

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

```
In [2]: df=pd.read_csv('Amazon Sale Report.csv',encoding= 'unicode_escape')
```

```
In [3]: df.shape
```

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

```
In [5]: df.head(10)
```

```
Out[5]:
```

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	...	currency	A
0	0	405-8078784-5731545	04-30-22	Cancelled	Merchant	Amazon.in	Standard	T-shirt	S	On the Way	...	INR	
1	1	171-9198151-1101146	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	3XL	Shipped	...	INR	
2	2	404-0687676-7273146	04-30-22	Shipped	Amazon	Amazon.in	Expedited	Shirt	XL	Shipped	...	INR	
3	3	403-9615377-8133951	04-30-22	Cancelled	Merchant	Amazon.in	Standard	Blazzer	L	On the Way	...	INR	
4	4	407-1069790-7240320	04-30-22	Shipped	Amazon	Amazon.in	Expedited	Trousers	3XL	Shipped	...	INR	
5	5	404-1490984-4578765	04-30-22	Shipped	Amazon	Amazon.in	Expedited	T-shirt	XL	Shipped	...	INR	
6	6	408-5748499-6859555	04-30-22	Shipped	Amazon	Amazon.in	Expedited	T-shirt	L	Shipped	...	INR	
7	7	406-7807733-3785945	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	S	Shipped	...	INR	
8	8	407-5443024-5233168	04-30-22	Cancelled	Amazon	Amazon.in	Expedited	T-shirt	3XL	Cancelled	...	NaN	
9	9	402-4393761-0311520	04-30-22	Shipped	Amazon	Amazon.in	Expedited	Shirt	XXL	Shipped	...	INR	

10 rows × 21 columns

```
In [6]: df.tail()
```

Out[6]:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	...	currency
128971	128970	406-6001380-7673107	05-31-22	Shipped	Amazon	Amazon.in	Expedited	Shirt	XL	Shipped	...	INR
128972	128971	402-9551604-7544318	05-31-22	Shipped	Amazon	Amazon.in	Expedited	T-shirt	M	Shipped	...	INR
128973	128972	407-9547469-3152358	05-31-22	Shipped	Amazon	Amazon.in	Expedited	Blazzer	XXL	Shipped	...	INR
128974	128973	402-6184140-0545956	05-31-22	Shipped	Amazon	Amazon.in	Expedited	T-shirt	XS	Shipped	...	INR
128975	128974	408-7436540-8728312	05-31-22	Shipped	Amazon	Amazon.in	Expedited	T-shirt	S	Shipped	...	INR

5 rows × 21 columns

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

In [9]: `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 [10]: pd.isnull(df)
# checking null value
```

```
Out[10]:
```

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	Qty	currency	A
0	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	
2	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	
4	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	
128972	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	
128974	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	

128976 rows × 19 columns

```
In [11]: pd.isnull(df).sum()
# sum will give total values of null values
```

```
Out[11]: 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 [12]: df.shape
```

```
Out[12]: (128976, 19)
```

```
In [13]: #drop null values
df.dropna(inplace=True)
```

```
In [14]: df.shape
```

```
Out[14]: (37514, 19)
```

```
In [16]: df.columns
```

```
Out[16]: 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 [17]: # change data type
df['ship-postal-code']=df['ship-postal-code'].astype('int')
```

```
In [18]: #checking whether the data type change or not
df['ship-postal-code'].dtype
```

```
Out[18]: dtype('int32')
```

```
In [19]: df['Date']=pd.to_datetime (df['Date'])
```

```
In [20]: df.columns
```

```
Out[20]: 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 [21]: #rename Columns
df.rename(columns={'Qty':'Quantity'})
```

Out[21]:

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

37514 rows × 19 columns

In [22]: `#describe() method return description of the data in the DataFrame(i.e count,mean,std,min)`  
`df.describe()`

Out[22]:

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

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

Out[23]:

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

In [24]:

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

Out[24]:

	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 [25]:

```
df.columns
```

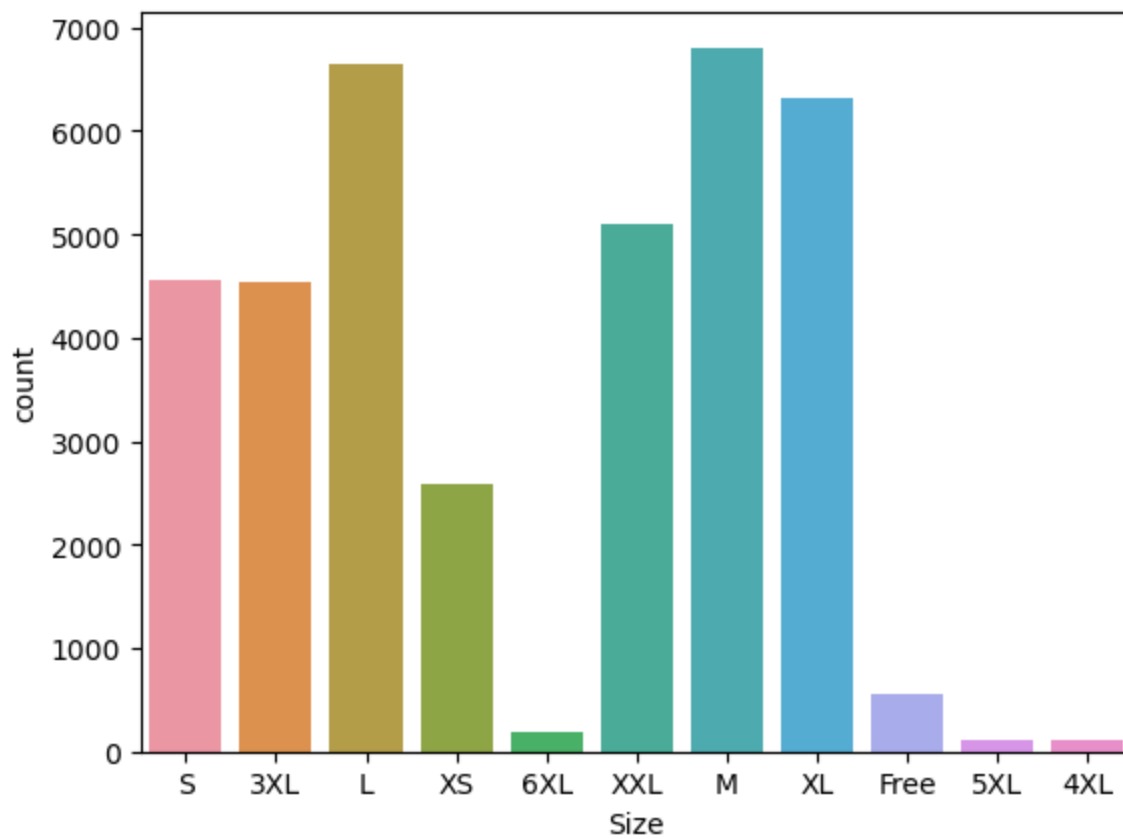
Out[25]:

```
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')
```

## Size

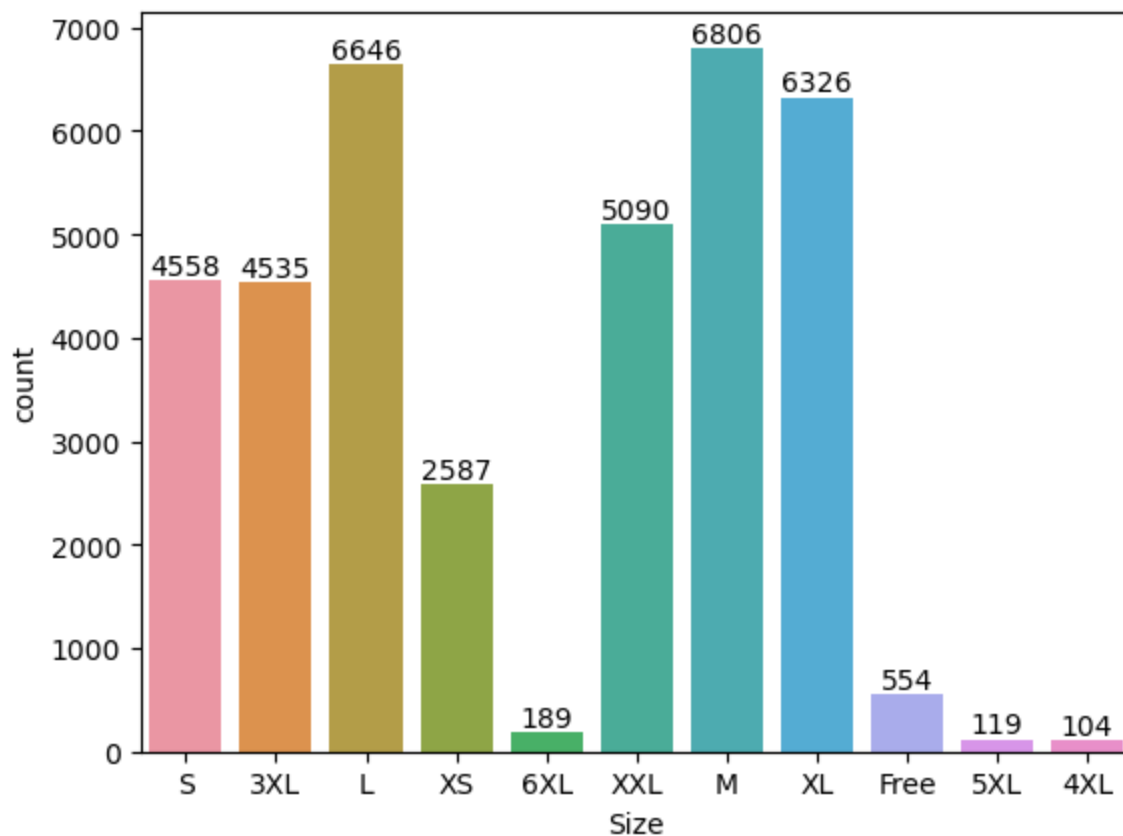
In [26]:

```
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)
```



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

# Group By

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

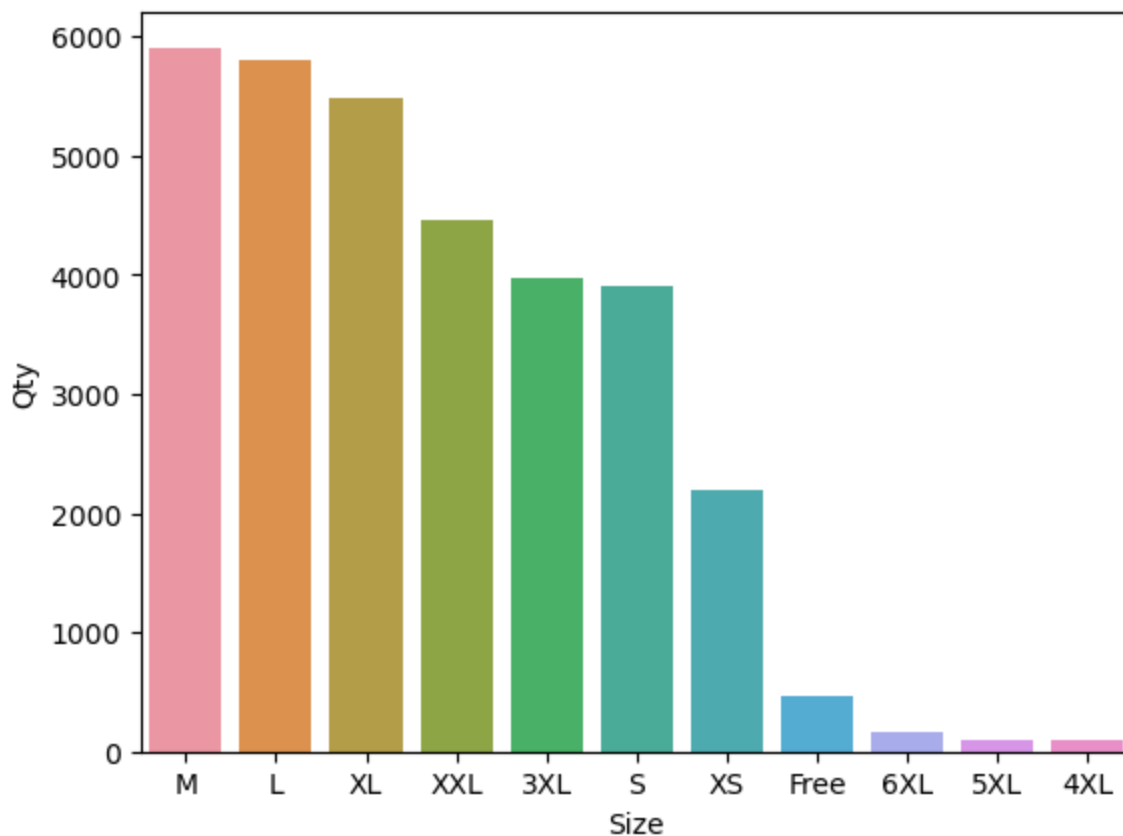
```
In [28]: df.groupby(['Size'], as_index=False)['Qty'].sum().sort_values(by='Qty', ascending=False)
```

```
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 [29]: 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[29]: <Axes: xlabel='Size', ylabel='Qty'>
```



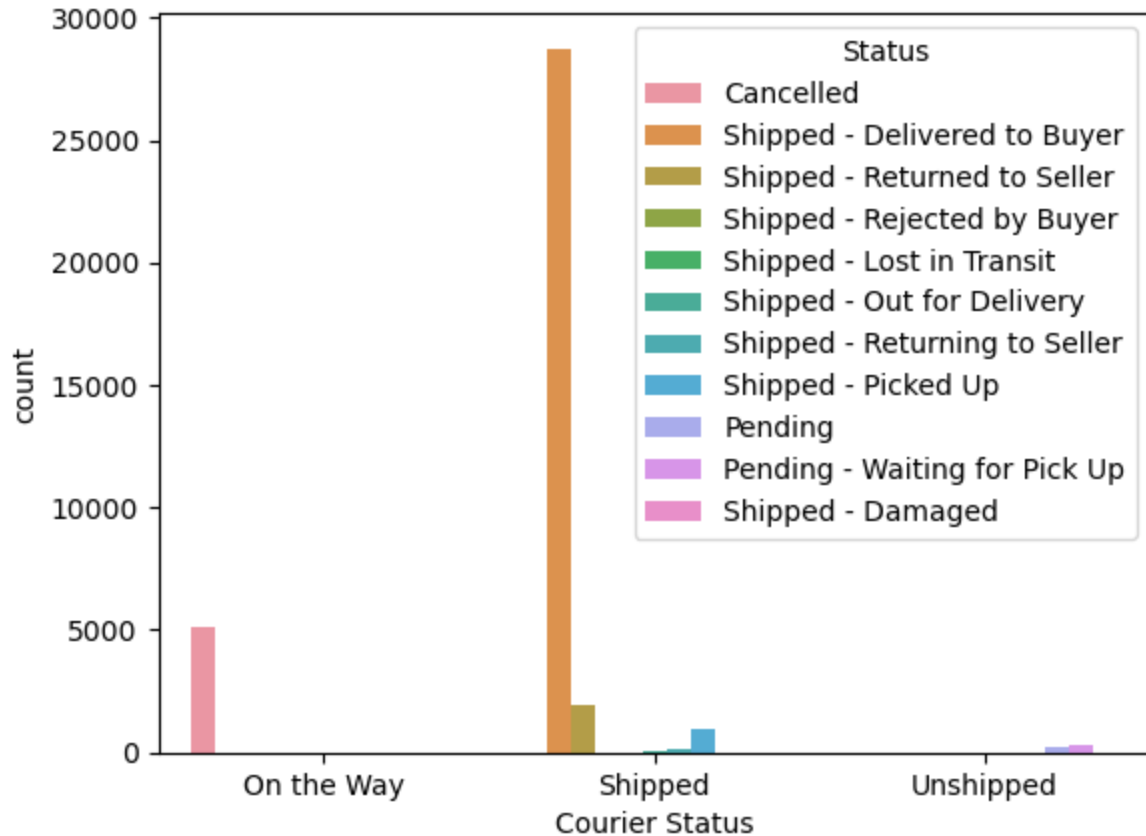


In [ ]: Note: From above Graph you can see that most of the Qty buys M-Size in the sales

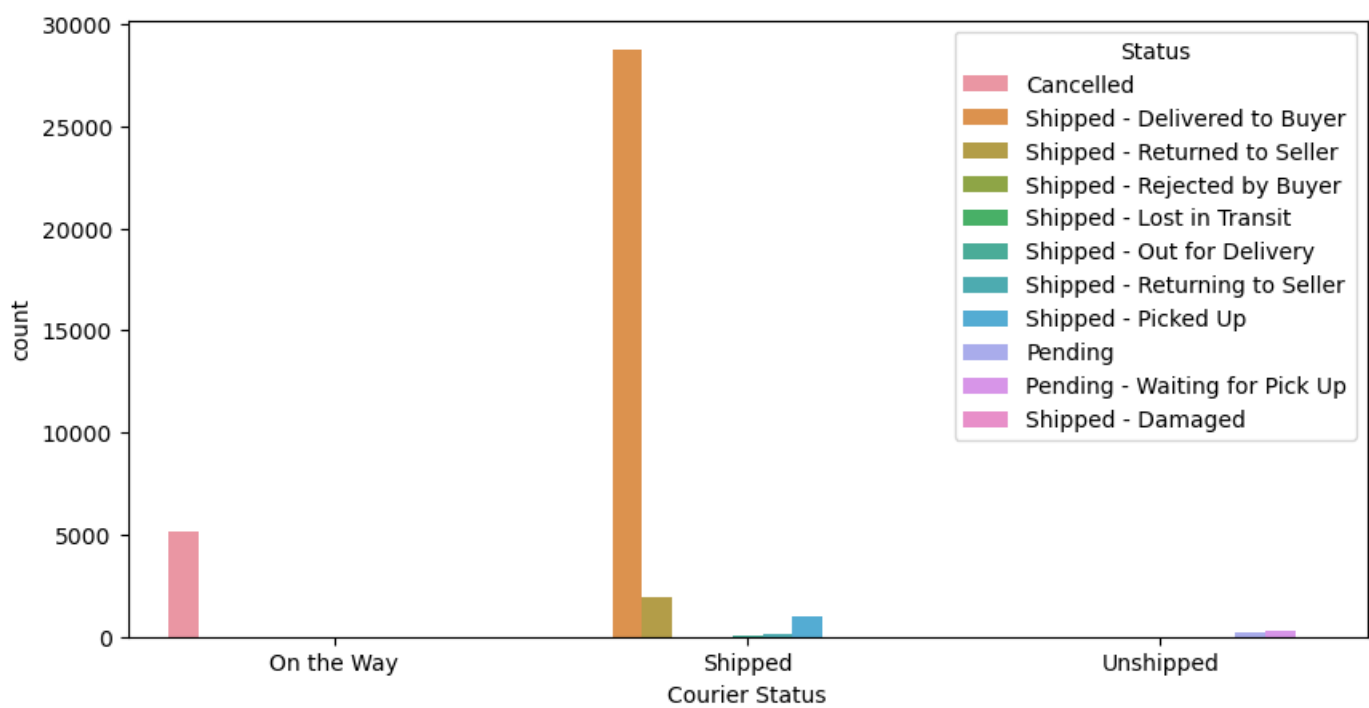
## Courier Status

```
In [30]: sns.countplot(data=df, x='Courier Status', hue= 'Status')
```

```
Out[30]: <Axes: xlabel='Courier Status', ylabel='count'>
```



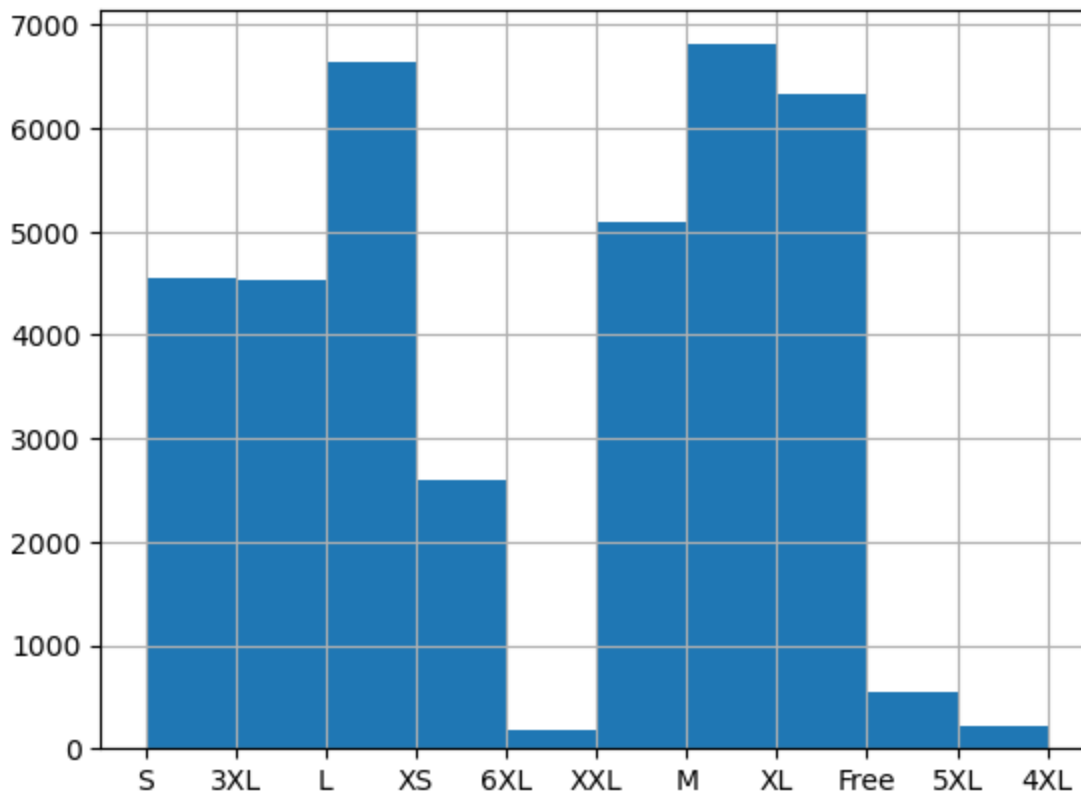
```
In [31]: plt.figure(figsize=(10,5))  
ax=sns.countplot(data=df, x='Courier Status', hue= 'Status')  
plt.show()
```



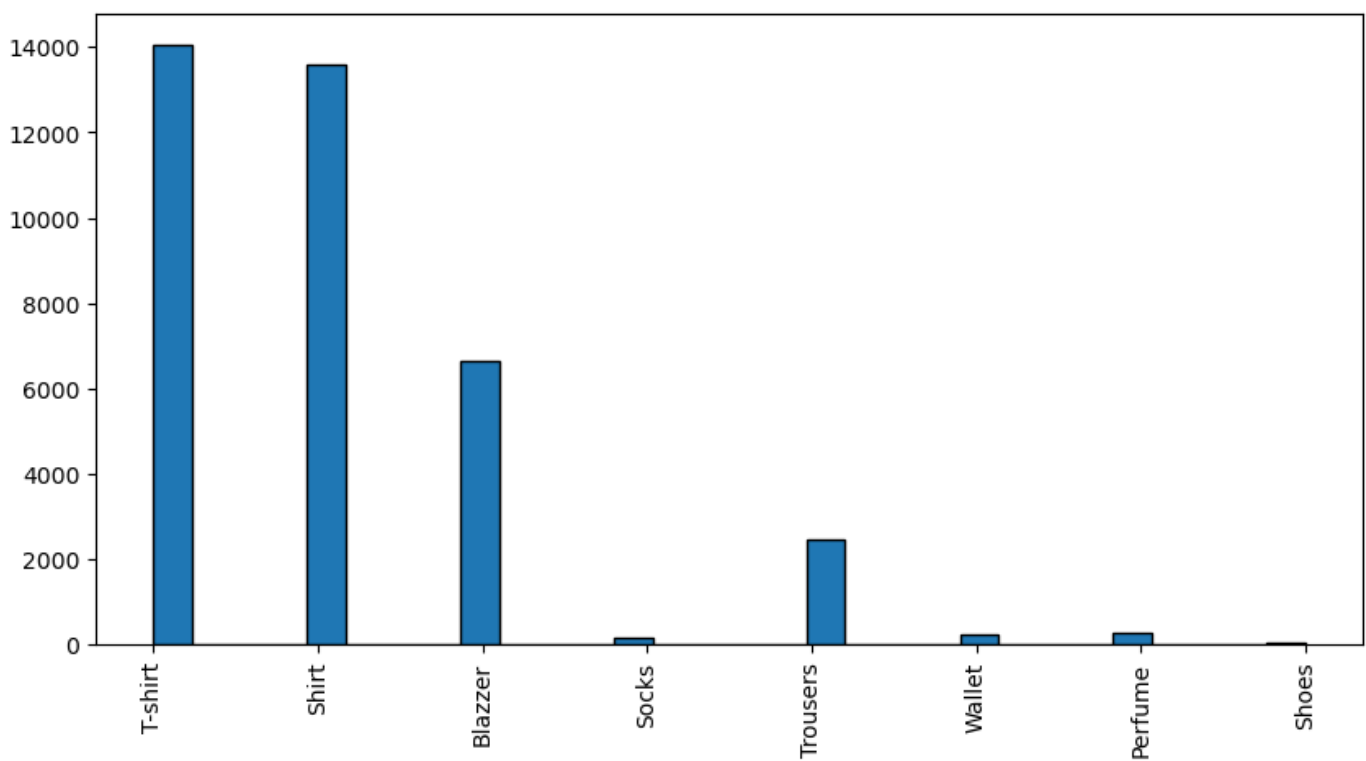
In [ ]: Note: From above Graph the majority of the orders are shipped through the courier.

In [32]: `#histogram`  
`df['Size'].hist()`

Out[32]: <Axes: >



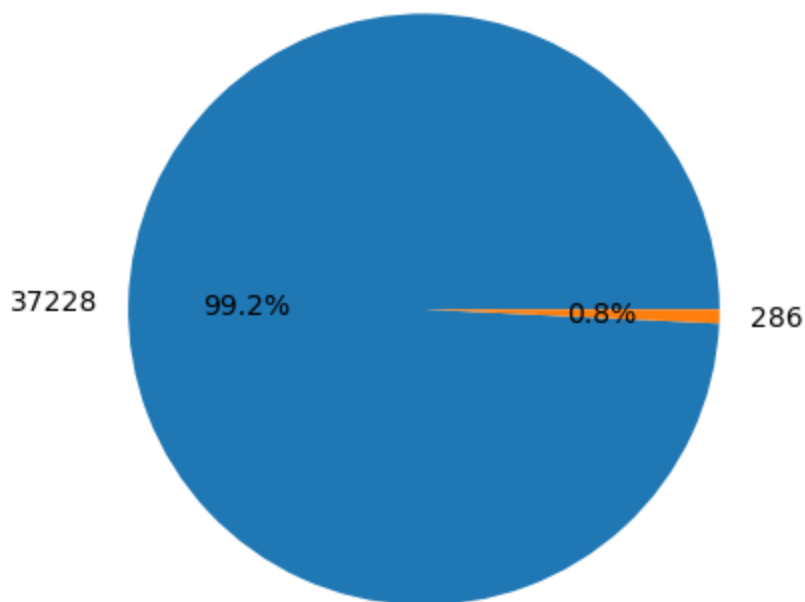
In [33]: `df['Category'] = df['Category'].astype(str)`  
`column_data = df['Category']`  
`plt.figure(figsize=(10, 5))`  
`plt.hist(column_data, bins=30, edgecolor='Black')`  
`plt.xticks(rotation=90)`  
`plt.show()`



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

```
In [34]: # Checking B2B Data by using pie chart
B2B_Check = df['B2B'].value_counts()

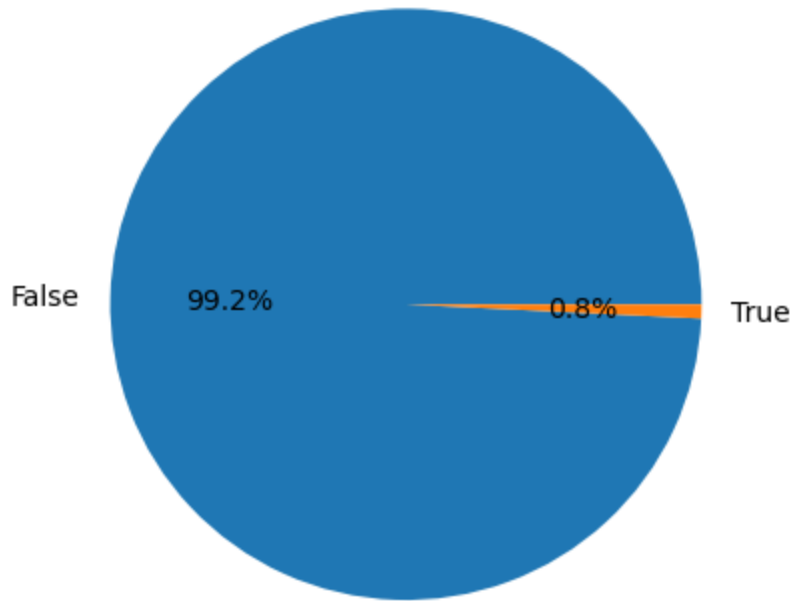
# Plot the pie chart
plt.pie(B2B_Check, labels=B2B_Check, autopct='%1.1f%%')
#plt.axis('equal')
plt.show()
```



```
In [35]: # Checking B2B Data by using pie chart
B2B_Check = df['B2B'].value_counts()
```

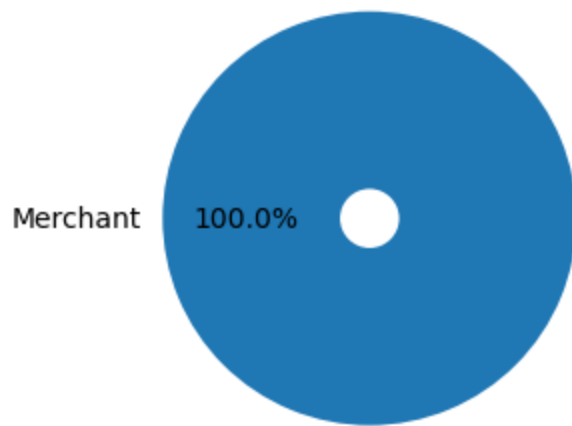
```
# Plot the pie chart
```

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



In [ ]: Note : From above chart we can see that maximum i.e. 99.3% of buyers are retailers and 0

```
In [36]: # 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()
```



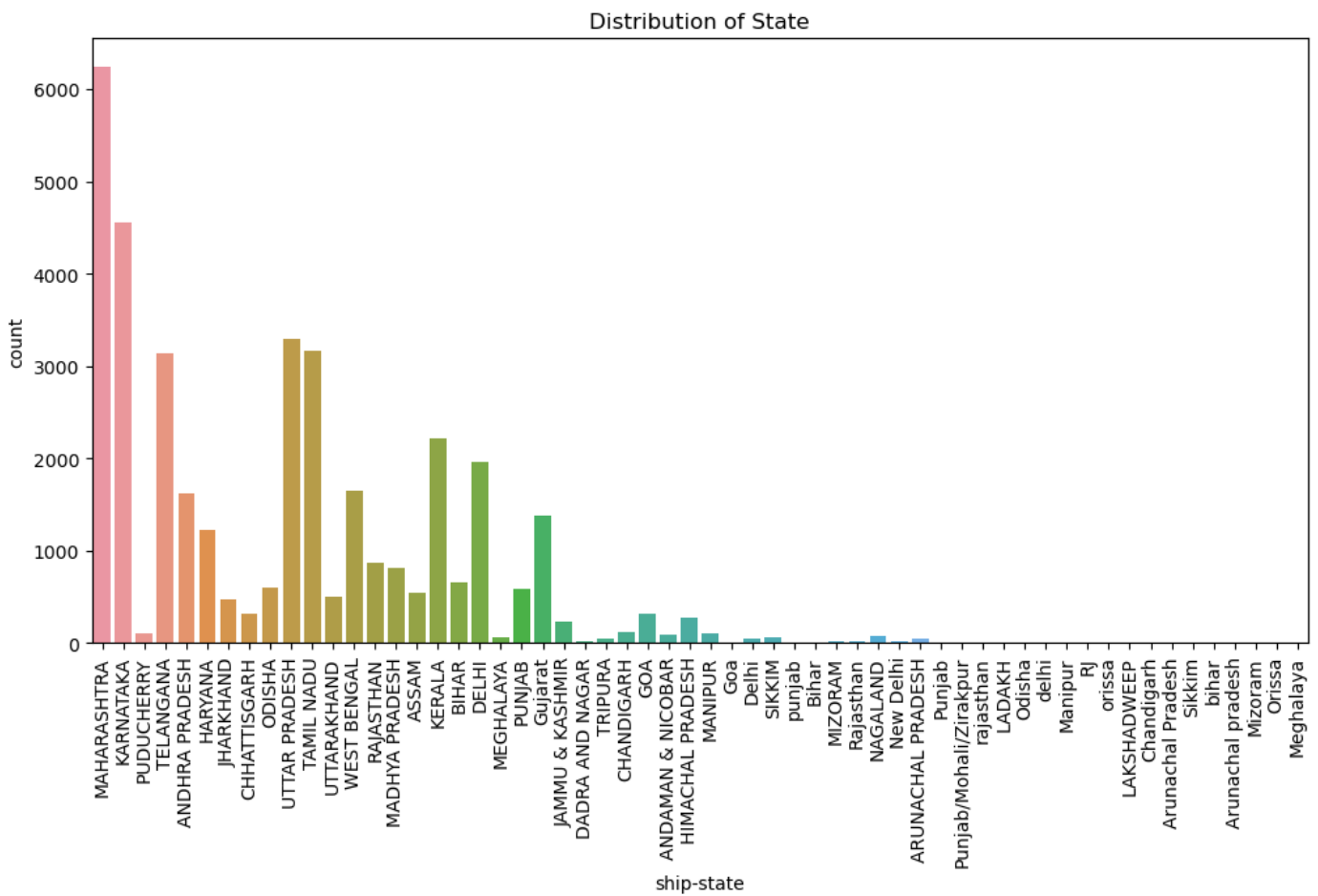
In [ ]: Note: From above chart you can see that most of the Fulfilment are amazon

```
In [37]: # Prepare data for scatter plot
x_data = df['Category']
y_data = df['Size']

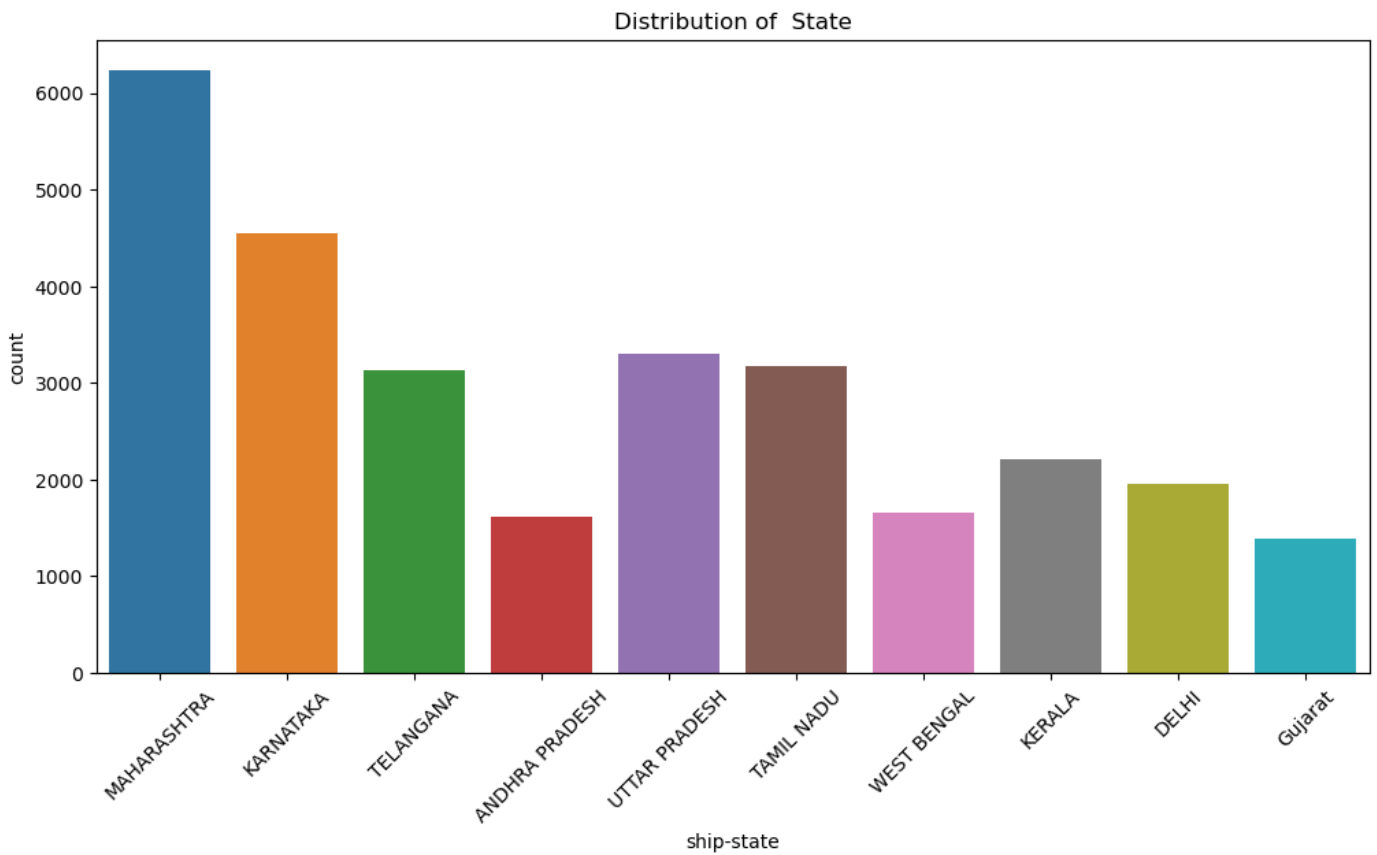
# Plot the scatter plot
plt.scatter(x_data, y_data)
plt.xlabel('Category ')
plt.ylabel('Size')
plt.title('Scatter Plot')
plt.show()
```



```
In [38]: # 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 [39]: # 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()
```



In [ ]: Note: From above Graph you can see that most of the buyers are Maharashtra state

## Conclusion

In [ ]: The data analysis reveals that the business has a significant customer base in Maharasht