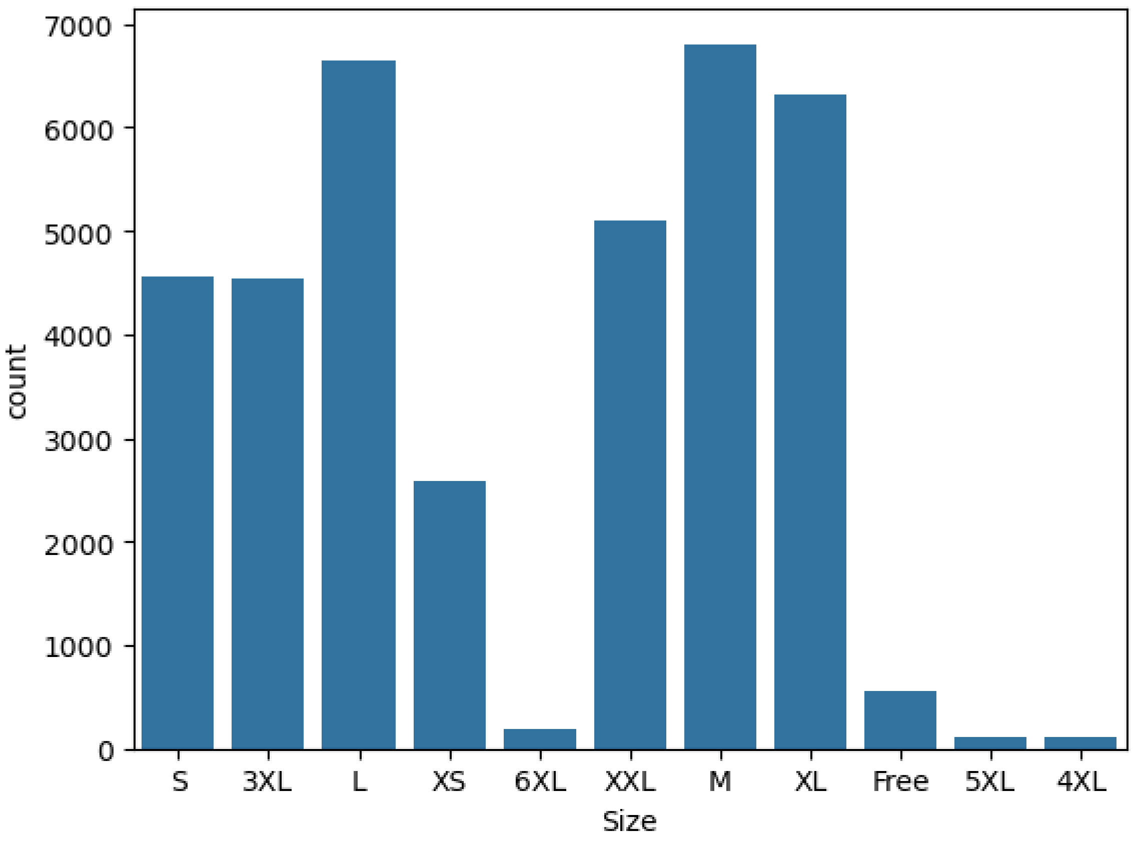
Out[24]:
```
count    37514.000000
mean       646.553960
std        279.952414
min          0.000000
25%        458.000000
50%        629.000000
75%        771.000000
max       5495.000000
Name: Amount, dtype: float64
```

Exploratory Data Analysis

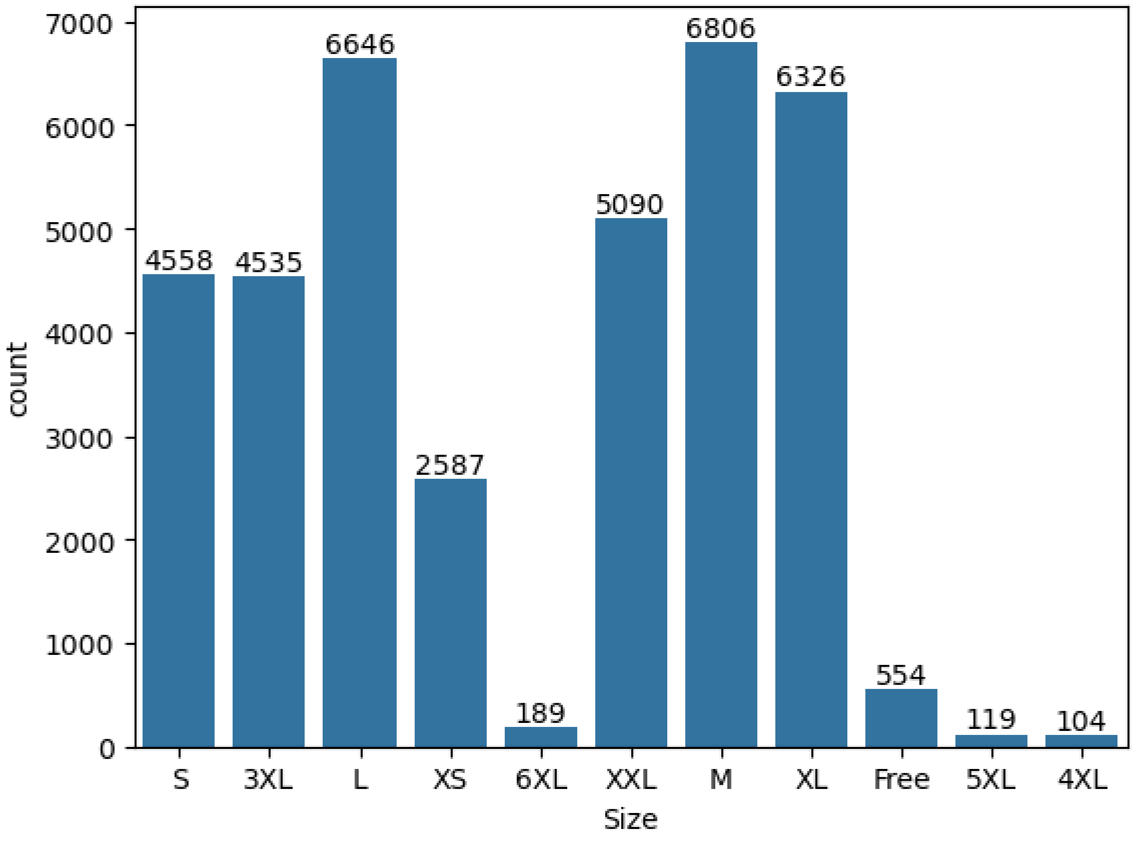
In [25]:
```
df.columns
```

Out[25]:
```
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 [26]:
```
# Check for number of sizes available in the data

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



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

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



GroupBy () Function

It is use to group data based on one or more columns in DataFrame.

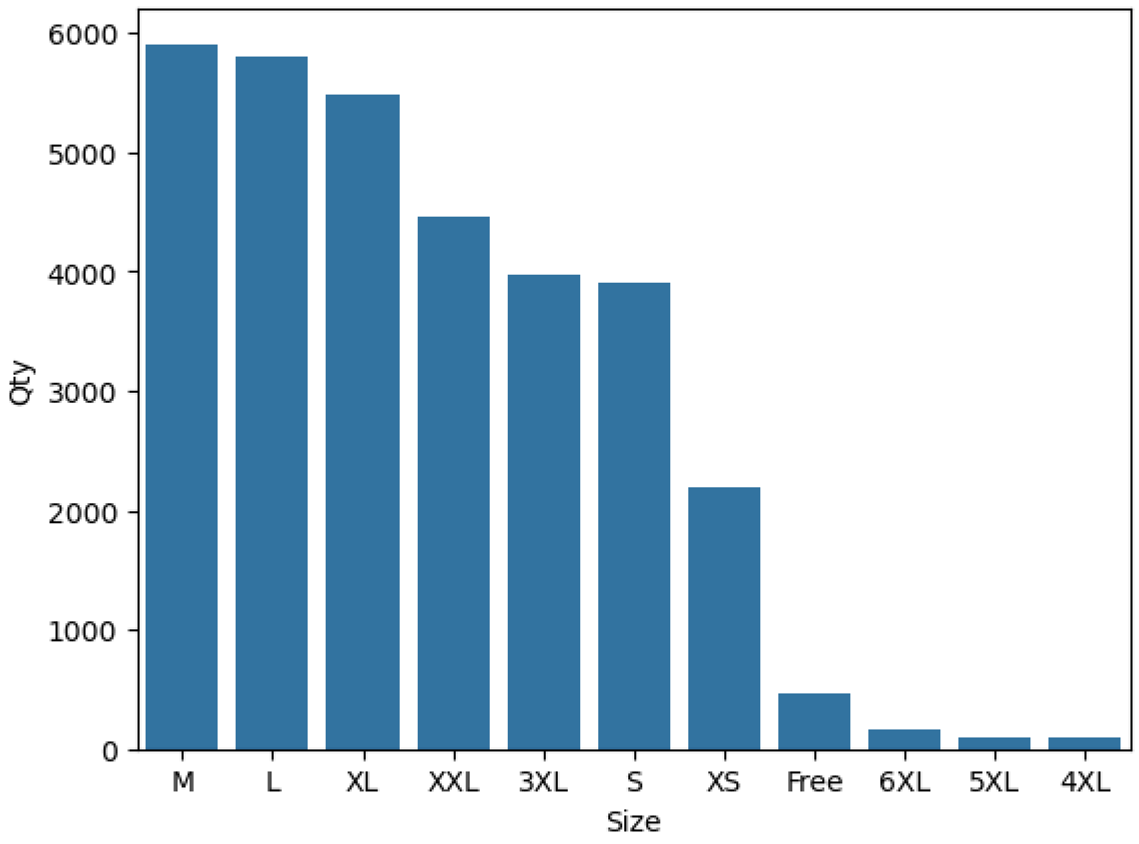
In [28]:
```
df.groupby(['Size'], as_index=False)['Qty'].sum().sort_values(by="Qty", ascendin
```

Out[28]:

	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 [30]:
```
# Graphical representation of abobe result
quant= df.groupby(['Size'], as_index=False)['Qty'].sum().sort_values(by="Qty", a
sns.barplot(x='Size', y='Qty', data=quant)
```

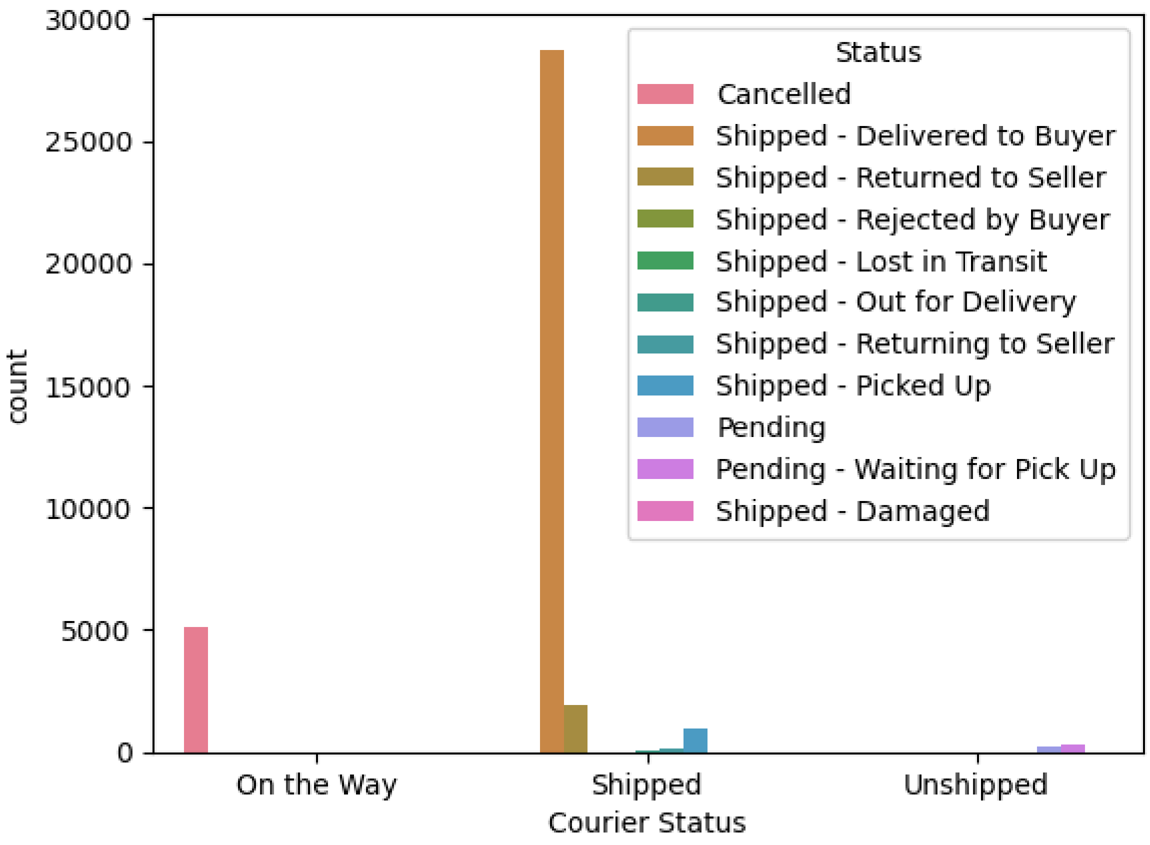
```
Out[30]: <Axes: xlabel='Size', ylabel='Qty'>
```



Courier Status

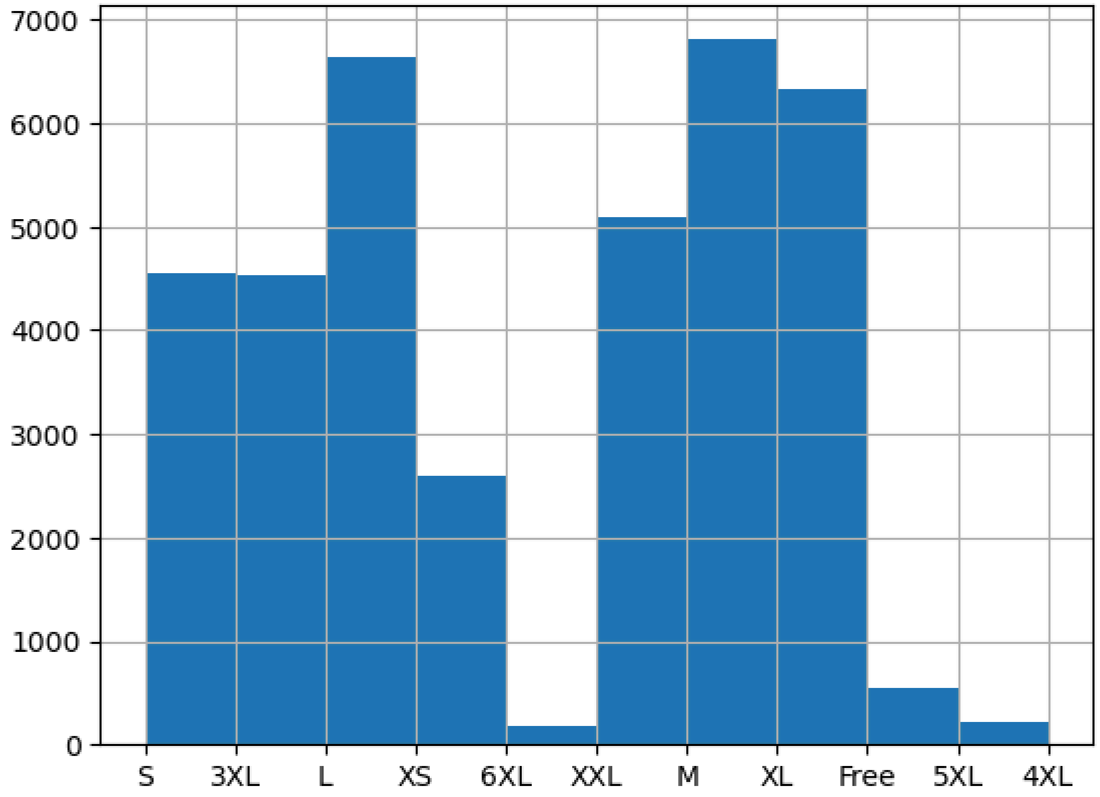
```
In [31]: sns.countplot(data=df, x='Courier Status', hue='Status')
```

```
Out[31]: <Axes: xlabel='Courier Status', ylabel='count'>
```



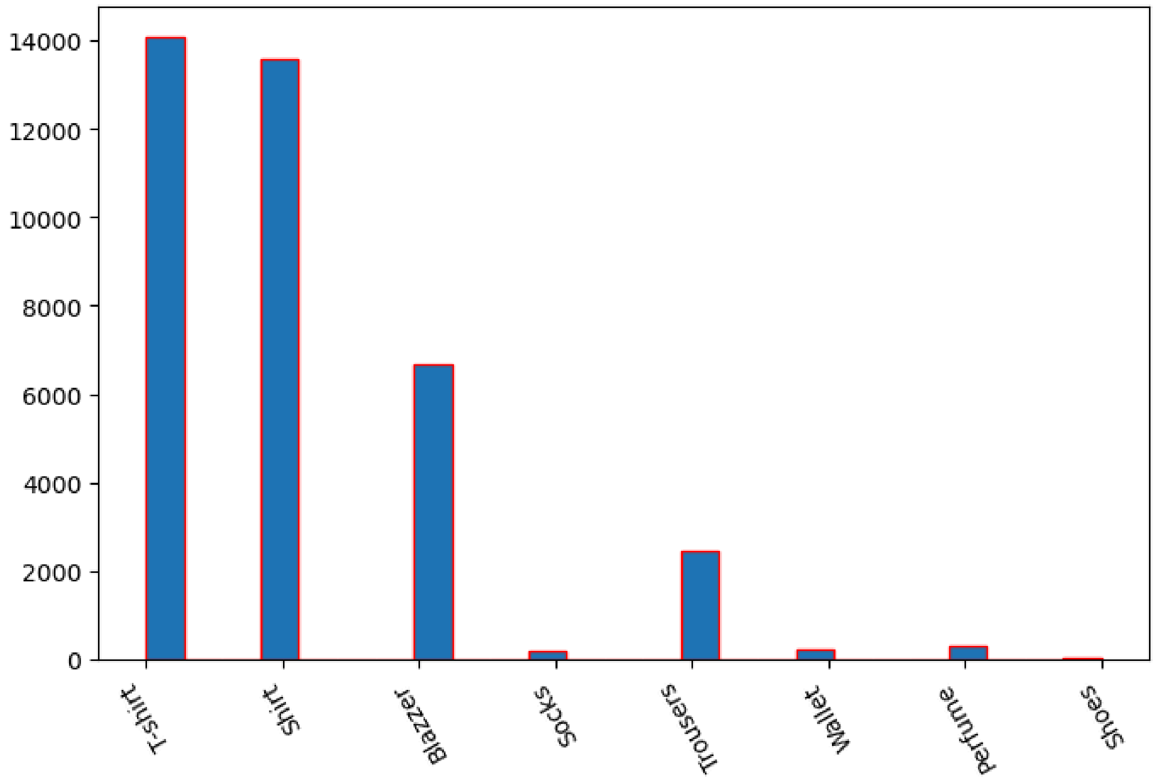
In [32]:
```
# Prepare a histogram on Size column

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

Out[32]:
```
<Axes: >
```

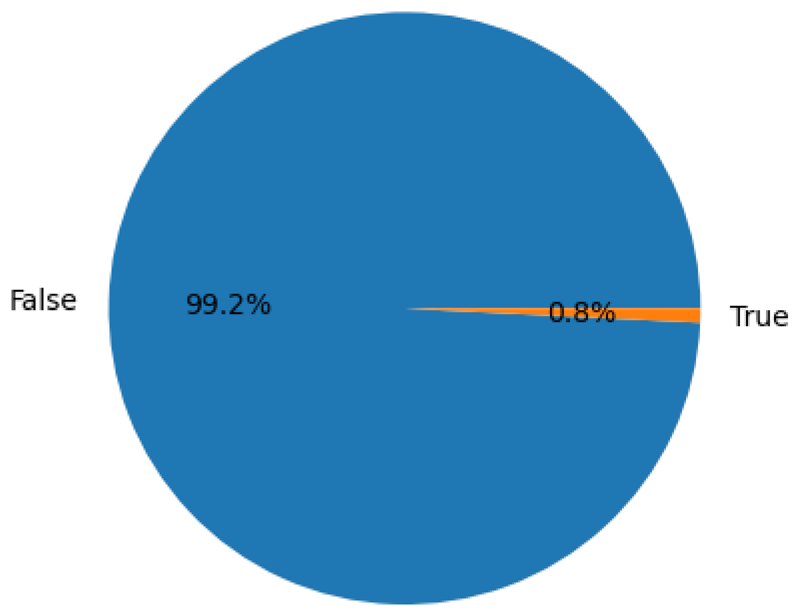


In [36]:
```
df['Category'] = df['Category'].astype(str)
column_data = df['Category']
```

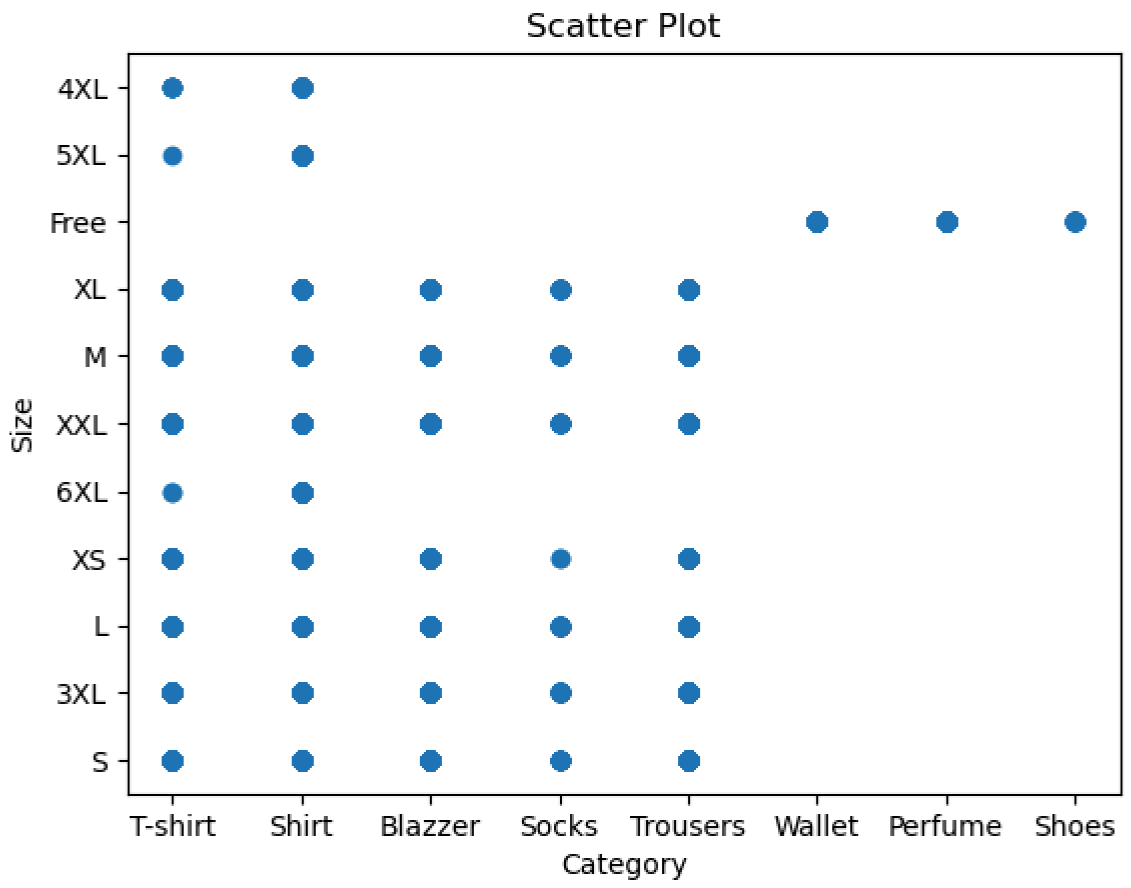
```
plt.figure(figsize=(8,5))
plt.hist(column_data, bins=25, edgecolor='Red')
plt.xticks(rotation = 120)
plt.show()
```



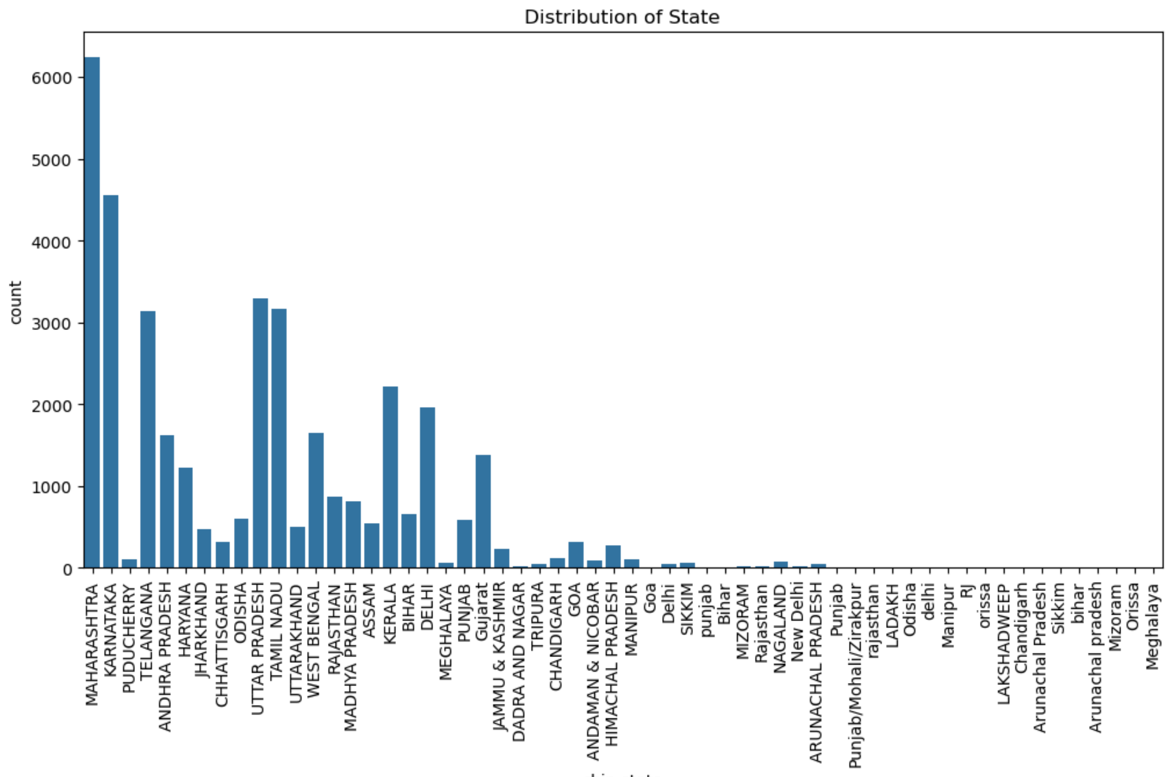
In [37]:
```
# checking B2B Data using pie chart
B2B_Check = df['B2B'].value_counts()

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

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



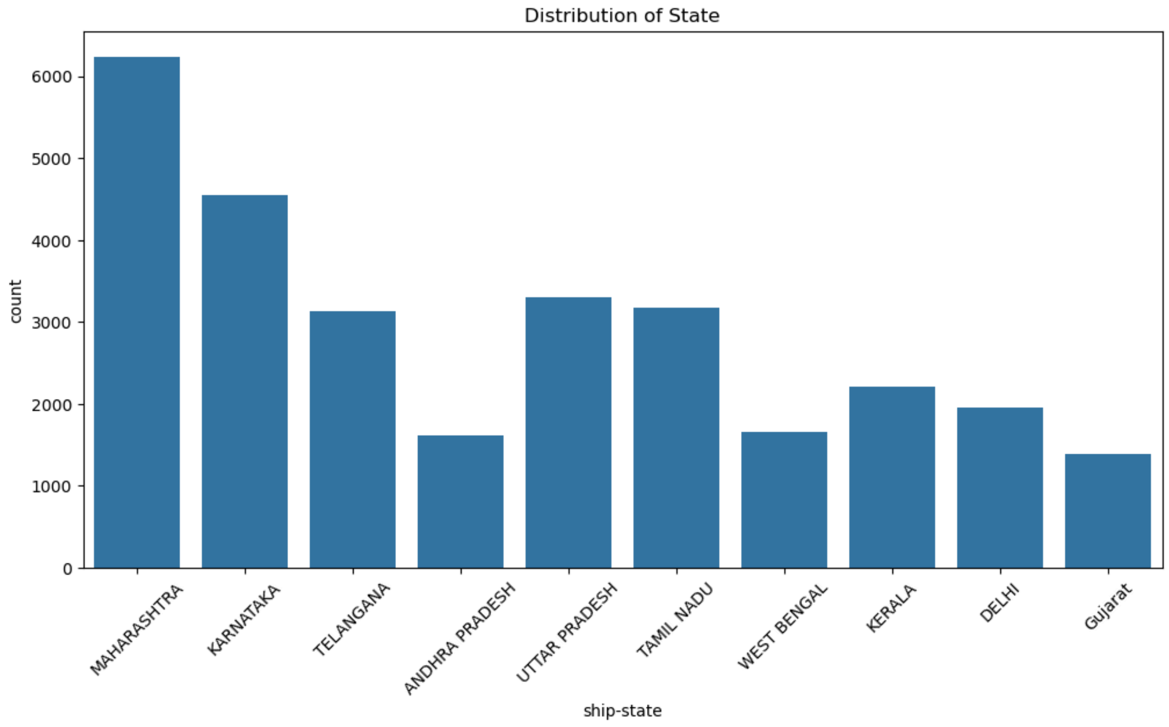
In [38]:
```python
# Represent how many sizes are available for different category

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

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



In [39]:
```
# 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 [40]:
```python
# Top 10 states
top_10_states = 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_states.index)], x= 'ship-stat
plt.xlabel('ship-state')
plt.ylabel('count')
plt.title('Distribution of State')
plt.xticks(rotation=45)
plt.show()
```



Conclusion::

- Most of the people buy M-size.
- The majority of the orders are shipped through the courier.
- Most of the buyers buys T-shirt.
- We can observe that, maximum (i.e 99.2%) buyers are retailers and (0.8%) are B2B buyers.
- Most of the buyers are from Maharashtra state.

In [ ]:
```
# reference - https://www.youtube.com/watch?v=1TmrFEHTg54
```