In [1]:
```
#Importing python libraries

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

In [2]:
```
#importing data

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

Out[2]:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	...	currency	Amount	
0	0	405-8078784-5731545	04-30-22	Cancelled	Merchant	Amazon.in	Standard	T-shirt	S	On the Way	...	INR	647.62	
1	1	171-9198151-1101146	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	3XL	Shipped	...	INR	406.00	BE
2	2	404-0687676-7273146	04-30-22	Shipped	Amazon	Amazon.in	Expedited	Shirt	XL	Shipped	...	INR	329.00	NA
3	3	403-9615377-8133951	04-30-22	Cancelled	Merchant	Amazon.in	Standard	Blazzer	L	On the Way	...	INR	753.33	PUD
4	4	407-1069790-7240320	04-30-22	Shipped	Amazon	Amazon.in	Expedited	Trousers	3XL	Shipped	...	INR	574.00	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
128971	128970	406-6001380-7673107	05-31-22	Shipped	Amazon	Amazon.in	Expedited	Shirt	XL	Shipped	...	INR	517.00	HY
128972	128971	402-9551604-7544318	05-31-22	Shipped	Amazon	Amazon.in	Expedited	T-shirt	M	Shipped	...	INR	999.00	GI
128973	128972	407-9547469-3152358	05-31-22	Shipped	Amazon	Amazon.in	Expedited	Blazzer	XXL	Shipped	...	INR	690.00	HY
128974	128973	402-6184140-0545956	05-31-22	Shipped	Amazon	Amazon.in	Expedited	T-shirt	XS	Shipped	...	INR	1199.00	
128975	128974	408-7436540-8728312	05-31-22	Shipped	Amazon	Amazon.in	Expedited	T-shirt	S	Shipped	...	INR	696.00	

128976 rows × 21 columns

In [3]:
```
#checking no.of rows and columns

df.shape
```

Out[3]:
```
(128976, 21)
```

In [4]:
```
#To check first 5 records

df.head()
```

Out[4]:

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

5 rows × 21 columns

In [5]:
```
#to check last 5 records

df.tail()
```

Out[5]:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	...	currency	Amount	sl
128971	128970	406-6001380-7673107	05-31-22	Shipped	Amazon	Amazon.in	Expedited	Shirt	XL	Shipped	...	INR	517.0	HYDE
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	HYDE
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 [6]:
```
#checking detailed information about dataframes

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 [7]:
```
#dropping columns

df.drop(columns=['New','PendingS'],inplace=True)
df
```

Out[7]:

Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	Qty	currency	Amount	ship-city	ship-state	s po
Cancelled	Merchant	Amazon.in	Standard	T-shirt	S	On the Way	0	INR	647.62	MUMBAI	MAHARASHTRA	4000
Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	3XL	Shipped	1	INR	406.00	BENGALURU	KARNATAKA	5600
Shipped	Amazon	Amazon.in	Expedited	Shirt	XL	Shipped	1	INR	329.00	NAVI MUMBAI	MAHARASHTRA	4102
Cancelled	Merchant	Amazon.in	Standard	Blazzer	L	On the Way	0	INR	753.33	PUDUCHERRY	PUDUCHERRY	6050
Shipped	Amazon	Amazon.in	Expedited	Trousers	3XL	Shipped	1	INR	574.00	CHENNAI	TAMIL NADU	6000
...	...	...	...	...	...	...	...	...	...	...	...	
Shipped	Amazon	Amazon.in	Expedited	Shirt	XL	Shipped	1	INR	517.00	HYDERABAD	TELANGANA	5000
Shipped	Amazon	Amazon.in	Expedited	T-shirt	M	Shipped	1	INR	999.00	GURUGRAM	HARYANA	1220
Shipped	Amazon	Amazon.in	Expedited	Blazzer	XXL	Shipped	1	INR	690.00	HYDERABAD	TELANGANA	5000
Shipped	Amazon	Amazon.in	Expedited	T-shirt	XS	Shipped	1	INR	1199.00	Halol	Gujarat	3893
Shipped	Amazon	Amazon.in	Expedited	T-shirt	S	Shipped	1	INR	696.00	Raipur	CHHATTISGARH	4920

In [8]:
```
#After dropping checking the information of data

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]:
```
#checking null values in columns

df.isnull().sum()
```

Out[9]:
```
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 [10]:
```
#getting unqiue values of all columns

for i in df.columns:
    unique_values=df[i].unique()
    print(f"Unique values in column'{i}':{unique_values}")
```

```
Unique values in column'index':[     0      1      2 ... 128972 128973 128974]
Unique values in column'Order ID':['405-8078784-5731545' '171-9198151-1101146' '404-0687676-7273146'
...
 '407-9547469-3152358' '402-6184140-0545956' '408-7436540-8728312']
Unique values in column'Date':['04-30-22' '04-29-22' '04-28-22' '04-27-22' '04-26-22' '04-25-22'
 '04-24-22' '04-23-22' '04-22-22' '04-21-22' '04-20-22' '04-19-22'
 '04-18-22' '04-17-22' '04-16-22' '04-15-22' '04-14-22' '04-13-22'
 '04-12-2022' '04-11-2022' '04-10-2022' '04-09-2022' '04-08-2022'
 '04-07-2022' '04-06-2022' '04-05-2022' '04-04-2022' '04-03-2022'
 '04-02-2022' '04-01-2022' '03-31-22' '05-31-22' '05-30-22' '05-29-22'
 '05-28-22' '05-27-22' '05-26-22' '05-25-22' '05-24-22' '05-23-22'
 '05-22-22' '05-21-22' '05-20-22' '05-19-22' '05-18-22' '05-17-22'
 '05-16-22' '05-15-22' '05-14-22' '05-13-22' '05-12-2022' '05-11-2022'
 '05-10-2022' '05-09-2022' '05-08-2022' '05-07-2022' '05-06-2022'
 '05-05-2022' '05-04-2022' '05-03-2022' '05-02-2022' '05-01-2022'
 '06-29-22' '06-28-22' '06-27-22' '06-26-22' '06-25-22' '06-24-22'
 '06-23-22' '06-22-22' '06-21-22' '06-20-22' '06-19-22' '06-18-22'
 '06-17-22' '06-16-22' '06-15-22' '06-14-22' '06-13-22' '06-12-2022'
 '06-11-2022' '06-10-2022' '06-09-2022' '06-08-2022' '06-07-2022'
 '06-06-2022' '06-05-2022' '06-04-2022' '06-03-2022' '06-02-2022'
 '06-01-2022']
Unique values in column'Status':['Cancelled' 'Shipped - Delivered to Buyer' 'Shipped'
 'Shipped - Returned to Seller' 'Shipped - Rejected by Buyer'
 'Shipped - Lost in Transit' 'Shipped - Out for Delivery'
 'Shipped - Returning to Seller' 'Shipped - Picked Up' 'Pending'
 'Pending - Waiting for Pick Up' 'Shipped - Damaged' 'Shipping']
Unique values in column'Fulfilment':['Merchant' 'Amazon']
Unique values in column'Sales Channel':['Amazon.in' 'Non-Amazon']
Unique values in column'ship-service-level':['Standard' 'Expedited']
Unique values in column'Category':['T-shirt' 'Shirt' 'Blazzer' 'Trousers' 'Perfume' 'Socks' 'Shoes' 'Wa
llet'
 'Watch']
Unique values in column'Size':['S' '3XL' 'XL' 'L' 'XXL' 'XS' '6XL' 'M' '4XL' 'Free' '5XL']
Unique values in column'Courier Status':['On the Way' 'Shipped' 'Cancelled' 'Unshipped']
Unique values in column'Qty':[ 0  1  2 15  3  9 13  5  4  8]
Unique values in column'currency':['INR' nan]
Unique values in column'Amount':[ 647.62  406.    329.   ...  708.58 1244.    639.  ]
Unique values in column'ship-city':['MUMBAI' 'BENGALURU' 'NAVI MUMBAI' ... 'GULABPURA, Distt BHILWARA'
 'Prayagraj (ALLAHABAD)' 'Halol']
Unique values in column'ship-state':['MAHARASHTRA' 'KARNATAKA' 'PUDUCHERRY' 'TAMIL NADU' 'UTTAR PRADES
H'
 'CHANDIGARH' 'TELANGANA' 'ANDHRA PRADESH' 'RAJASTHAN' 'DELHI' 'HARYANA'
 'ASSAM' 'JHARKHAND' 'CHHATTISGARH' 'ODISHA' 'KERALA' 'MADHYA PRADESH'
 'WEST BENGAL' 'NAGALAND' 'Gujarat' 'UTTARAKHAND' 'BIHAR'
 'JAMMU & KASHMIR' 'PUNJAB' 'HIMACHAL PRADESH' 'ARUNACHAL PRADESH' 'Goa'
 'MEGHALAYA' 'GOA' 'MANIPUR' 'TRIPURA' 'LADAKH' 'DADRA AND NAGAR' 'SIKKIM'
 'Delhi' nan 'ANDAMAN & NICOBAR' 'Punjab' 'Rajshthan' 'Manipur'
 'rajasthan' 'Odisha' 'NL' 'Bihar' 'MIZORAM' 'punjab' 'New Delhi'
 'Rajasthan' 'Punjab/Mohali/Zirakpur' 'Puducherry' 'delhi' 'RJ'
 'Chandigarh' 'orissa' 'LAKSHADWEEP' 'goa' 'PB' 'APO' 'Arunachal Pradesh'
 'AR' 'Pondicherry' 'Sikkim' 'Arunachal pradesh' 'Nagaland' 'bihar'
 'Mizoram' 'rajsthan' 'Orissa' 'Rajsthan' 'Meghalaya']
Unique values in column'ship-postal-code':[400081. 560085. 410210. ... 609603. 851205. 629152.]
Unique values in column'ship-country':['IN' nan]
Unique values in column'B2B':[False  True]
Unique values in column'fulfilled-by':['Easy Ship' nan]
```

In [11]:
```
#dropping null values

df.dropna(inplace=True)
```

In [12]:
```
#checking again null values after dropping

df.isnull().sum()
```

Out[12]:
```
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             0
Amount               0
ship-city            0
ship-state           0
ship-postal-code     0
ship-country         0
B2B                  0
fulfilled-by         0
dtype: int64
```

In [13]:
```
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 [15]:
```
#changing the datatype of column

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

In [17]:
```
df['Date']=pd.to_datetime(df['Date'])
```

```
C:\Users\Manju\AppData\Local\Temp\ipykernel_11612\3023999556.py:1: UserWarning: Could not infer format,
so each element will be parsed individually, falling back to `dateutil`. To ensure parsing is consisten
t and as-expected, please specify a format.
  df['Date']=pd.to_datetime(df['Date'])
```

In [19]:
```
df['Date']=pd.to_datetime(df['Date'],format="Y%-m%-d%")
```

In [20]:
```
#renaming columns from qty to quanity

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

Out[20]:

	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 [21]:
```
df.describe()
```

Out[21]:

	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 [22]:
```
df.drop(columns=['index'],inplace=True)
df
```

Out[22]:

Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	Quantity	currency	Amount	ship-city	ship-state	s po
ancelled	Merchant	Amazon.in	Standard	T-shirt	S	On the Way	0	INR	647.62	MUMBAI	MAHARASHTRA	40(
hipped -)elivered to Buyer	Merchant	Amazon.in	Standard	Shirt	3XL	Shipped	1	INR	406.00	BENGALURU	KARNATAKA	56(
ancelled	Merchant	Amazon.in	Standard	Blazzer	L	On the Way	0	INR	753.33	PUDUCHERRY	PUDUCHERRY	60!
hipped -)elivered to Buyer	Merchant	Amazon.in	Standard	Shirt	S	Shipped	1	INR	399.00	HYDERABAD	TELANGANA	50(
hipped -)elivered to Buyer	Merchant	Amazon.in	Standard	Shirt	XS	Shipped	1	INR	399.00	Amravati.	MAHARASHTRA	44-
...	...	...	...	...	...	...	...	...	...	...	...	
hipped -)elivered to Buyer	Merchant	Amazon.in	Standard	T-shirt	S	Shipped	1	INR	854.00	ALLUR	ANDHRA PRADESH	52-
ancelled	Merchant	Amazon.in	Standard	Blazzer	XL	On the Way	0	INR	734.29	Barabanki	UTTAR PRADESH	22!
hipped -)elivered to Buyer	Merchant	Amazon.in	Standard	Trousers	M	Shipped	1	INR	518.00	NOIDA	UTTAR PRADESH	20'
ancelled	Merchant	Amazon.in	Standard	Wallet	Free	On the Way	0	INR	398.10	MADURAI	TAMIL NADU	62!
hipped -)elivered to Buyer	Merchant	Amazon.in	Standard	Blazzer	M	Shipped	1	INR	721.00	UTTAR BAGDOGRA	WEST BENGAL	73-

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

Out[23]:

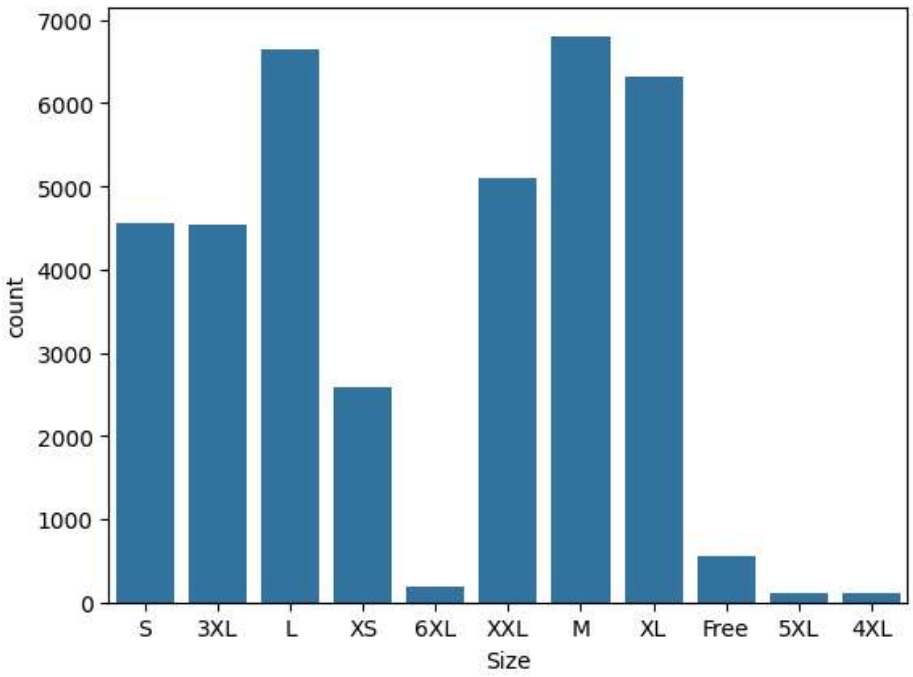
	Order ID	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	currency	ship-city	ship-state
count	37514	37514	37514	37514	37514	37514	37514	37514	37514	37514	37514
unique	34664	11	1	1	1	8	11	3	1	4698	58
top	171-5057375-2831560	Shipped - Delivered to Buyer	Merchant	Amazon.in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

Exploratory Data Analysis (EDA)

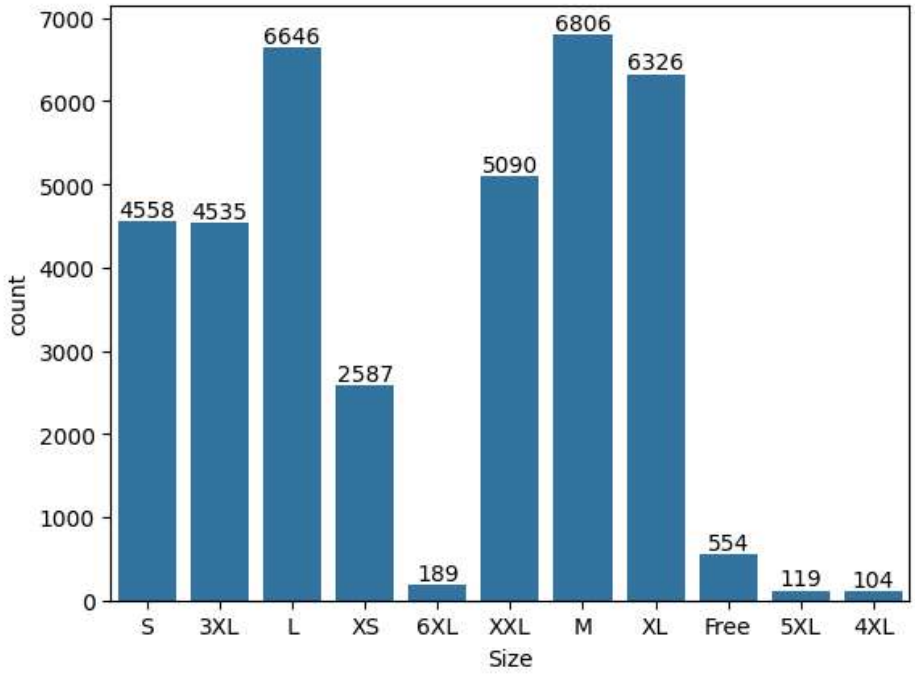
In [24]:
```
#Fetching all columns names

df.columns
```

Out[24]:
```
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 [25]:
```
ax=sns.countplot(x='Size',data=df)
```



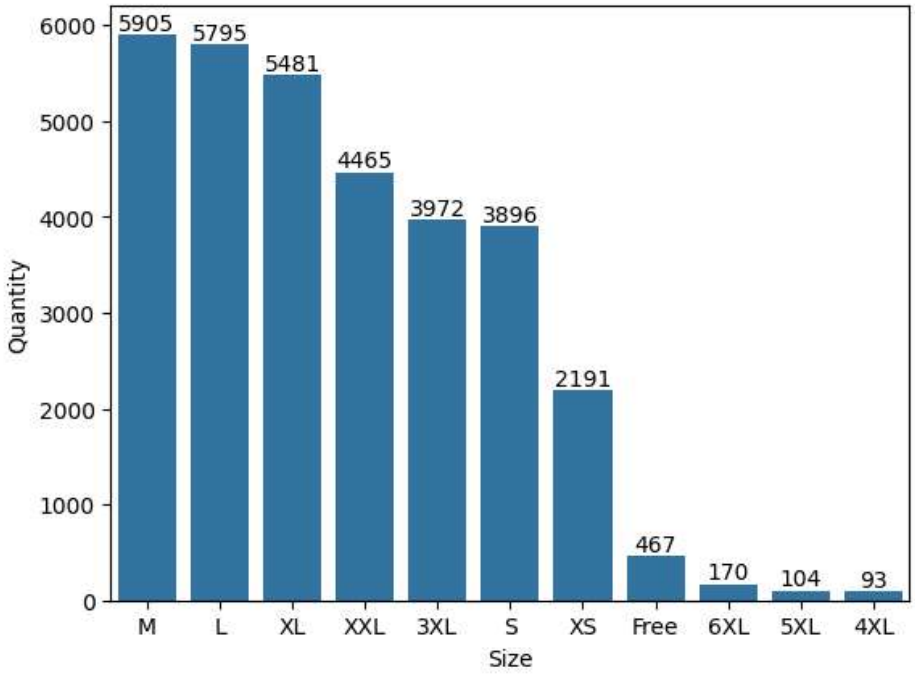
In [28]:
```
#counting size

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


for bars in ax.containers:
    ax.bar_label(bars)  #giving label to graph
```



From above graph we came to know that Size:'M' has most selling

In [29]:
```
Size_Qty=df.groupby(['Size'],as_index=False)['Quantity'].sum().sort_values(by='Quantity',ascending=False
```

In [31]:
```
#Checking size on the basis of quanity sold

qty=sns.barplot(x='Size',y='Quantity',data=Size_Qty)

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

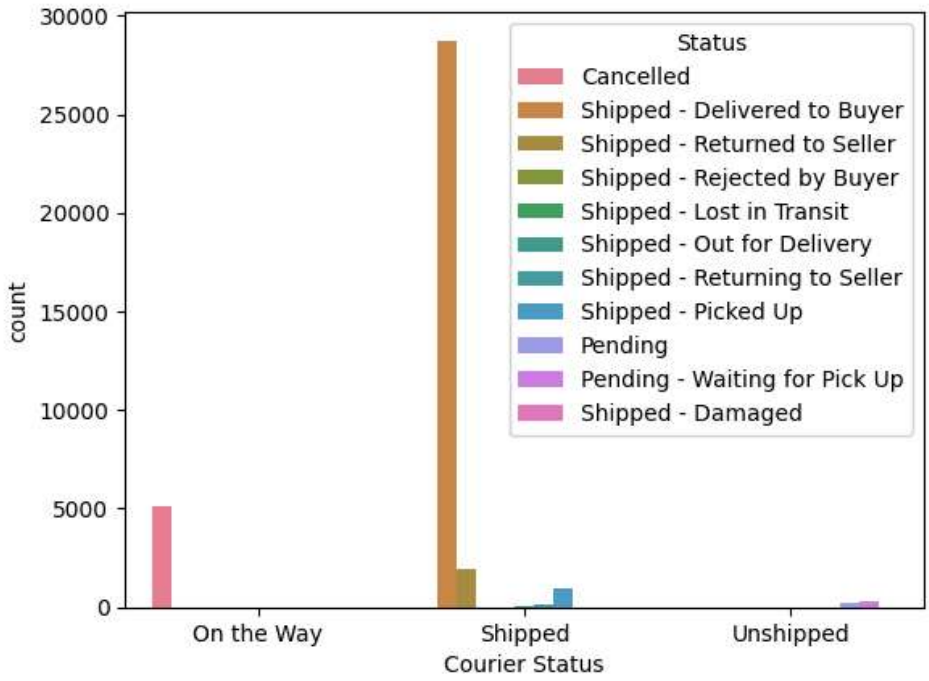


From above graph we can see that Size:'M' has most sales as compared to other size

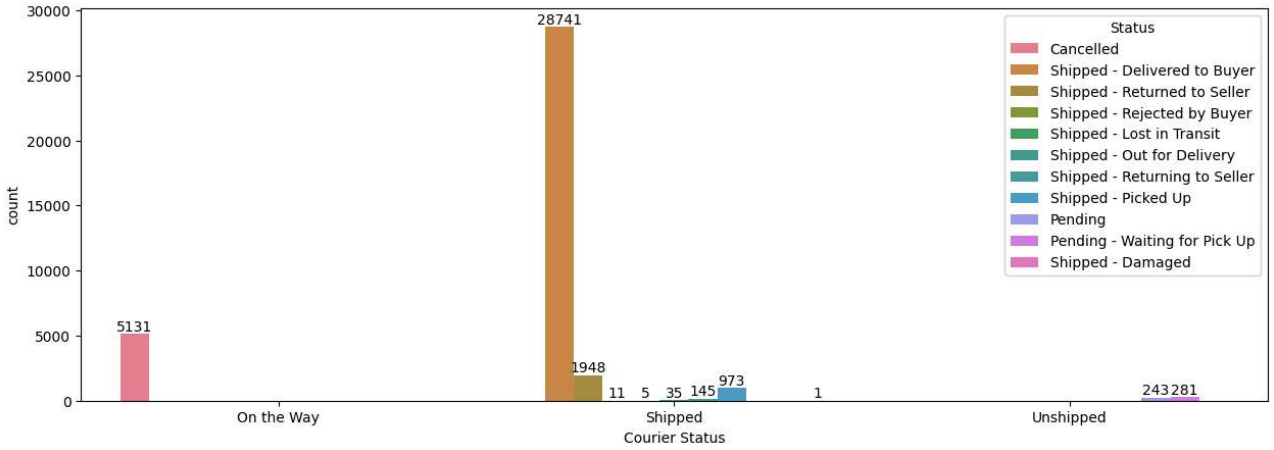
Courier Status

In [32]:
```
sns.countplot(x='Courier Status',hue='Status',data=df)
```

Out[32]:
```
<Axes: xlabel='Courier Status', ylabel='count'>
```



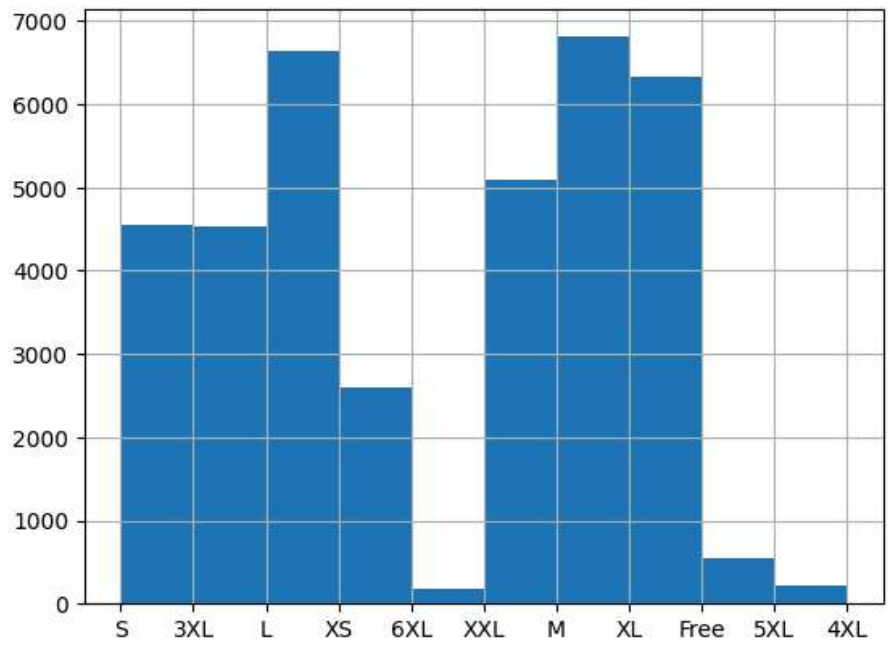
In [36]:
```
plt.figure(figsize=(15,5))

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

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



From above graph we can see that most of courier status are delivered and on the other side we can see that there are items who gets cancelled or hasn't picked up

In [37]:
```
#histogram
df['Size'].hist()
```

Out[37]: <Axes: >

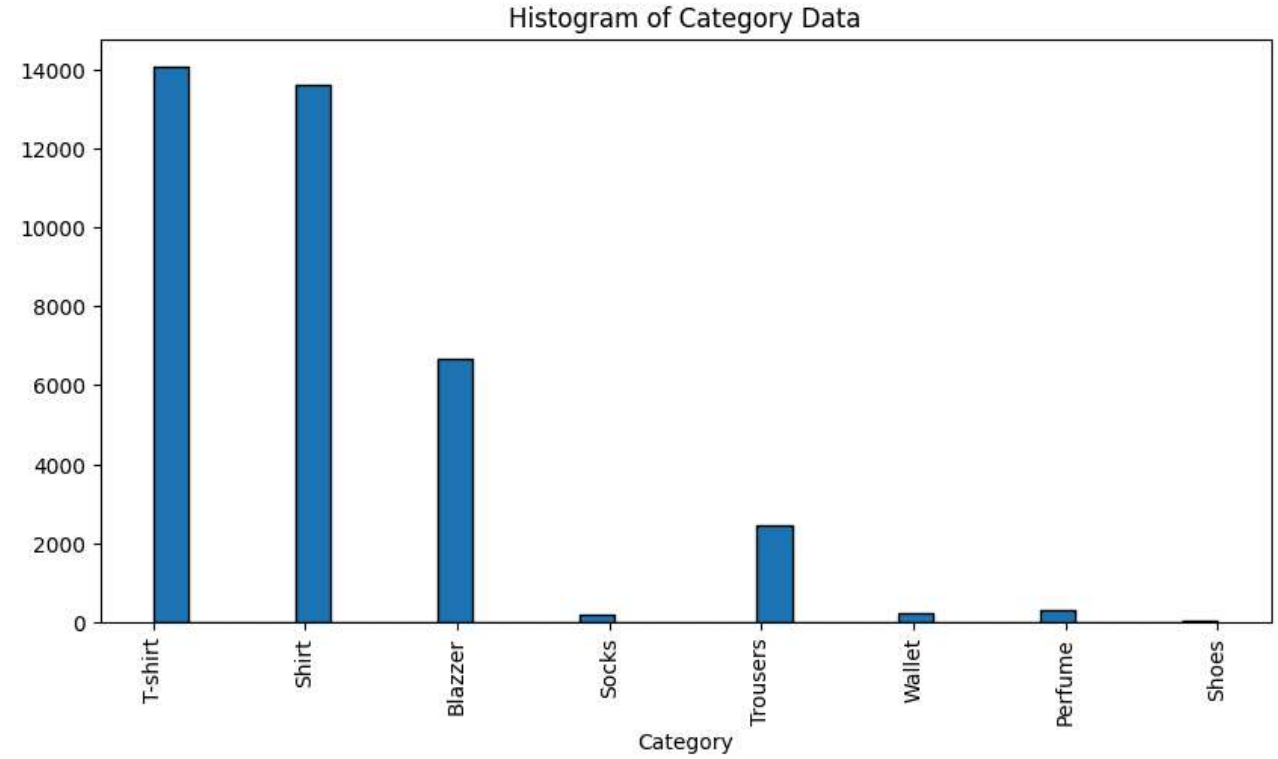


In [38]:
```
df['Category']=df['Category'].astype(str)
```

In [39]:
```
df.info()
```

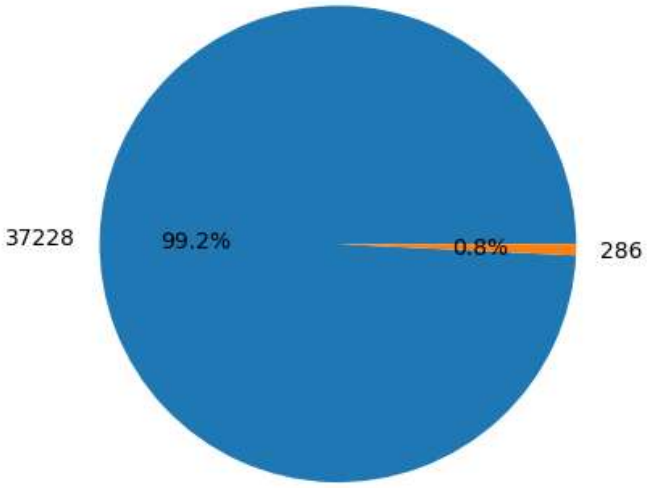
```
<class 'pandas.core.frame.DataFrame'>
Index: 37514 entries, 0 to 128892
Data columns (total 18 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Order ID            37514 non-null  object
 1   Date                37514 non-null  datetime64[ns]
 2   Status              37514 non-null  object
 3   Fulfilment          37514 non-null  object
 4   Sales Channel       37514 non-null  object
 5   ship-service-level  37514 non-null  object
 6   Category            37514 non-null  object
 7   Size                37514 non-null  object
 8   Courier Status      37514 non-null  object
 9   Quantity            37514 non-null  int64
 10  currency            37514 non-null  object
 11  Amount              37514 non-null  float64
 12  ship-city           37514 non-null  object
 13  ship-state          37514 non-null  object
 14  ship-postal-code    37514 non-null  int32
 15  ship-country        37514 non-null  object
 16  B2B                 37514 non-null  bool
 17  fulfilled-by        37514 non-null  object
dtypes: bool(1), datetime64[ns](1), float64(1), int32(1), int64(1), object(13)
memory usage: 5.0+ MB
```

In [44]:
```
column_data=df['Category']
plt.figure(figsize=(10,5))
plt.hist(column_data, bins=30, edgecolor='Black')
plt.xticks(rotation=90)
plt.xlabel('Category')
plt.title('Histogram of Category Data')
plt.show()
```

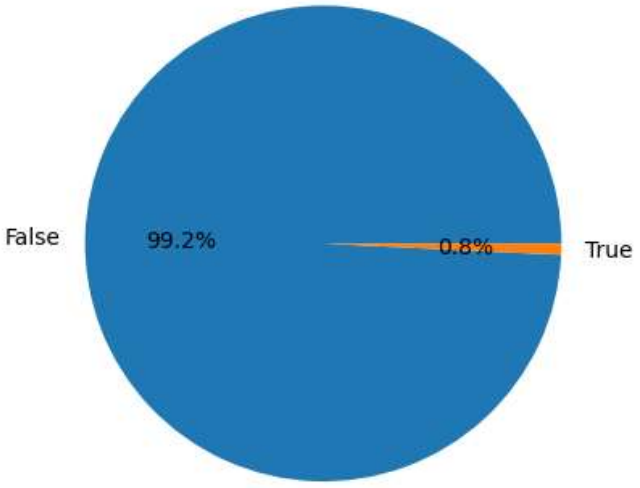


From above graph we can see that buyers mostly buys T-shirt as compared to other category

In [45]:
```
#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()
```

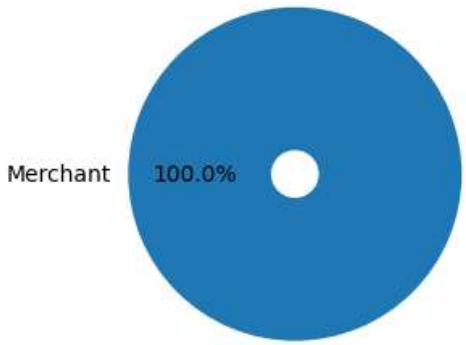


In [46]:
```
#Checking B2B data by using pie chart

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()
```

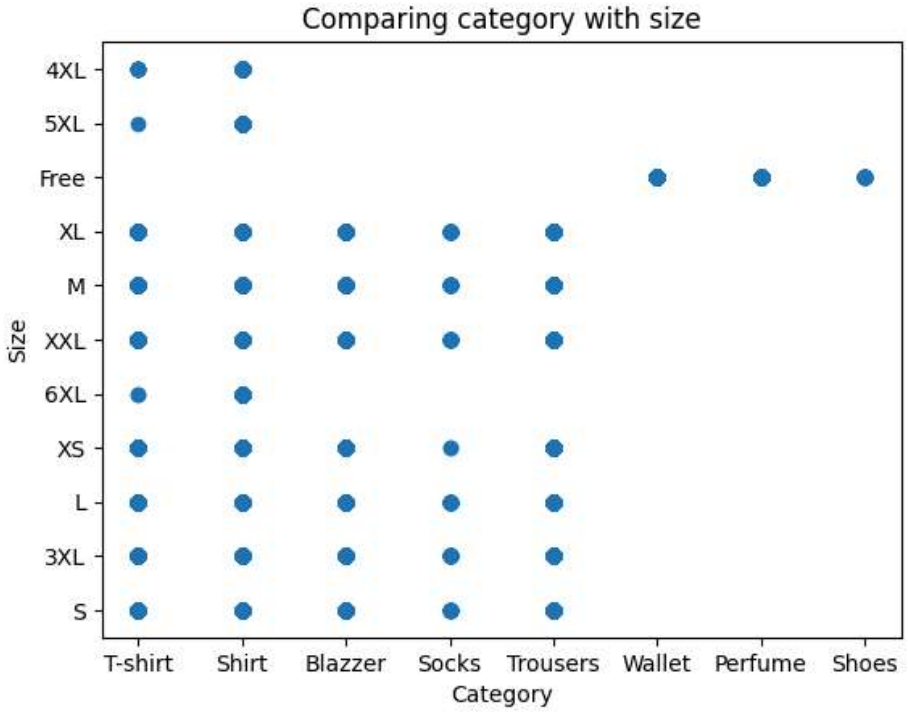


From above chart we can see that maximum i.e. 99.2% of buyers are retailers and 0.8% are B2B Buyers

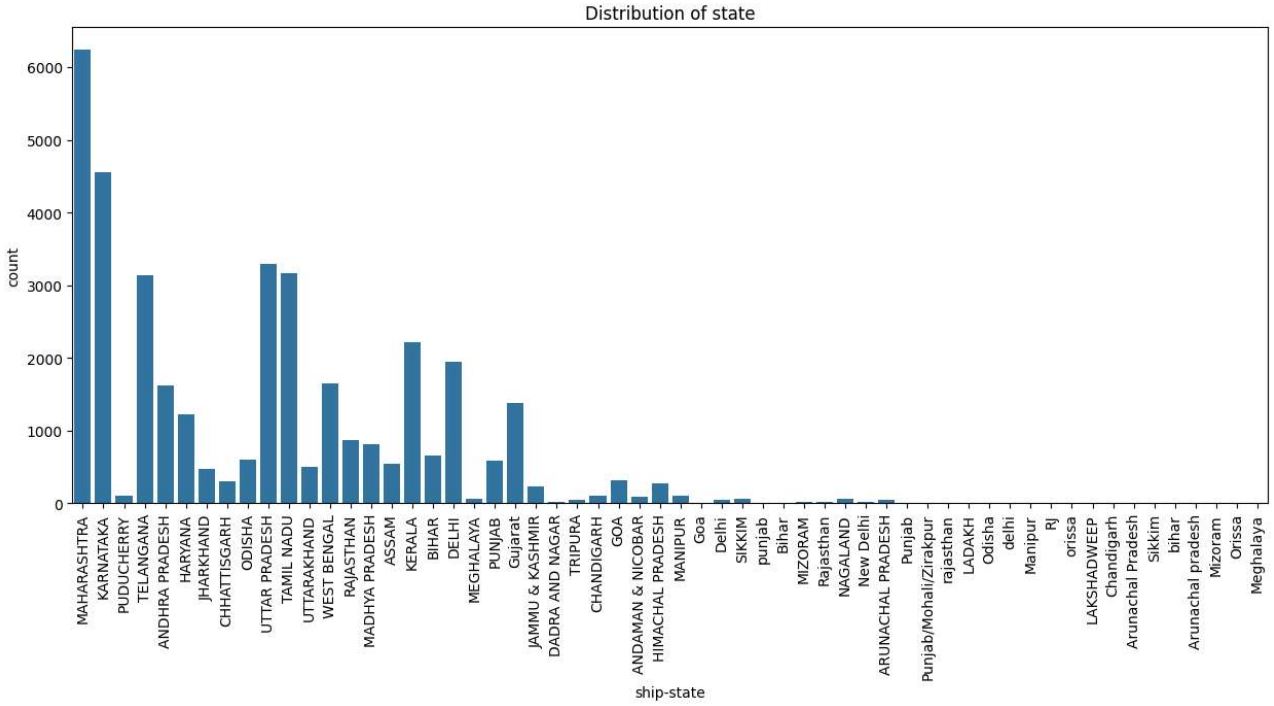
In [53]:
```
#preparing data for piechart

a1=df['Fulfilment'].value_counts()

#plotting 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()
```



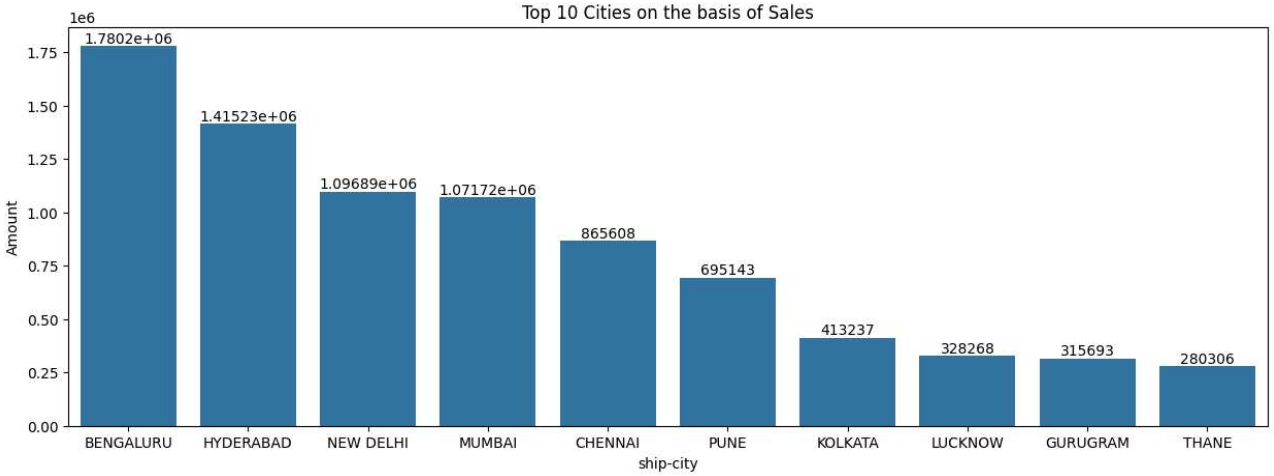
In [54]:
```
#Scatter Plot

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

plt.scatter(x_data,y_data)
plt.xlabel('Category')
plt.ylabel('Size')
plt.title('Comparing category with size')
plt.show()
```



In [56]:
```
plt.figure(figsize=(15,6))
sns.countplot(x='ship-state',data=df)
plt.xlabel('ship-state')
plt.ylabel('count')
plt.title('Distribution of state')
plt.xticks(rotation=90)
plt.show()
```

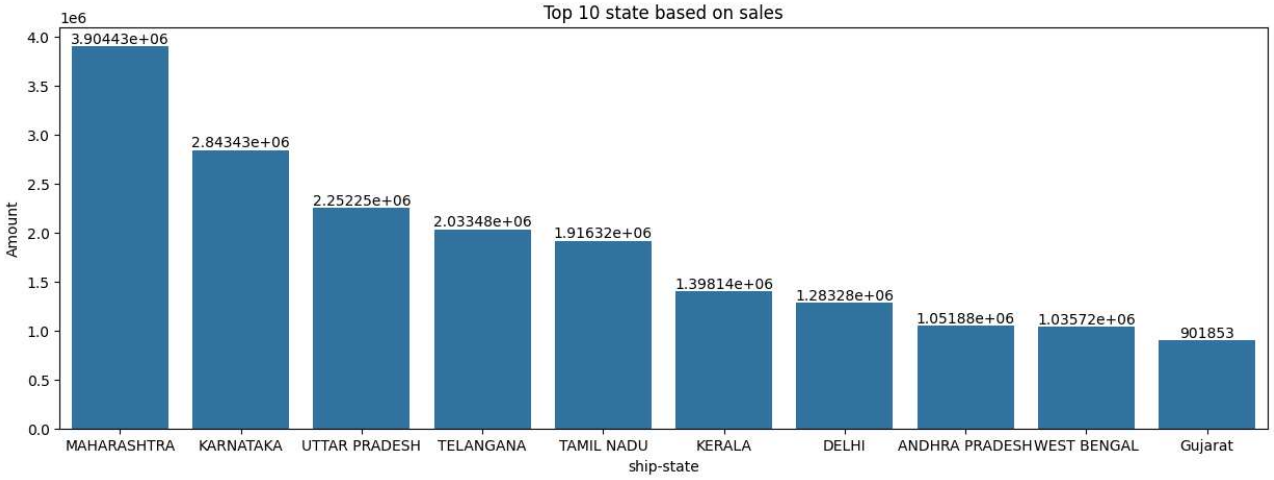


In [63]:
```
#total number of sales by top 10 cities

plt.figure(figsize=(15,5))
sales_city=df.groupby(['ship-city'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=Fa
ca=sns.barplot(x='ship-city',y='Amount',data=sales_city)
for bars in ca.containers:
    ca.bar_label(bars)
plt.title('Top 10 Cities on the basis of Sales')
plt.show()
```



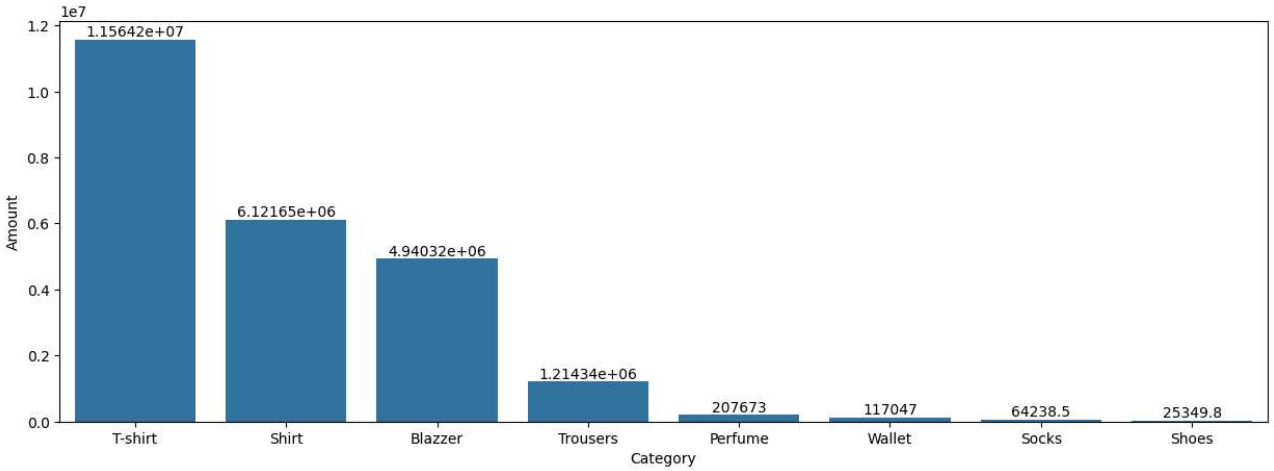
From above graph we can see that city bengaluru has most sales as compared to other city

In [66]:
```
#total number of sales by top 10 state

plt.figure(figsize=(15,5))
sales_state=df.groupby(['ship-state'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=
sa=sns.barplot(x='ship-state',y='Amount',data=sales_state)

#giving labels to data
for bars in sa.containers:
    sa.bar_label(bars)

plt.title('Top 10 state based on sales')
plt.show()
```

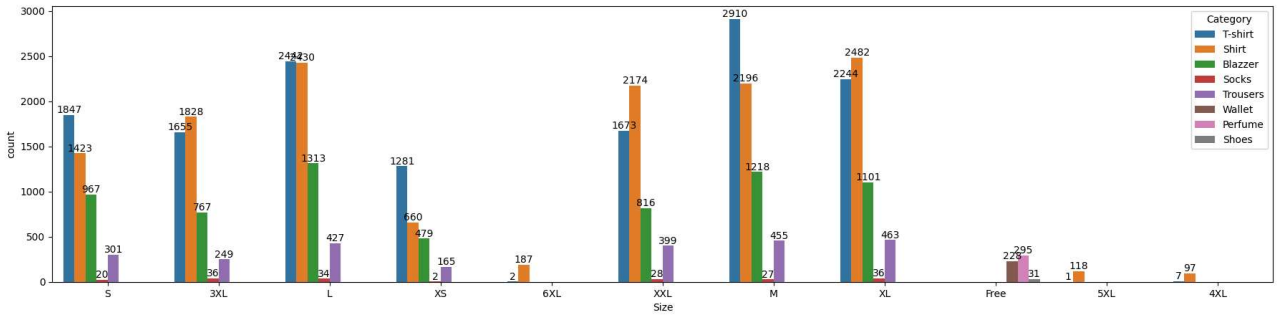


From above graph we can see that state : 'Maharashtra' has most sales as compared to other state

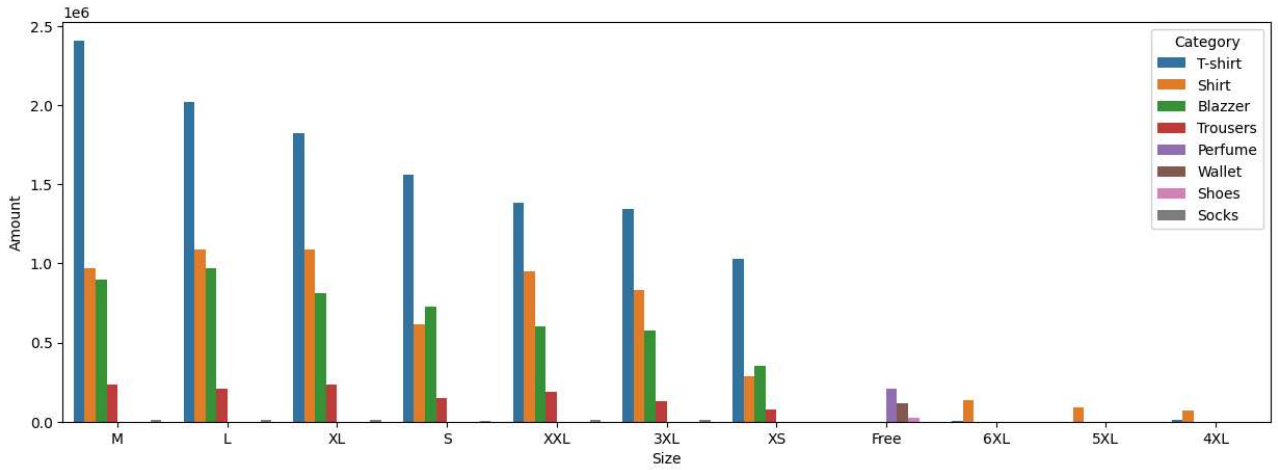
In [68]:
```
#total sales on the basis of category

plt.figure(figsize=(15,5))

cat_sales=df.groupby(['Category'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=Fals
cc=sns.barplot(x='Category',y='Amount',data=cat_sales)

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



In [74]:
```
#setting size of chart
plt.figure(figsize=(22,5))

aa = sns.countplot(data = df, x = 'Size', hue = 'Category')

#giving labels to data
for bars in aa.containers:
    aa.bar_label(bars)
```



In [76]:
```
plt.figure(figsize=(15,5))

size_amount=df.groupby(['Size','Category'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascen
sns.barplot(x='Size',y='Amount',hue='Category',data=size_amount)

plt.show()
```



Above graph showing sales on the basis of size related to category

Conclusion:

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

In [ ]: