

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

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

df.shape

(128976, 21)

df.head()
```

	index	Order ID	Date	Status
\				
0	0	405-8078784-5731545	04-30-22	Cancelled
1	1	171-9198151-1101146	04-30-22	Shipped - Delivered to Buyer
2	2	404-0687676-7273146	04-30-22	Shipped
3	3	403-9615377-8133951	04-30-22	Cancelled
4	4	407-1069790-7240320	04-30-22	Shipped

	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier
Status \						
0	Merchant	Amazon.in	Standard	T-shirt	S	On the Way
1	Merchant	Amazon.in	Standard	Shirt	3XL	Shipped
2	Amazon	Amazon.in	Expedited	Shirt	XL	Shipped
3	Merchant	Amazon.in	Standard	Blazzer	L	On the Way
4	Amazon	Amazon.in	Expedited	Trousers	3XL	Shipped

	...	currency	Amount	ship-city	ship-state	ship-postal-code	\
0	...	INR	647.62	MUMBAI	MAHARASHTRA	400081.0	
1	...	INR	406.00	BENGALURU	KARNATAKA	560085.0	
2	...	INR	329.00	NAVI MUMBAI	MAHARASHTRA	410210.0	
3	...	INR	753.33	PUDUCHERRY	PUDUCHERRY	605008.0	
4	...	INR	574.00	CHENNAI	TAMIL NADU	600073.0	

	ship-country	B2B	fulfilled-by	New	PendingS
0	IN	False	Easy Ship	NaN	NaN
1	IN	False	Easy Ship	NaN	NaN
2	IN	True	NaN	NaN	NaN

3	IN	False	Easy Ship	NaN	NaN
4	IN	False		NaN	NaN

[5 rows x 21 columns]

df.tail()

	index	Order ID	Date	Status	Fulfilment	\
128971	128970	406-6001380-7673107	05-31-22	Shipped	Amazon	
128972	128971	402-9551604-7544318	05-31-22	Shipped	Amazon	
128973	128972	407-9547469-3152358	05-31-22	Shipped	Amazon	
128974	128973	402-6184140-0545956	05-31-22	Shipped	Amazon	
128975	128974	408-7436540-8728312	05-31-22	Shipped	Amazon	

	Sales Channel	ship-service-level	Category	Size	Courier
Status	...	\			
128971	Amazon.in	Expedited	Shirt	XL	
Shipped	...				
128972	Amazon.in	Expedited	T-shirt	M	
Shipped	...				
128973	Amazon.in	Expedited	Blazzer	XXL	
Shipped	...				
128974	Amazon.in	Expedited	T-shirt	XS	
Shipped	...				
128975	Amazon.in	Expedited	T-shirt	S	
Shipped	...				

	currency	Amount	ship-city	ship-state	ship-postal-code	\
128971	INR	517.0	HYDERABAD	TELANGANA	500013.0	
128972	INR	999.0	GURUGRAM	HARYANA	122004.0	
128973	INR	690.0	HYDERABAD	TELANGANA	500049.0	
128974	INR	1199.0	Halol	Gujarat	389350.0	
128975	INR	696.0	Raipur	CHHATTISGARH	492014.0	

	ship-country	B2B	fulfilled-by	New	PendingS
128971	IN	False	NaN	NaN	NaN
128972	IN	False	NaN	NaN	NaN
128973	IN	False	NaN	NaN	NaN
128974	IN	False	NaN	NaN	NaN
128975	IN	False	NaN	NaN	NaN

[5 rows x 21 columns]

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

#drop unrelated/blank columns

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

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

```

```

pd.isnull(df)
# checking null value

```

	index	Order ID	Date	Status	Fulfilment	Sales Channel \
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	False
...
128971	False	False	False	False	False	False
128972	False	False	False	False	False	False
128973	False	False	False	False	False	False
128974	False	False	False	False	False	False
128975	False	False	False	False	False	False

	ship-service-level	Category	Size	Courier	Status	Qty
0	False	False	False		False	False
1	False	False	False		False	False
2	False	False	False		False	False
3	False	False	False		False	False
4	False	False	False		False	False
...
...						
128971	False	False	False		False	False
128972	False	False	False		False	False
128973	False	False	False		False	False
128974	False	False	False		False	False
128975	False	False	False		False	False

	Amount	ship-city	ship-state	ship-postal-code	ship-country
0	False	False	False	False	False

1	False	False	False	False	False
False					
2	False	False	False	False	False
False					
3	False	False	False	False	False
False					
4	False	False	False	False	False
False					
...
...					
128971	False	False	False	False	False
False					
128972	False	False	False	False	False
False					
128973	False	False	False	False	False
False					
128974	False	False	False	False	False
False					
128975	False	False	False	False	False
False					

	fulfilled-by
0	False
1	False
2	True
3	False
4	True
...	...
128971	True
128972	True
128973	True
128974	True
128975	True

[128976 rows x 19 columns]

```
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
```

```
df.shape
```

```
(128976, 19)
```

```
#drop null values
```

```
df.dropna(inplace=True)
```

```
df.shape
```

```
(37514, 19)
```

```
df.columns
```

```
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')
```

```
# change data type
```

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

```
#checking whether the data type change or not
```

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

```
dtype('int32')
```

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

```
df.columns
```

```
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')
```

```
#rename Columns
```

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

	index	Order ID	Date	
Status \				
0	0	405-8078784-5731545	2022-04-30	
Cancelled				
1	1	171-9198151-1101146	2022-04-30	Shipped - Delivered to Buyer
3	3	403-9615377-8133951	2022-04-30	
Cancelled				
7	7	406-7807733-3785945	2022-04-30	Shipped - Delivered to Buyer
12	12	405-5513694-8146768	2022-04-30	Shipped - Delivered to Buyer
...	
...				
128875	128874	405-4724097-1016369	2022-06-01	Shipped - Delivered to Buyer
128876	128875	403-9524128-9243508	2022-06-01	
Cancelled				
128888	128887	405-6493630-8542756	2022-05-31	Shipped - Delivered to Buyer
128891	128890	407-0116398-1810752	2022-05-31	
Cancelled				
128892	128891	403-0317423-9322704	2022-05-31	Shipped - Delivered to Buyer

	Fulfilment	Sales Channel	ship-service-level	Category	Size \
0	Merchant	Amazon.in	Standard	T-shirt	S
1	Merchant	Amazon.in	Standard	Shirt	3XL
3	Merchant	Amazon.in	Standard	Blazzer	L
7	Merchant	Amazon.in	Standard	Shirt	S
12	Merchant	Amazon.in	Standard	Shirt	XS
...
128875	Merchant	Amazon.in	Standard	T-shirt	S
128876	Merchant	Amazon.in	Standard	Blazzer	XL
128888	Merchant	Amazon.in	Standard	Trousers	M
128891	Merchant	Amazon.in	Standard	Wallet	Free
128892	Merchant	Amazon.in	Standard	Blazzer	M

	Courier Status	Quantity	currency	Amount	ship-city \
0	On the Way	0	INR	647.62	MUMBAI
1	Shipped	1	INR	406.00	BENGALURU
3	On the Way	0	INR	753.33	PUDUCHERRY
7	Shipped	1	INR	399.00	HYDERABAD
12	Shipped	1	INR	399.00	Amravati.
...
128875	Shipped	1	INR	854.00	ALLUR
128876	On the Way	0	INR	734.29	Barabanki

128888	Shipped	1	INR	518.00		NOIDA
128891	On the Way	0	INR	398.10		MADURAI
128892	Shipped	1	INR	721.00	UTTAR	BAGDOGRA

	ship-state	ship-postal-code	ship-country	B2B
fulfilled-by				
0	MAHARASHTRA	400081	IN	False Easy
Ship				
1	KARNATAKA	560085	IN	False Easy
Ship				
3	PUDUCHERRY	605008	IN	False Easy
Ship				
7	TELANGANA	500032	IN	False Easy
Ship				
12	MAHARASHTRA	444606	IN	False Easy
Ship				
...
...				
128875	ANDHRA PRADESH	524315	IN	False Easy
Ship				
128876	UTTAR PRADESH	225001	IN	False Easy
Ship				
128888	UTTAR PRADESH	201301	IN	False Easy
Ship				
128891	TAMIL NADU	625007	IN	False Easy
Ship				
128892	WEST BENGAL	734014	IN	False Easy
Ship				

[37514 rows x 19 columns]

#describe() method return description of the data in the DataFrame(i.e count,mean,std,min..etc)
df.describe()

	index	Date	Qty \
count	37514.000000	37514	37514.000000
mean	60953.809858	2022-05-11 07:56:47.303939840	0.867383
min	0.000000	2022-03-31 00:00:00	0.000000
25%	27235.250000	2022-04-20 00:00:00	1.000000
50%	63470.500000	2022-05-09 00:00:00	1.000000
75%	91790.750000	2022-06-01 00:00:00	1.000000
max	128891.000000	2022-06-29 00:00:00	5.000000
std	36844.853039	NaN	0.354160

	Amount	ship-postal-code
count	37514.000000	37514.000000
mean	646.553960	463291.552754
min	0.000000	110001.000000
25%	458.000000	370465.000000

50%	629.000000	500019.000000
75%	771.000000	600042.000000
max	5495.000000	989898.000000
std	279.952414	194550.425637

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

	Order ID	Status
Fulfilment \		
count	37514	37514
unique	34664	11
top	171-5057375-2831560	Shipped - Delivered to Buyer
freq	12	28741

	Sales Channel	ship-service-level	Category	Size	Courier	Status
\						
count	37514	37514	37514	37514		37514
unique	1	1	8	11		3
top	Amazon.in	Standard	T-shirt	M		Shipped
freq	37514	37514	14062	6892		31859

	currency	ship-city	ship-state	ship-country	fulfilled-by
count	37514	37514	37514	37514	37514
unique	1	4698	58	1	1
top	INR	BENGALURU	MAHARASHTRA	IN	Easy Ship
freq	37514	2839	6236	37514	37514

#use describe() for specific columns

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

	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

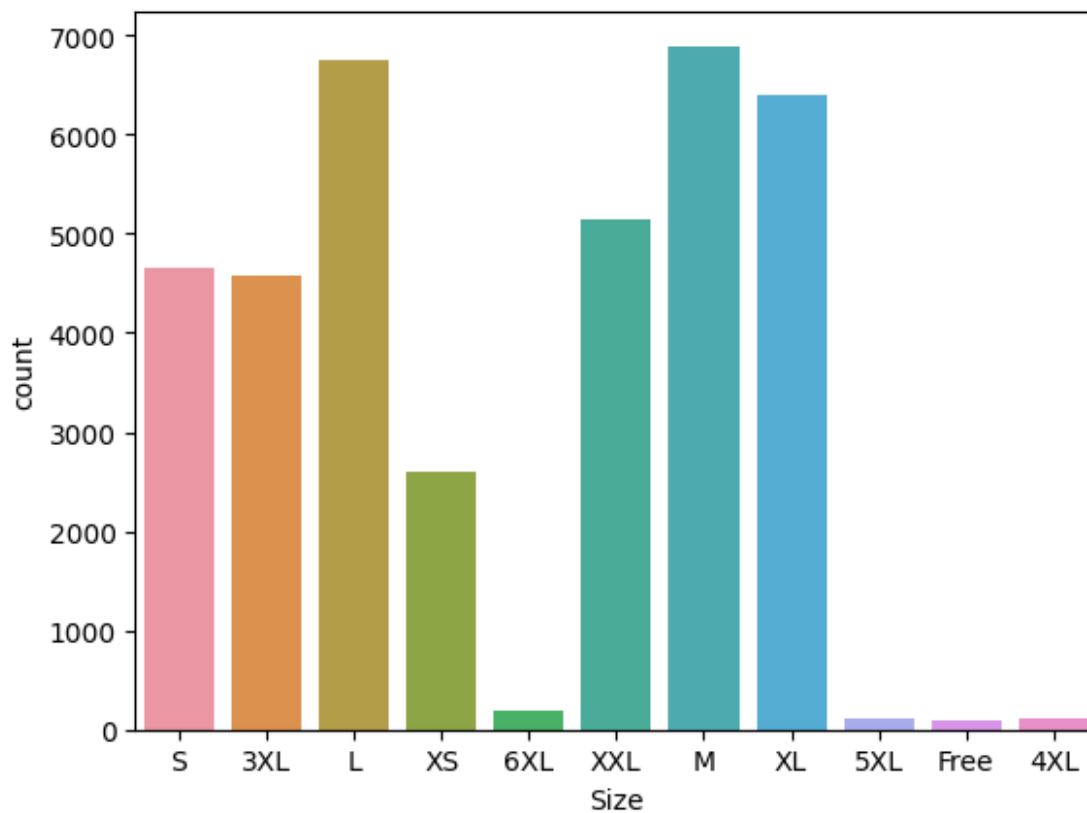
Exploratory Data Analysis

```
df.columns
```

```
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')
```

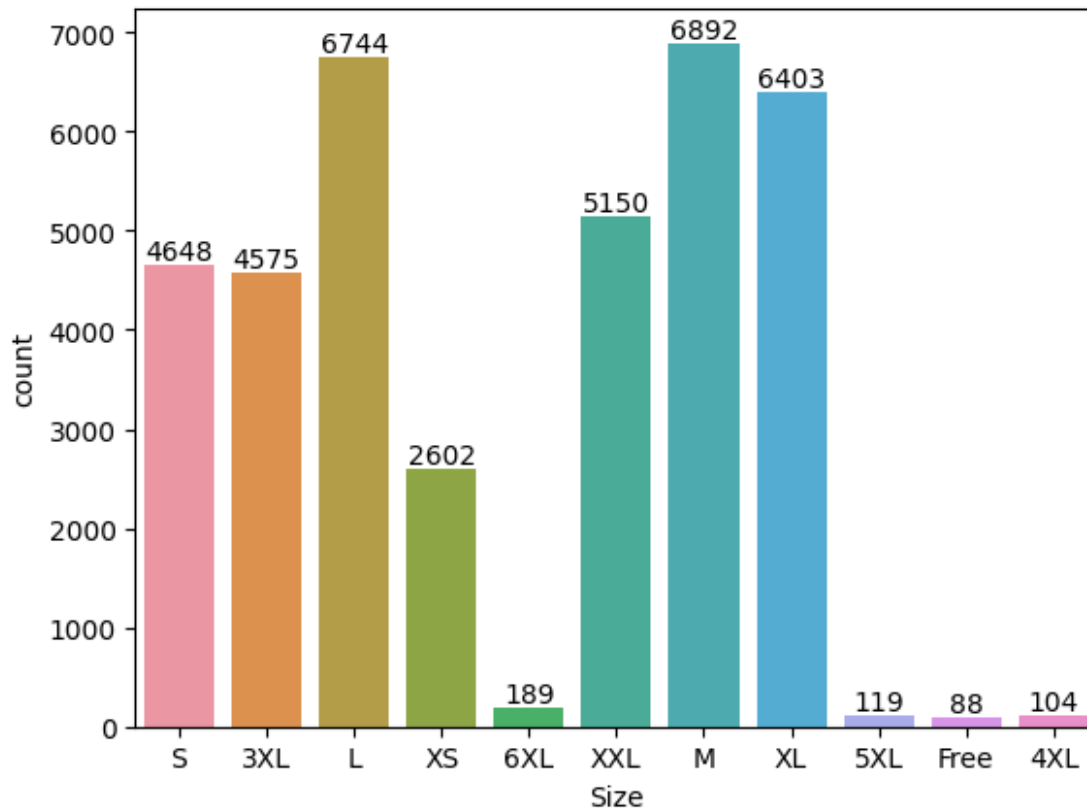
size

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



```
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

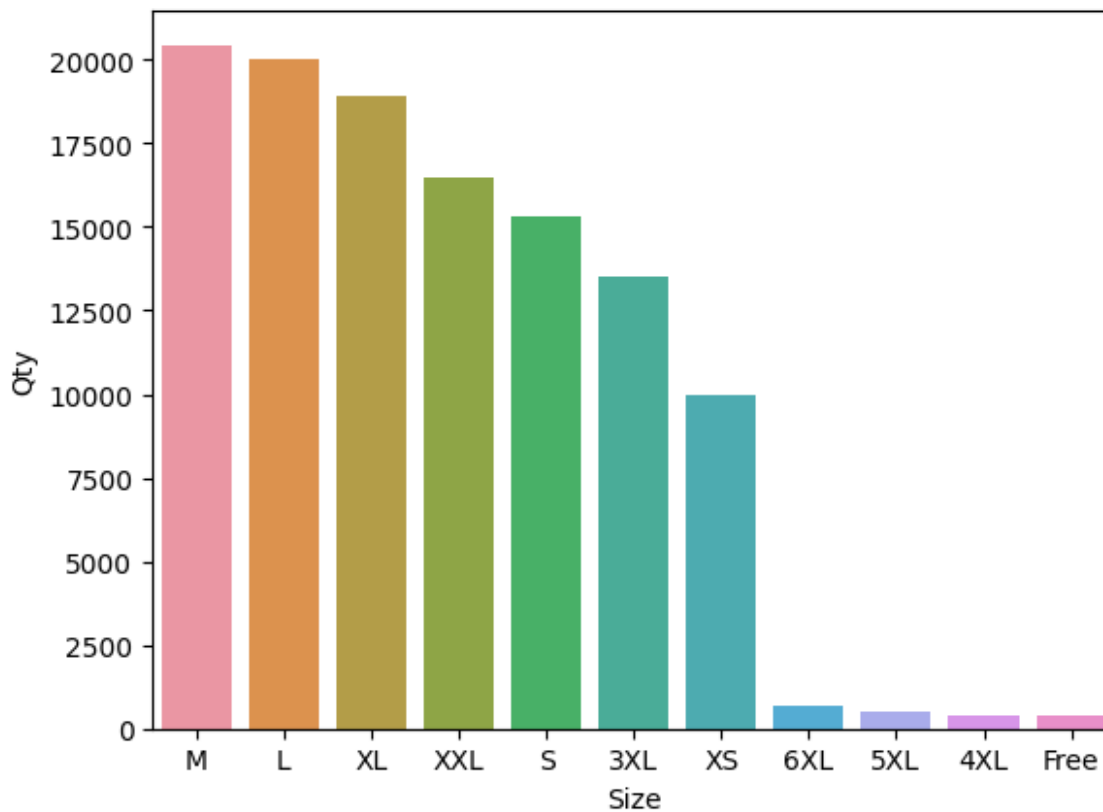
The `groupby()` function in pandas is used to group data based on one or more columns in a DataFrame

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

	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

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

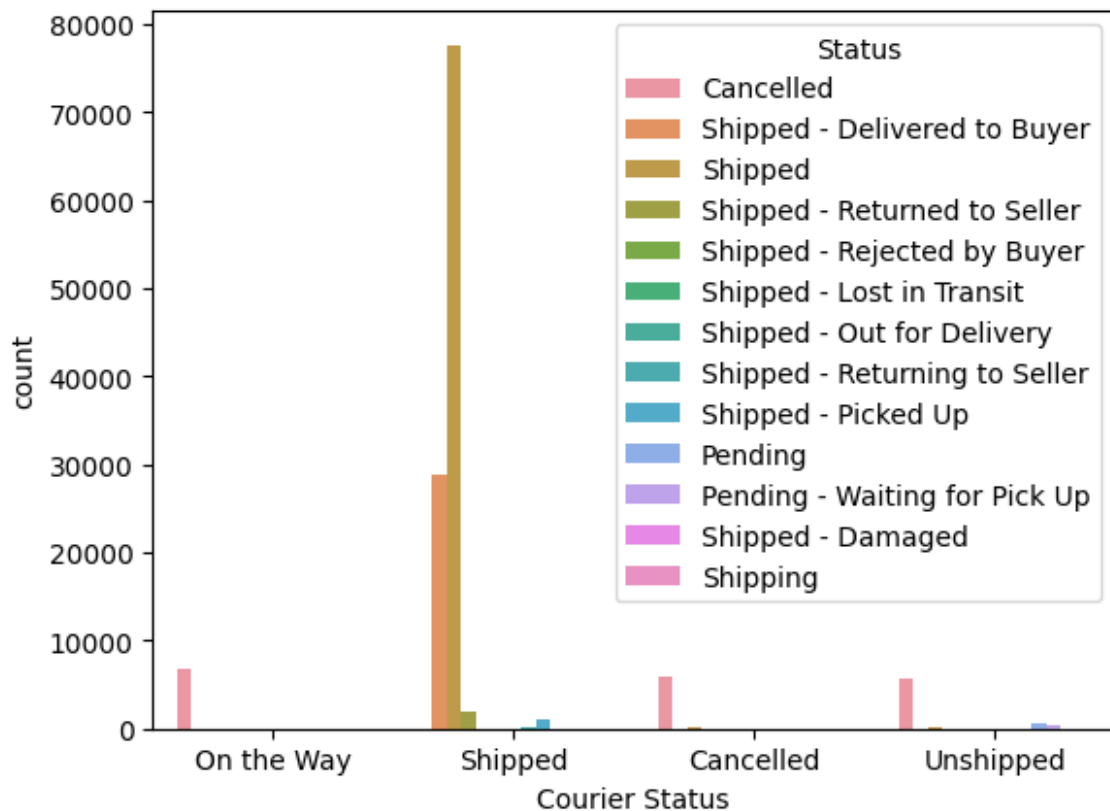
sns.barplot(x='Size',y='Qty', data=S_Qty)
<Axes: xlabel='Size', ylabel='Qty'>
```



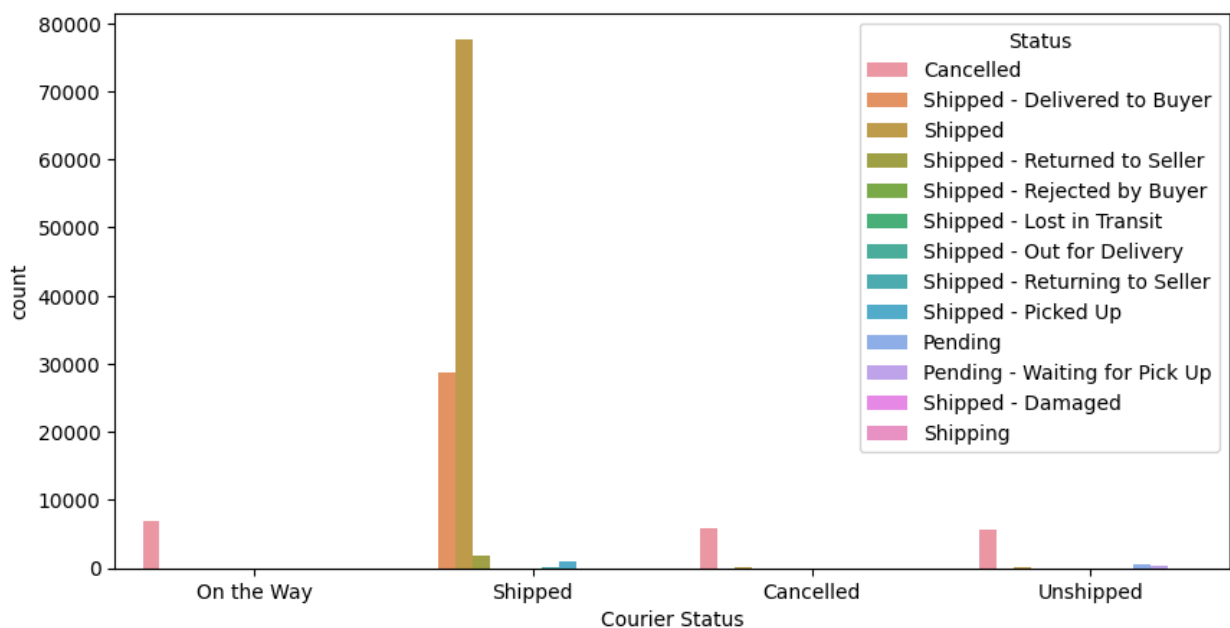
Note: From above Graph you can see that most of the Qty buys M-Size in the sales

Courier Status

```
sns.countplot(data=df, x='Courier Status',hue= 'Status')
<Axes: xlabel='Courier Status', ylabel='count'>
```



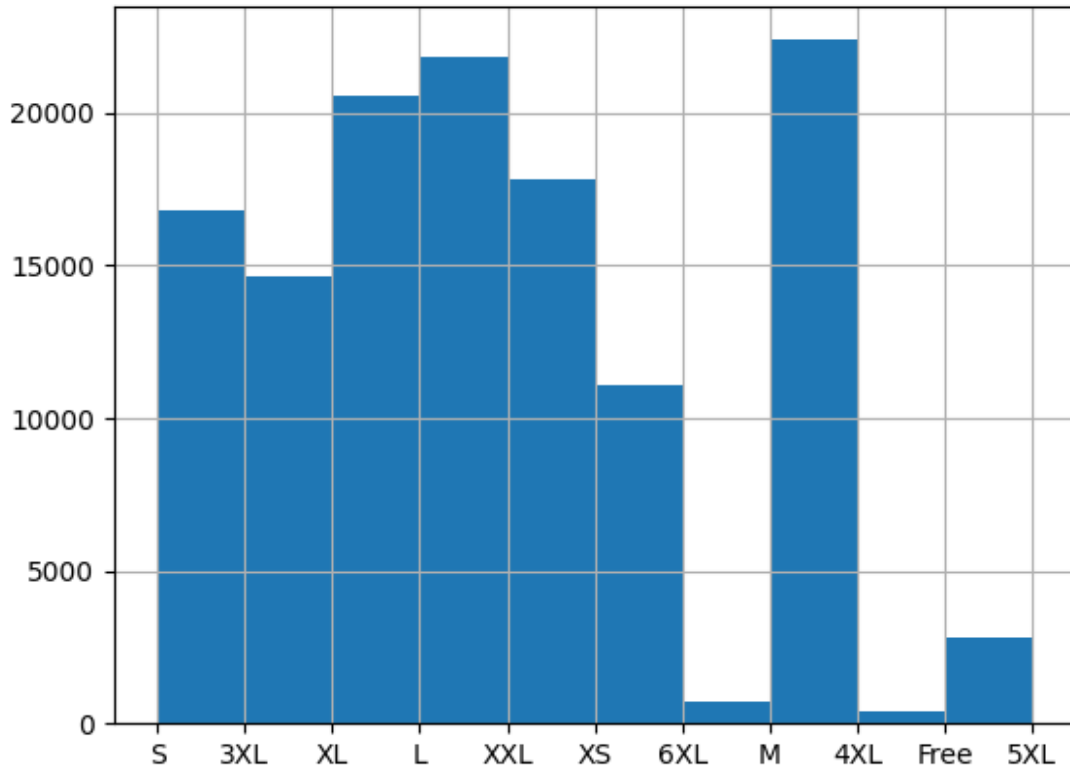
```
plt.figure(figsize=(10,5))
ax=sns.countplot(data=df, x='Courier Status',hue= 'Status')
plt.show()
```



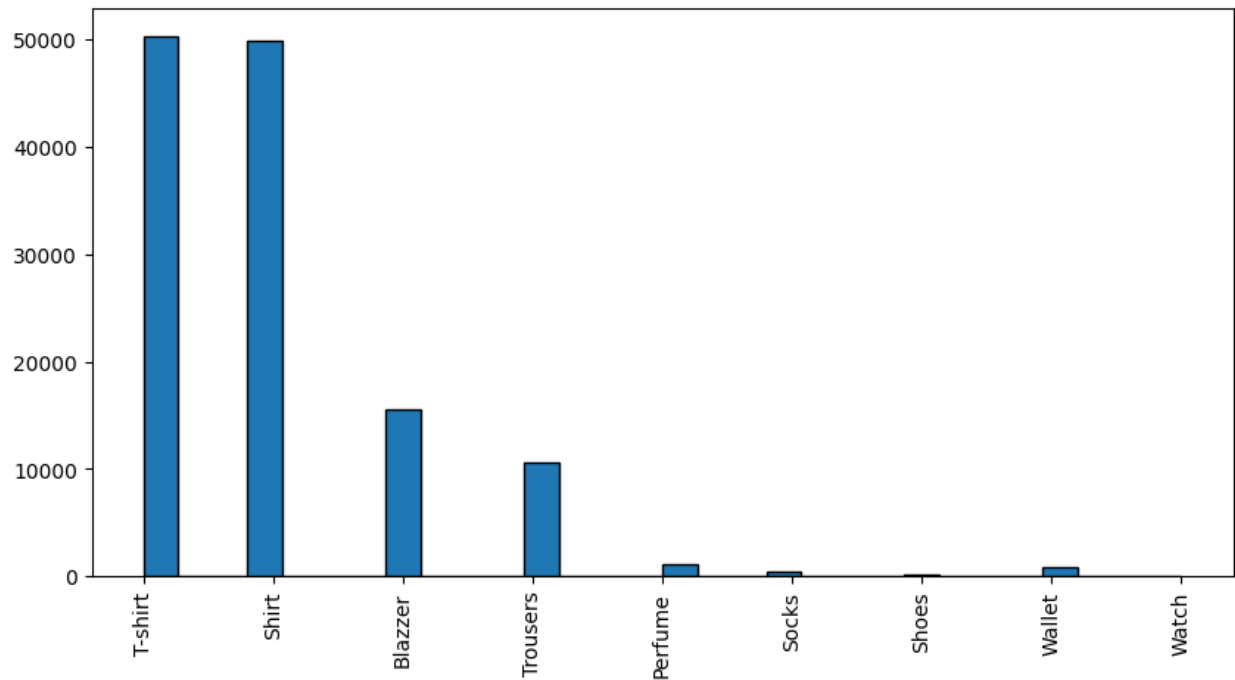
Note: From above Graph the majority of the orders are shipped through the courier.

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

<Axes: >



```
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

```
# Checking B2B Data by using pie chart
```

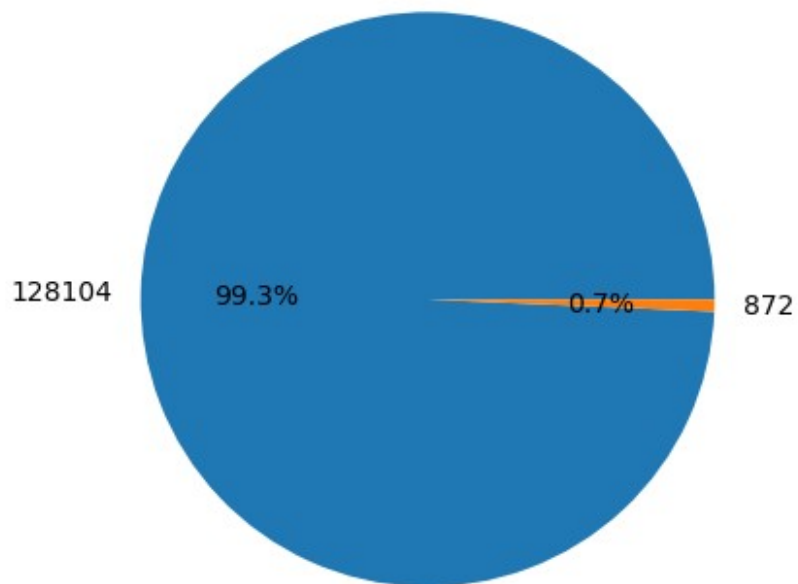
```
B2B_Check = df['B2B'].value_counts()
```

```
# Plot the pie chart
```

```
plt.pie(B2B_Check, labels=B2B_Check, autopct='%1.1f%%')
```

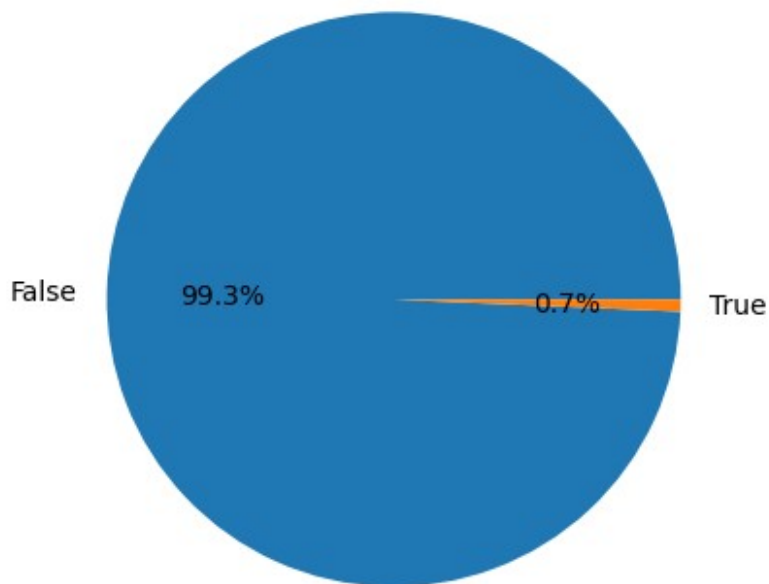
```
#plt.axis('equal')
```

```
plt.show()
```



```
# 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()
```

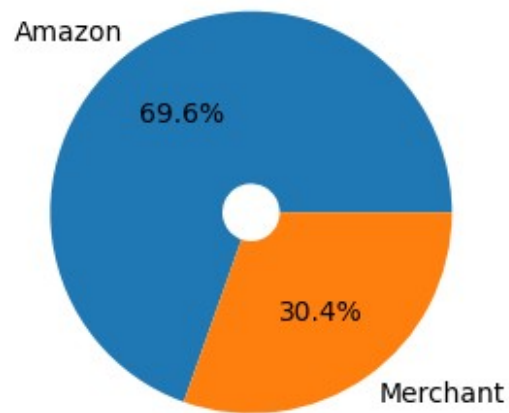
Note : From above chart we can see that maximum i.e. 99.3% of buyers are retailers and 0.7% are B2B buyers

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

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

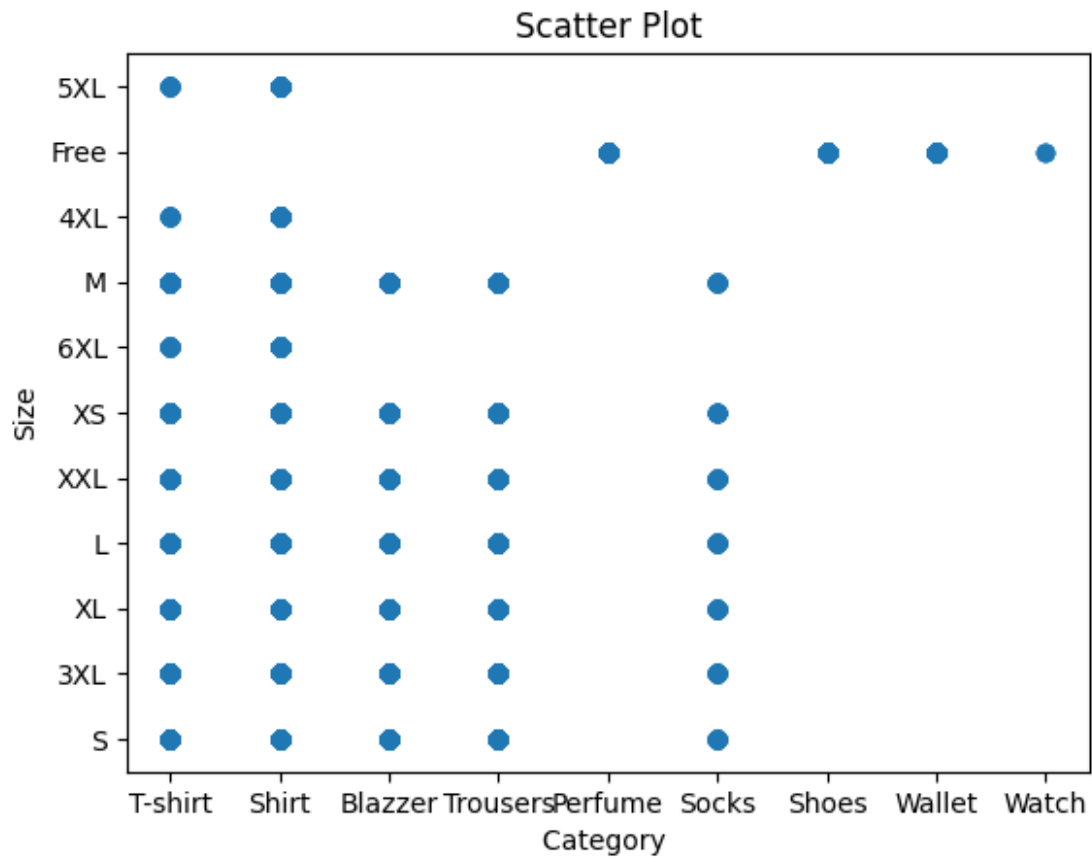
ax.pie(a1, labels=a1.index, autopct='%1.1f%%', radius=0.7,
       wedgeprops=dict(width=0.6))
ax.set(aspect="equal")

plt.show()
```

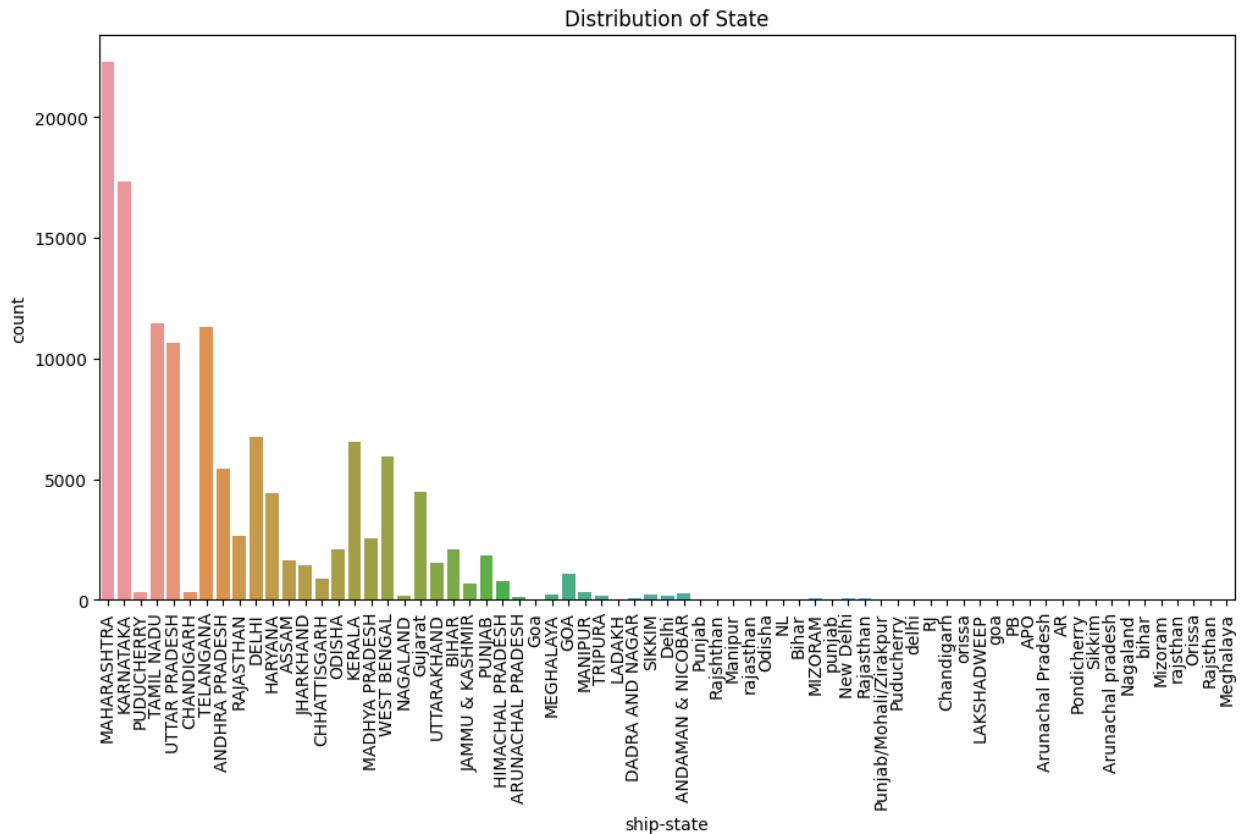


Note: From above chart you can see that most of the Fulfilment are amazon

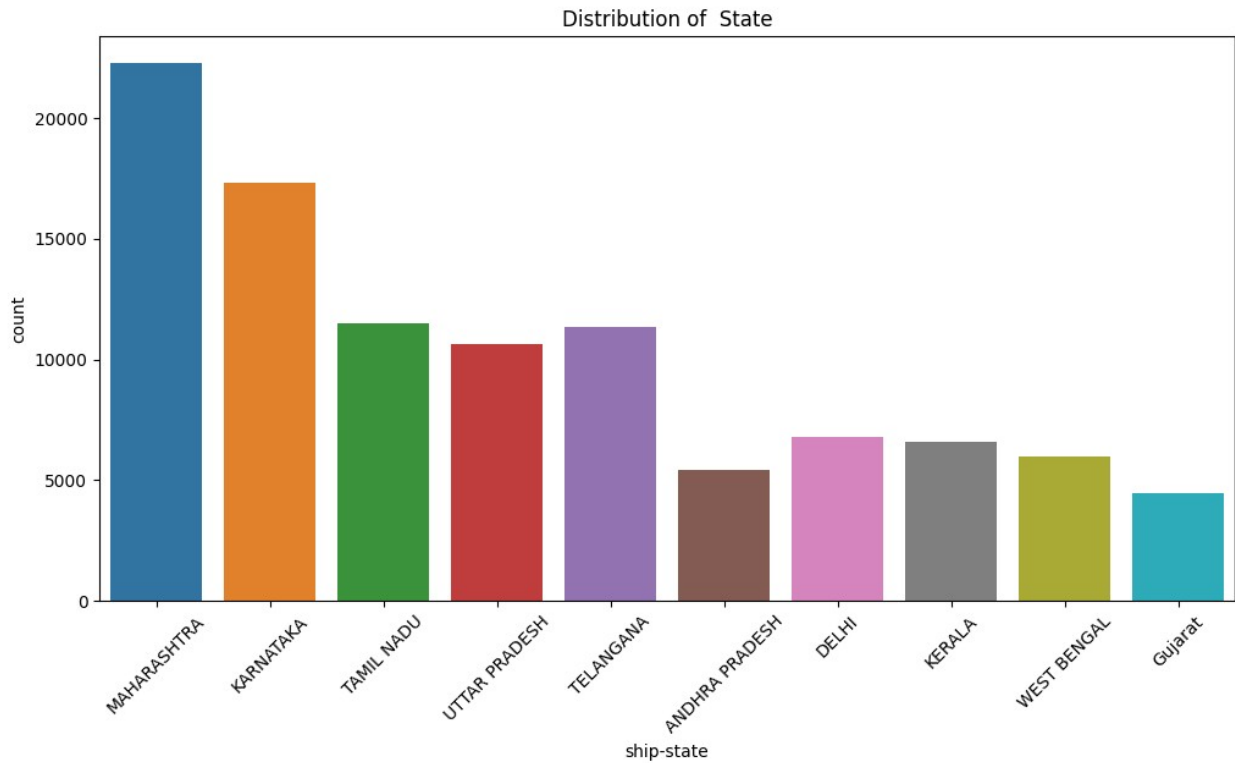
```
# Prepare data for scatter plot  
x_data = df['Category']  
y_data = df['Size']  
  
# Plot the scatter plot  
plt.scatter(x_data, y_data)  
plt.xlabel('Category')  
plt.ylabel('Size')  
plt.title('Scatter Plot')  
plt.show()
```



```
# 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()
```



```
# 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.