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

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

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

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

In [4]:
```
df.head(5)
```

Out[4]:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	C s
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	Sh
2	2	404-0687676-7273146	04-30-22	Shipped	Amazon	Amazon.in	Expedited	Shirt	XL	Sh
3	3	403-9615377-8133951	04-30-22	Cancelled	Merchant	Amazon.in	Standard	Blazzer	L	(
		407-	04-							

In [78]:
```
df.tail(5)
```

Out[78]:

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

5 rows × 21 columns

In [79]:
```
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 [80]:
```
# Block Unrelated/Blank Columns

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

In [81]:
```
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 [82]:
```
# Check the Null values

pd.isnull(df)
```

Out[82]:

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

128976 rows × 19 columns

In [83]:
```
# Sum wil give total values of null values

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

Out[83]:
```
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 [84]:
```
df.shape
```

Out[84]:
```
(128976, 19)
```

In [85]:
```
df.columns
```

Out[85]:
```
Index(['index', 'Order ID', 'Date', 'Status', 'Fulfilment', 'Sales Chan
nel',
       'ship-service-level', 'Category', 'Size', 'Courier Status', 'Qt
y',
       'currency', 'Amount', 'ship-city', 'ship-state', 'ship-postal-co
de',
       'ship-country', 'B2B', 'fulfilled-by'],
      dtype='object')
```

In [86]:
```
# Change Data Type
df[ 'ship-postal-code' ]=df[ 'ship-postal-code' ].astype('float')
```

In [87]:
```
df['ship-postal-code'].dtype
```

Out[87]:
```
dtype('float64')
```

In [88]:
```
df['Date']=pd.to_datetime(df[ 'Date' ])
```

```
<ipython-input-88-d300efe89e92>:1: UserWarning: Could 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 fo
rmat.
  df['Date']=pd.to_datetime(df[ 'Date' ])
```

In [89]:
```
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  datetime64[ns]
 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), datetime64[ns](1), float64(2), int64(2), object(13)
memory usage: 17.8+ MB
```

In [90]:
```
df.columns
```

Out[90]:
```
Index(['index', 'Order ID', 'Date', 'Status', 'Fulfilment', 'Sales Chan
nel',
       'ship-service-level', 'Category', 'Size', 'Courier Status', 'Qt
y',
       'currency', 'Amount', 'ship-city', 'ship-state', 'ship-postal-co
de',
       'ship-country', 'B2B', 'fulfilled-by'],
      dtype='object')
```

In [91]:
```
# Rename Columns
df.rename(columns={'Qty':'Quantity'})
```

Out[91]:

Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	Quantity	currency
)22-4-30	Cancelled	Merchant	Amazon.in	Standard	T-shirt	S	On the Way	0	INR
)22-4-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	3XL	Shipped	1	INR
)22-4-30	Shipped	Amazon	Amazon.in	Expedited	Shirt	XL	Shipped	1	INR
)22-4-30	Cancelled	Merchant	Amazon.in	Standard	Blazzer	L	On the Way	0	INR
)22-4-30	Shipped	Amazon	Amazon.in	Expedited	Trousers	3XL	Shipped	1	INR
...	...	...	...	...	...	...	...	...	...
)22-5-31	Shipped	Amazon	Amazon.in	Expedited	Shirt	XL	Shipped	1	INR
)22-5-31	Shipped	Amazon	Amazon.in	Expedited	T-shirt	M	Shipped	1	INR
)22-5-31	Shipped	Amazon	Amazon.in	Expedited	Blazzer	XXL	Shipped	1	INR
)22-5-31	Shipped	Amazon	Amazon.in	Expedited	T-shirt	XS	Shipped	1	INR
)22-5-31	Shipped	Amazon	Amazon.in	Expedited	T-shirt	S	Shipped	1	INR

In [92]:
```
df.describe(include='object')
```

Out[92]:

	Order ID	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	cur
count	128976	128976	128976	128976	128976	128976	128976	128976	12
unique	120229	13	2	2	2	9	11	4	
top	403-4984515-8861958	Shipped	Amazon	Amazon.in	Expedited	T-shirt	M	Shipped	
freq	12	77815	89713	128852	88630	50292	22373	109486	12

In [93]:
```
# Use describe() for specific columns
df[['Qty','Amount']].describe()
```

Out[93]:

	Qty	Amount
count	128976.000000	121176.000000
mean	0.904401	648.562176
std	0.313368	281.185041
min	0.000000	0.000000
25%	1.000000	449.000000
50%	1.000000	605.000000
75%	1.000000	788.000000
max	15.000000	5584.000000

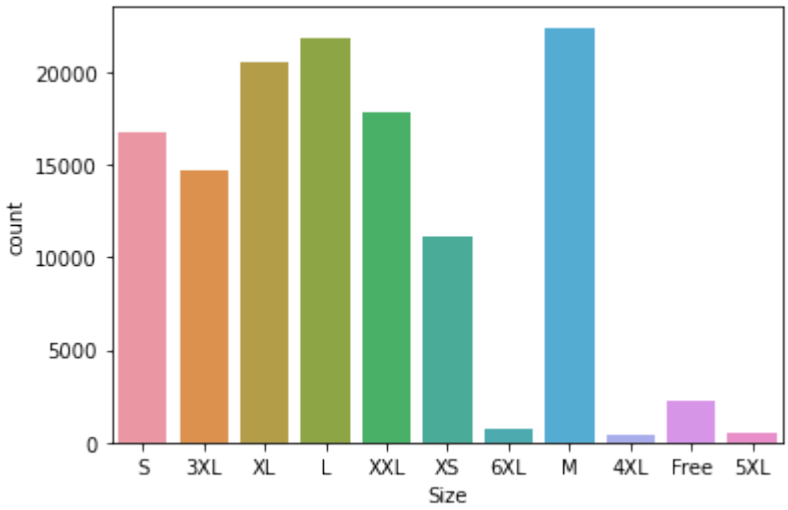
Exploratory Data Analysis

In [94]:
```
df.columns
```

Out[94]:
```
Index(['index', 'Order ID', 'Date', 'Status', 'Fulfilment', 'Sales Chan
nel',
       'ship-service-level', 'Category', 'Size', 'Courier Status', 'Qt
y',
       'currency', 'Amount', 'ship-city', 'ship-state', 'ship-postal-co
de',
       'ship-country', 'B2B', 'fulfilled-by'],
      dtype='object')
```

* Size

In [95]:
```
ax = sns.countplot(x='Size',data=df)
```



* Group By

The GroupBy() function in pandas is used to group data based on 1 or more columns in a Dataframe

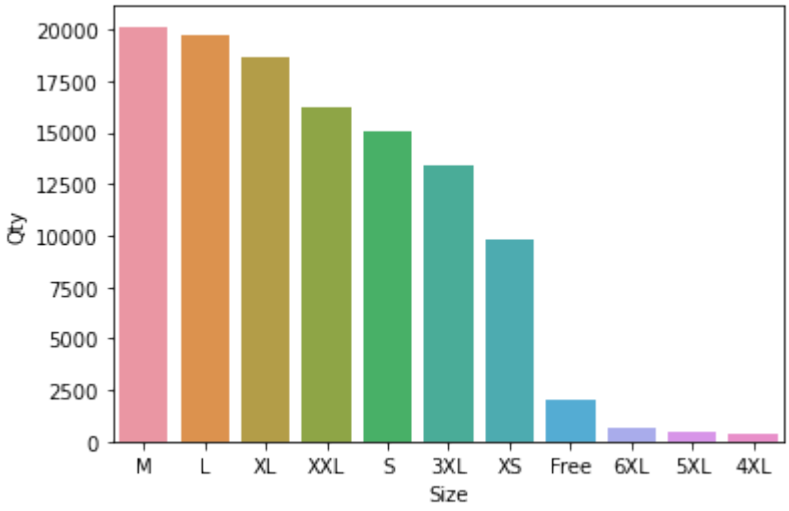
In [98]:
```
df.groupby(['Size'], as_index=False)['Qty'].sum().sort_values(by='Qty',a
```

Out[98]:

	Size	Qty
6	M	20138
5	L	19706
8	XL	18636
10	XXL	16246
7	S	15041
0	3XL	13360
9	XS	9850
4	Free	2070
3	6XL	688
2	5XL	513
1	4XL	398

In [99]:
```
S_Qty=df.groupby(['Size'], as_index=False)['Qty'].sum().sort_values(by='

sns.barplot(x='Size',y='Qty',data=S_Qty)
```

Out[99]: <AxesSubplot:xlabel='Size', ylabel='Qty'>

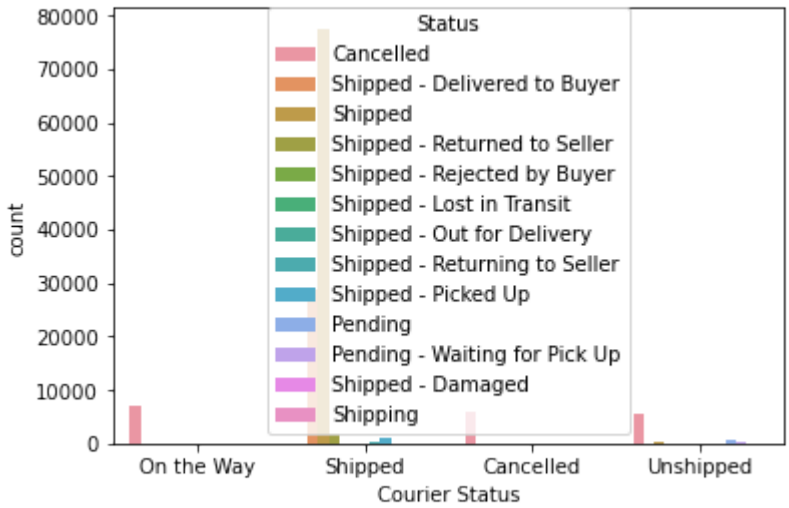


From above Graph you can see that most of the Qty buys M-size in the sales

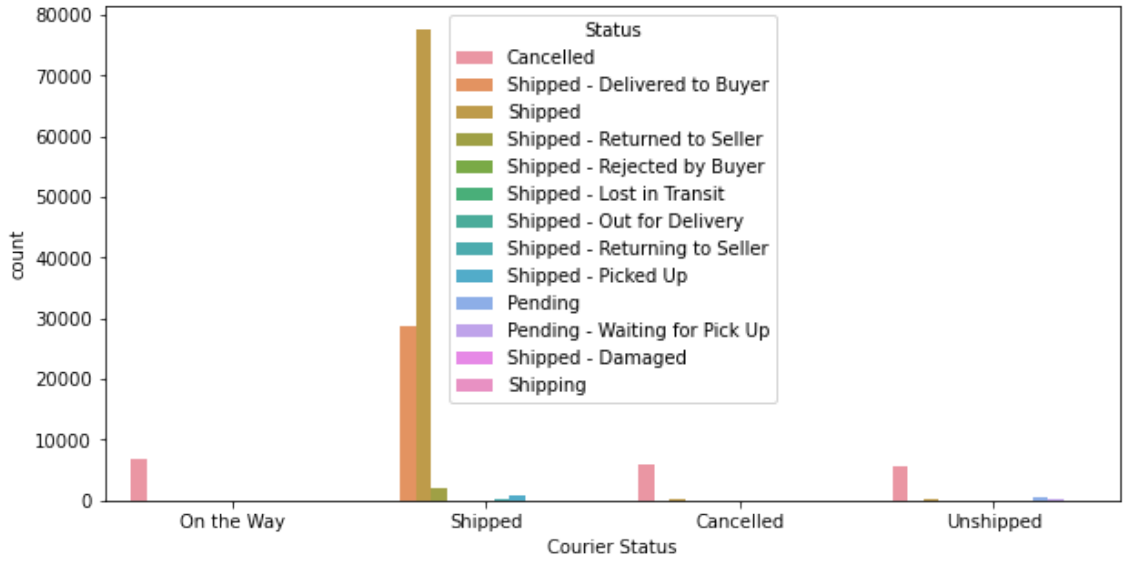
* Courier Status

In [100]:
```
sns.countplot(data=df, x='Courier Status', hue= 'Status')
```

Out[100]: <AxesSubplot:xlabel='Courier Status', ylabel='count'>



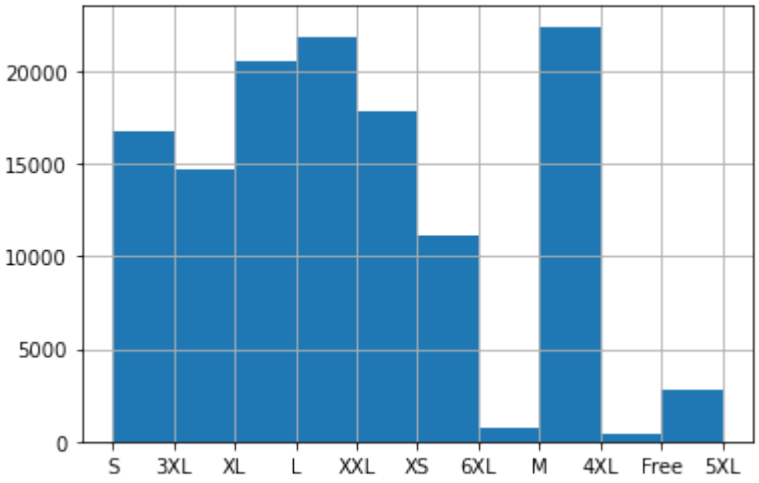
In [101]:
```
plt.figure(figsize=(10,5))
ax = sns.countplot(data=df, x='Courier Status', hue='Status')
plt.show()
```



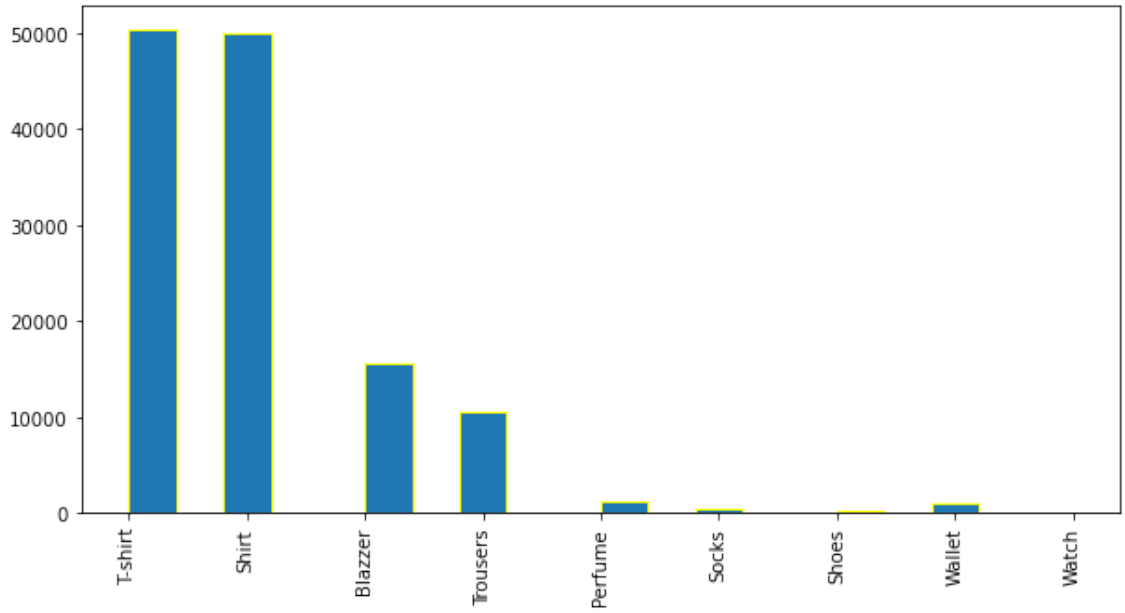
From Above Graph the majorityof the orders are shipped through the courier

In [102]:
```
# Histogram

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

Out[102]: <AxesSubplot:>

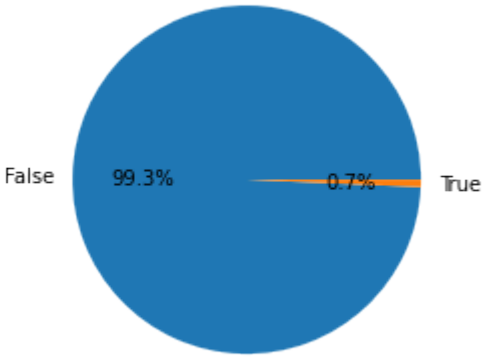


In [103]:
```
df['Category'] = df['Category'].astype(str)
column_data = df['Category']
plt.figure(figsize = (10,5))
plt.hist(column_data, bins=20, edgecolor='Yellow')
plt.xticks(rotation=90)
plt.show()
```



From above Graph you can see that most of the buyers are T-Shirt

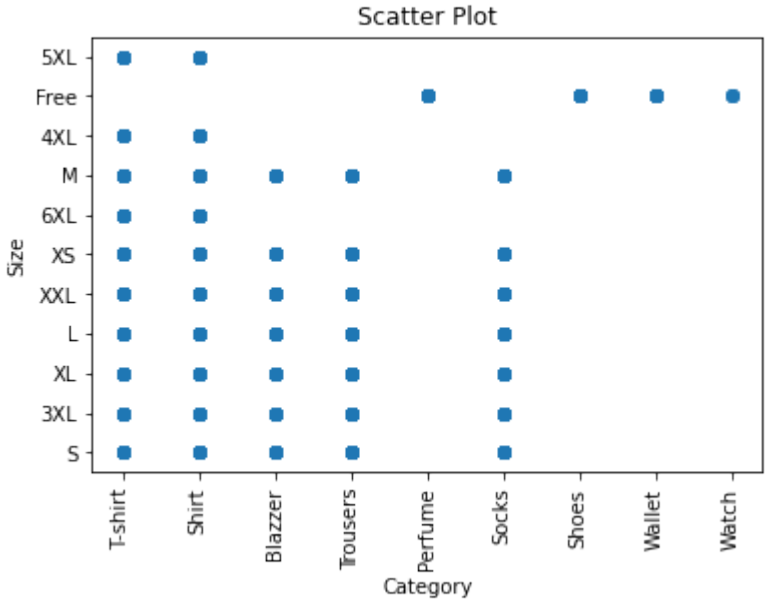
In [107]:
```
# Checking B2B Data by using pie chart

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. 99.3% of buyers are retailers and 0.7% are B2B buyers

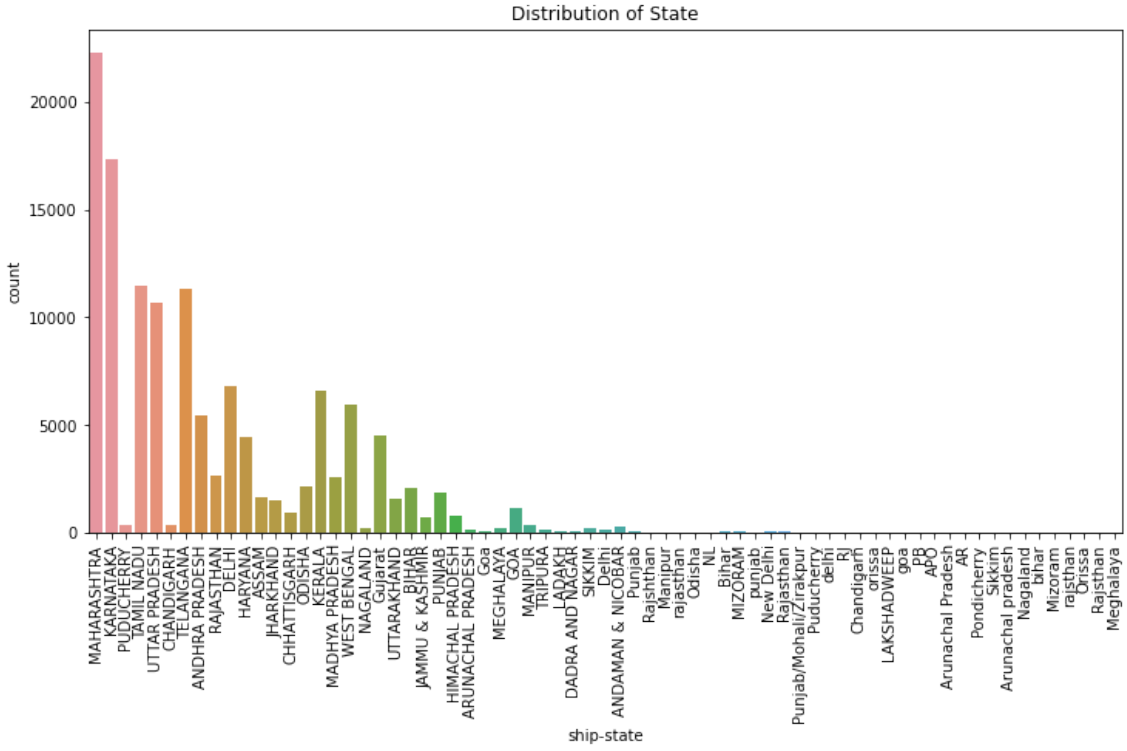
Category and Size the products

In [109]:
```
# 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.xticks(rotation=90)
plt.show()
```

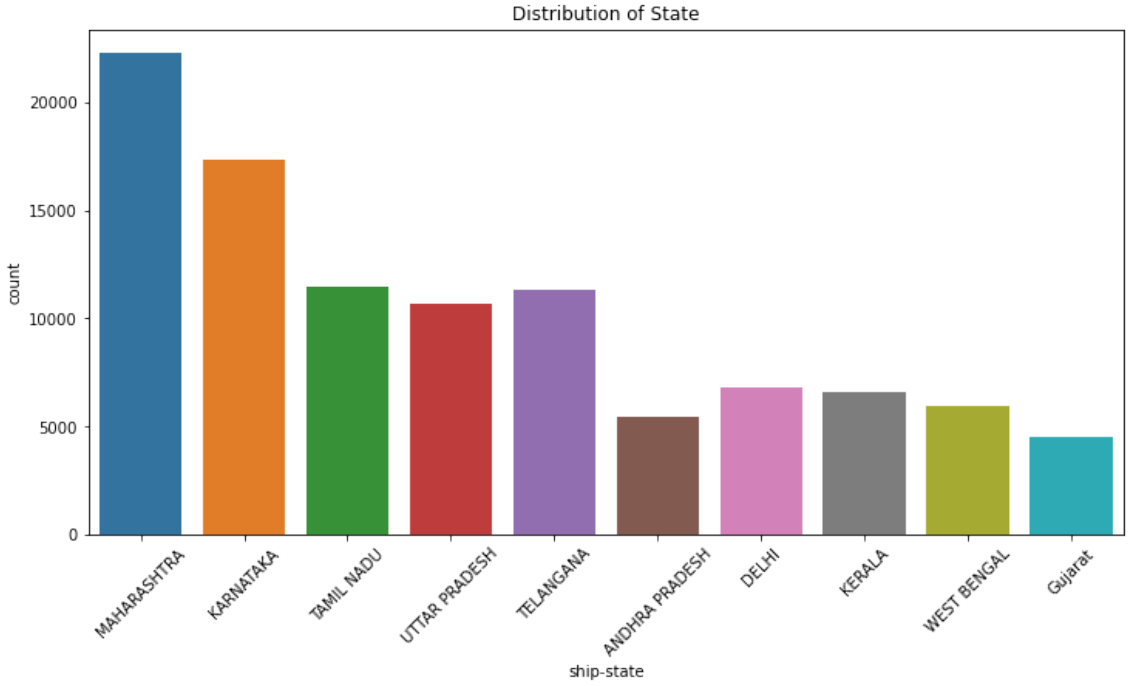


Which city or state buy the most of the products

In [110]:
```
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 [115]:
```
# 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='shi
plt.xlabel('ship-state')
plt.ylabel('count')
plt.title('Distribution of State')
plt.xticks(rotation=45)
plt.show()
```



From above Graph you can see that most of the buyers are Maharastra State

-- Conclusion

The Data Analysis reveals that the business has a significant customers base in Maharastra State, mainly serves retailers, orders through Amazon, experiences high demand for T-shirt, and sees M-Size as the preferred choice among buyers.

In [ ]:
```
```