Amazon Sales Analysis

Introduction:

This project analyzes Amazon sales data to uncover trends, customer preferences, and sales performance. Using data visualization and analysis, it provides actionable insights for better decision-making in the e-commerce space.

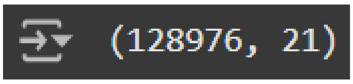
# import python libraries

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

# import csv file

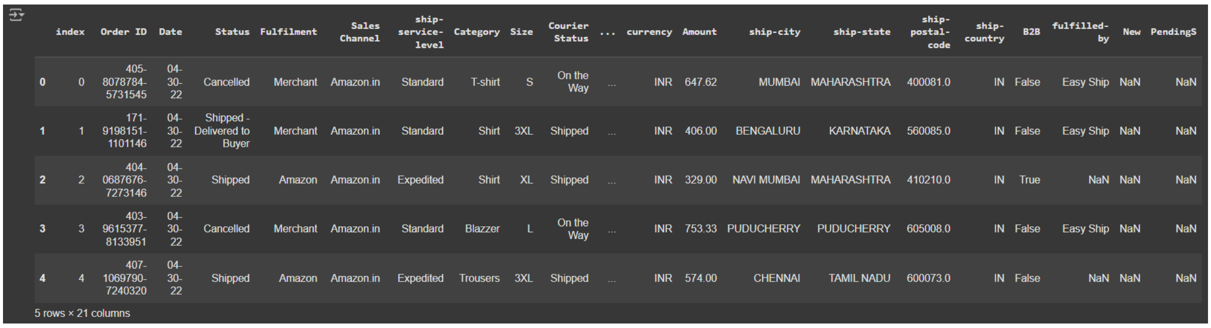
```
df = pd.read_csv('/content/Amazon Sale Report.csv', encoding= 'unicode_escape')
```

df.shape



```
(128976, 21)
```

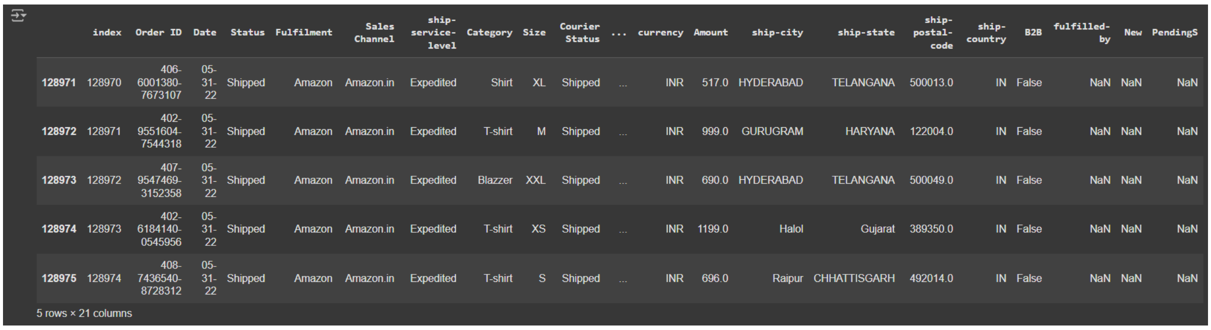
df.head()



	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	...	currency	Amount	ship-city	ship-state	ship-postal-code	ship-country	B2B	fulfilled-by	New	PendingS
0	0	405-8078784-5731545	04-30-22	Cancelled	Merchant	Amazon.in	Standard	T-shirt	S	On the Way	...	INR	647.62	MUMBAI	MAHARASHTRA	400081.0	IN	False	Easy Ship	NaN	NaN
1	1	171-9198151-1101146	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	3XL	Shipped	...	INR	406.00	BENGALURU	KARNATAKA	560085.0	IN	False	Easy Ship	NaN	NaN
2	2	404-0687676-7273146	04-30-22	Shipped	Amazon	Amazon.in	Expedited	Shirt	XL	Shipped	...	INR	329.00	NAVI MUMBAI	MAHARASHTRA	410210.0	IN	True	NaN	NaN	NaN
3	3	403-9615377-8133951	04-30-22	Cancelled	Merchant	Amazon.in	Standard	Blazzer	L	On the Way	...	INR	753.33	PUDUCHERRY	PUDUCHERRY	605008.0	IN	False	Easy Ship	NaN	NaN
4	4	407-1069790-7240320	04-30-22	Shipped	Amazon	Amazon.in	Expedited	Trousers	3XL	Shipped	...	INR	574.00	CHENNAI	TAMIL NADU	600073.0	IN	False	NaN	NaN	NaN

5 rows × 21 columns

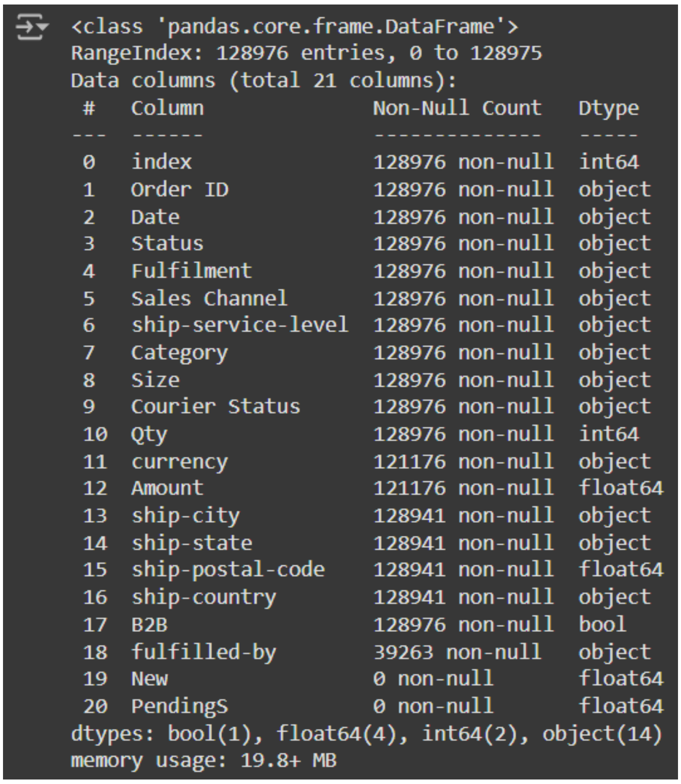
df.tail()



	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	...	currency	Amount	ship-city	ship-state	ship-postal-code	ship-country	B2B	fulfilled-by	New	PendingS
128971	128970	406-6001380-7673107	05-31-22	Shipped	Amazon	Amazon.in	Expedited	Shirt	XL	Shipped	...	INR	517.0	HYDERABAD	TELANGANA	500013.0	IN	False	NaN	NaN	NaN
128972	128971	402-9551604-7544318	05-31-22	Shipped	Amazon	Amazon.in	Expedited	T-shirt	M	Shipped	...	INR	999.0	GURUGRAM	HARYANA	122004.0	IN	False	NaN	NaN	NaN
128973	128972	407-9547469-3152358	05-31-22	Shipped	Amazon	Amazon.in	Expedited	Blazzer	XXL	Shipped	...	INR	690.0	HYDERABAD	TELANGANA	500049.0	IN	False	NaN	NaN	NaN
128974	128973	402-6184140-0545956	05-31-22	Shipped	Amazon	Amazon.in	Expedited	T-shirt	XS	Shipped	...	INR	1199.0	Halol	Gujarat	389350.0	IN	False	NaN	NaN	NaN
128975	128974	408-7436540-8728312	05-31-22	Shipped	Amazon	Amazon.in	Expedited	T-shirt	S	Shipped	...	INR	696.0	Raipur	CHHATTISGARH	492014.0	IN	False	NaN	NaN	NaN

5 rows × 21 columns

df.info()



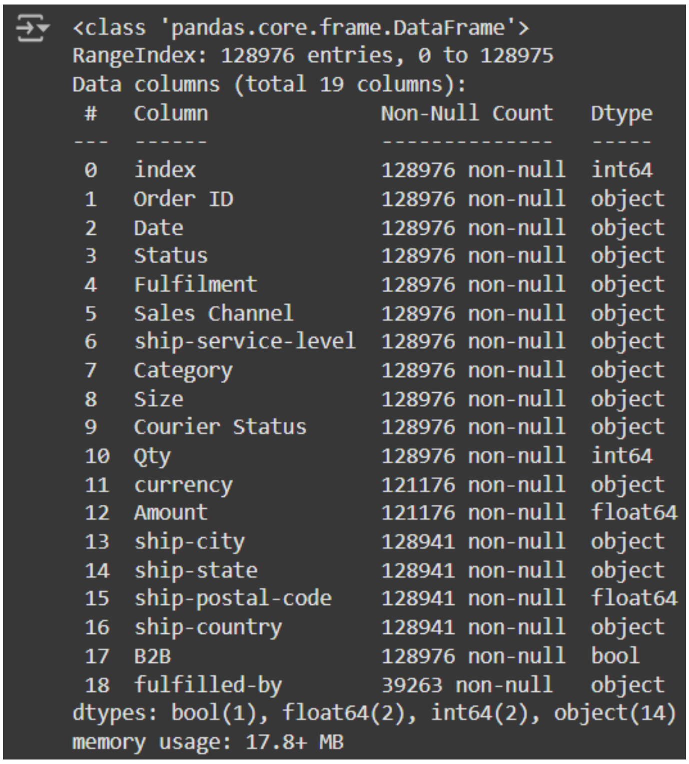
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 128976 entries, 0 to 128975
Data columns (total 21 columns):
 #   Column              Non-Null Count   Dtype  
---  ------              --------------   -----  
 0   index               128976 non-null  int64  
 1   Order ID            128976 non-null  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)

# after dropping unrelated/blank columns

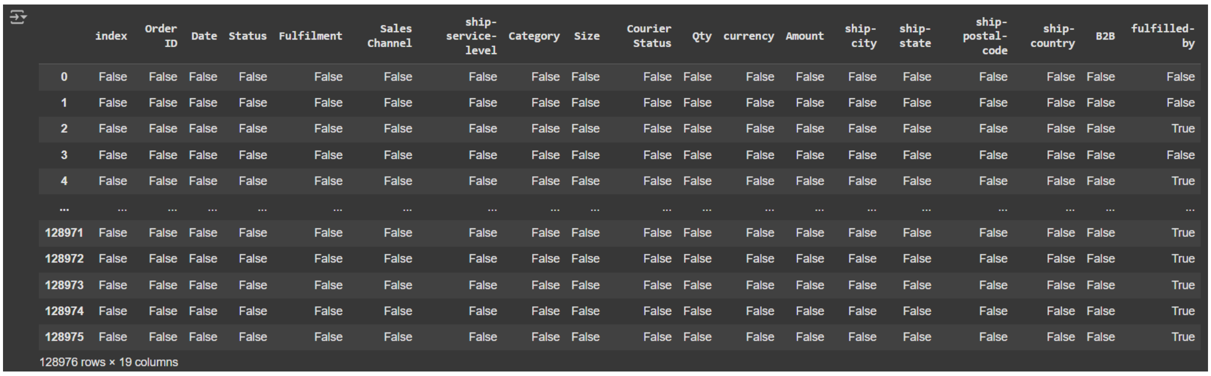
df.info()



```
<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
```

# checking null value

pd.isnull(df)

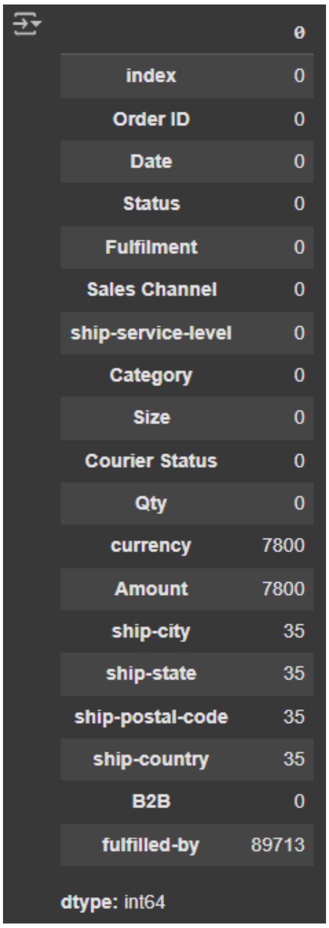


	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	Qty	currency	Amount	ship-city	ship-state	ship-postal-code	ship-country	B2B	fulfilled-by
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	True
3	False	False	False	False	False	False	False	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	False	False	False	False	False	False	True
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
128971	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	True
128972	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	True
128973	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	True
128974	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	True
128975	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	True

128976 rows × 19 columns

# sum will give total values of null values

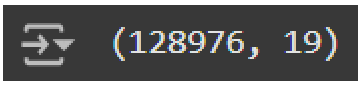
pd.isnull(df).sum()



	0
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

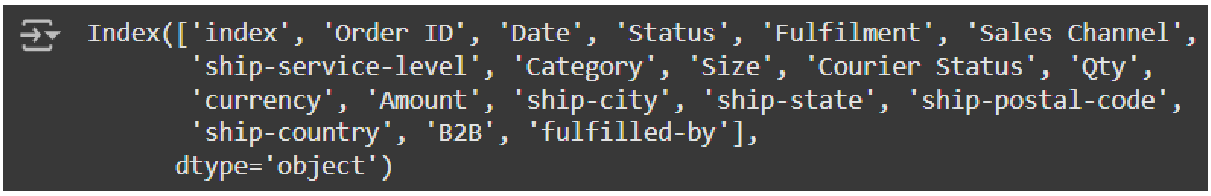


```
(128976, 19)
```

# drop null values

df.dropna(inplace=True)

```
df.columns
```



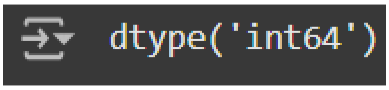
```
Index(['index', 'Order ID', 'Date', 'Status', 'Fulfilment', 'Sales Channel',
       'ship-service-level', 'Category', 'Size', 'Courier Status', 'Qty',
       'currency', 'Amount', 'ship-city', 'ship-state', 'ship-postal-code',
       'ship-country', 'B2B', 'fulfilled-by'],
      dtype='object')
```

# change data type

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

#checking whether the data type change or not

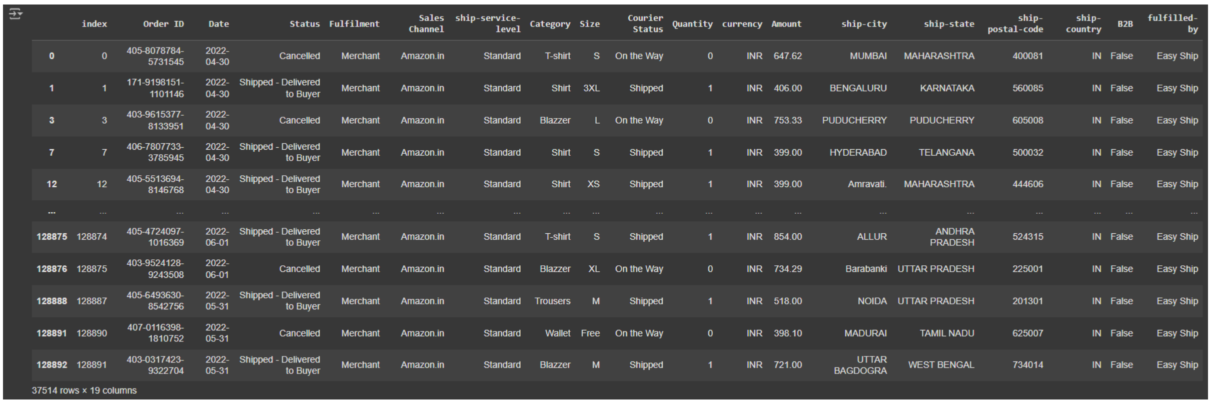
```
df['ship-postal-code'].dtype
```



```
dtype('int64')
```

#rename Columns

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

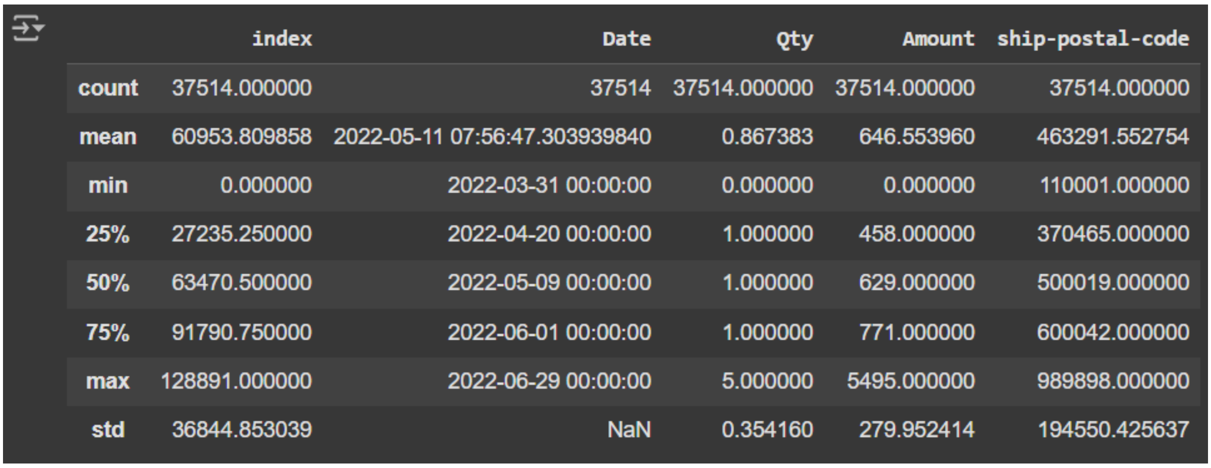


	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	Quantity	currency	Amount	ship-city	ship-state	ship-postal-code	ship-country	B2B	fulfilled-by
0	0	405-8078784-5731545	2022-04-30	Cancelled	Merchant	Amazon.in	Standard	T-shirt	S	On the Way	0	INR	647.62	MUMBAI	MAHARASHTRA	400081	IN	False	Easy Ship
1	1	171-9198151-1101146	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	3XL	Shipped	1	INR	406.00	BENGALURU	KARNATAKA	560085	IN	False	Easy Ship
3	3	403-9615377-8133951	2022-04-30	Cancelled	Merchant	Amazon.in	Standard	Blazzer	L	On the Way	0	INR	753.33	PUDUCHERRY	PUDUCHERRY	605008	IN	False	Easy Ship
7	7	406-7807733-3785945	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	S	Shipped	1	INR	399.00	HYDERABAD	TELANGANA	500032	IN	False	Easy Ship
12	12	405-5513694-8146768	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	XS	Shipped	1	INR	399.00	Amravati.	MAHARASHTRA	444606	IN	False	Easy Ship
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
128875	128874	405-4724097-1016369	2022-06-01	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	T-shirt	S	Shipped	1	INR	854.00	ALLUR	ANDHRA PRADESH	524315	IN	False	Easy Ship
128876	128875	403-9524128-9243508	2022-06-01	Cancelled	Merchant	Amazon.in	Standard	Blazzer	XL	On the Way	0	INR	734.29	Barabanki	UTTAR PRADESH	225001	IN	False	Easy Ship
128888	128887	405-6493630-8542756	2022-05-31	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Trousers	M	Shipped	1	INR	518.00	NOIDA	UTTAR PRADESH	201301	IN	False	Easy Ship
128891	128890	407-0116398-1810752	2022-05-31	Cancelled	Merchant	Amazon.in	Standard	Wallet	Free	On the Way	0	INR	398.10	MADURAI	TAMIL NADU	625007	IN	False	Easy Ship
128892	128891	403-0317423-9322704	2022-05-31	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Blazzer	M	Shipped	1	INR	721.00	UTTAR BAGDOGRA	WEST BENGAL	734014	IN	False	Easy Ship

37514 rows × 19 columns

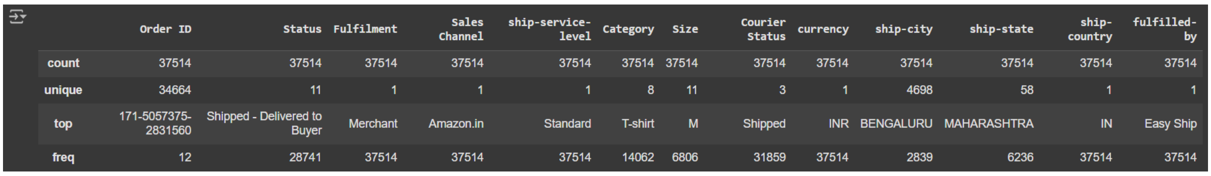
#describe() method return description of the data in the DataFrame(i.e count,mean,std,min..etc)

```
df.describe()
```



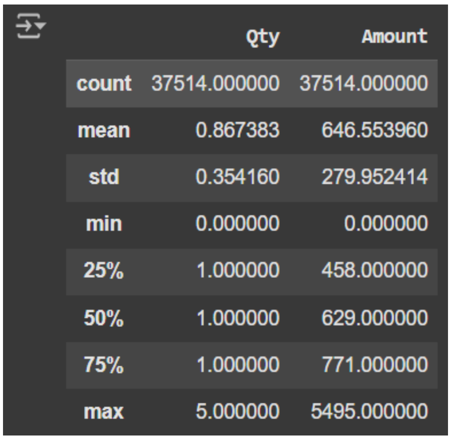
	index	Date	Qty	Amount	ship-postal-code
count	37514.000000	37514	37514.000000	37514.000000	37514.000000
mean	60953.809858	2022-05-11 07:56:47.303939840	0.867383	646.553960	463291.552754
min	0.000000	2022-03-31 00:00:00	0.000000	0.000000	110001.000000
25%	27235.250000	2022-04-20 00:00:00	1.000000	458.000000	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

```
# summary of categorical data
df.describe(include='object')
```



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

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

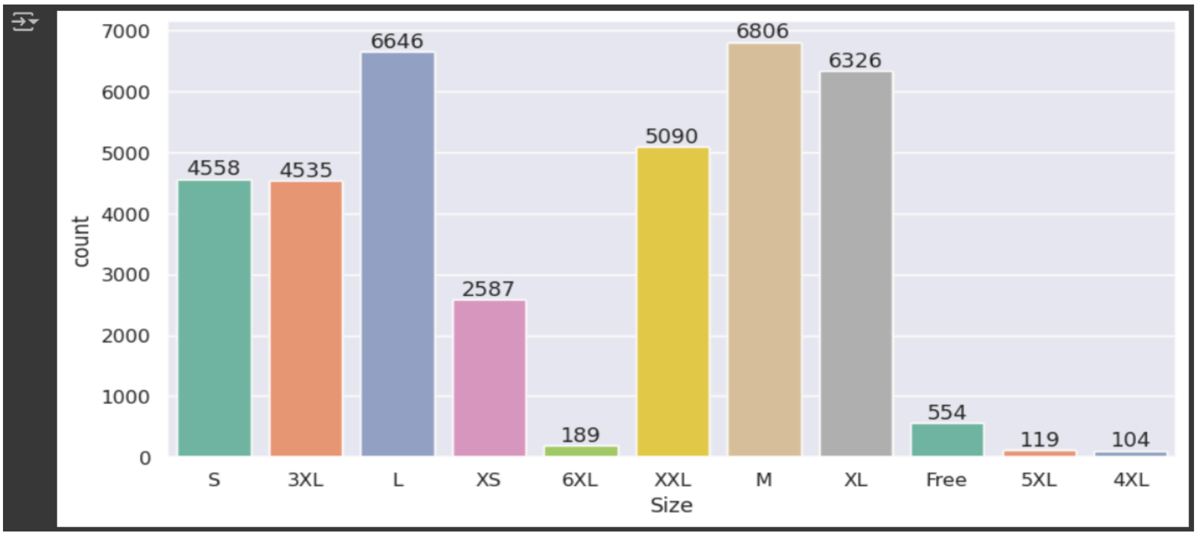


	Qty	Amount
count	37514.000000	37514.000000
mean	0.867383	646.553960
std	0.354160	279.952414
min	0.000000	0.000000
25%	1.000000	458.000000
50%	1.000000	629.000000
75%	1.000000	771.000000
max	5.000000	5495.000000

Exploratory Data Analysis

```
# Setting Seaborn style for better visuals
sns.set(style="whitegrid")
```

```
# Size Distribution
ax = sns.countplot(x='Size', data=df, hue='Size', palette="Set2")
for bars in ax.containers:
    ax.bar_label(bars)
```

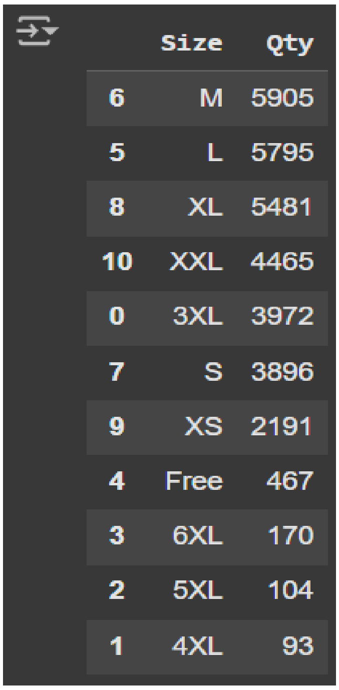


Note: From above Graph you can see that most of the people buys M-Size

Group By

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

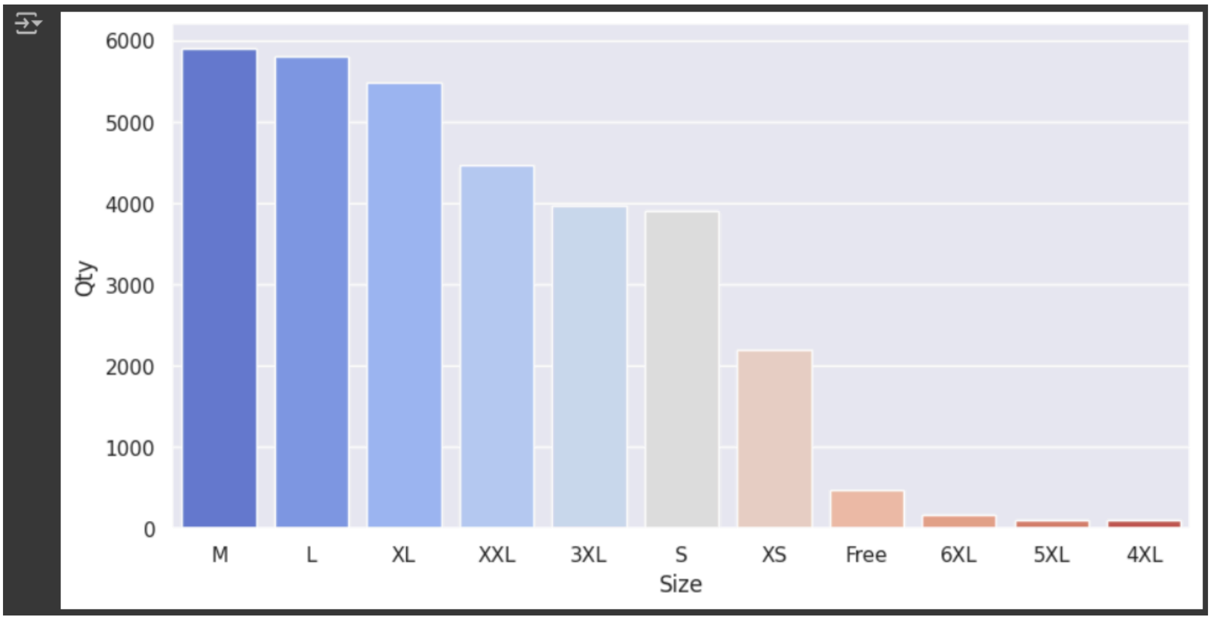
```
df.groupby(['Size'], as_index=False)['Qty'].sum().sort_values(by='Qty',ascending=False)
```



	Size	Qty
6	M	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

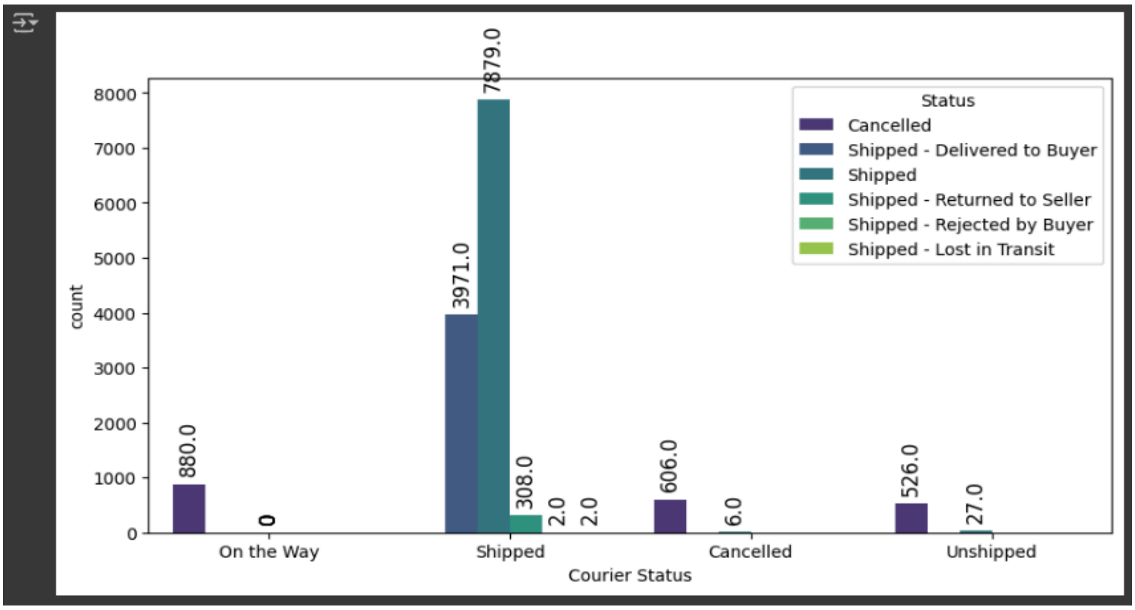
# Quantity Distribution by Size

```
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, hue='Size', palette="coolwarm")
plt.show()
```



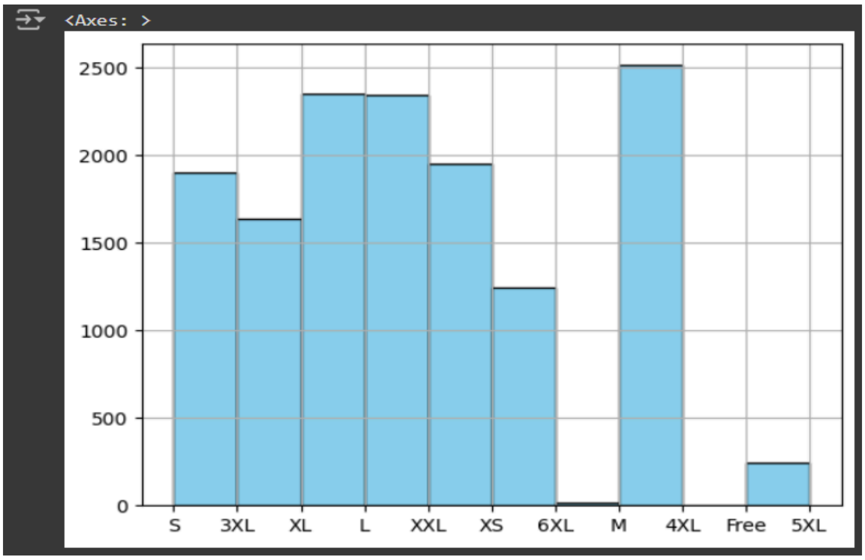
Note: From above Graph you can see that most of the Qty buys M-Size in the sales

```
# Courier Status Distribution with Count Annotations
plt.figure(figsize=(10, 5))
ax = sns.countplot(data=df, x='Courier Status', hue='Status', palette="viridis")
for p in ax.patches:
   height = p.get_height()
   ax.annotate(f'{height}',
            (p.get_x() + p.get_width() / 2., height),
            ha='center', va='bottom',
            fontsize=12, color='black',
            xytext=(0, 5), textcoords='offset points',
            rotation=90)
plt.show()
```



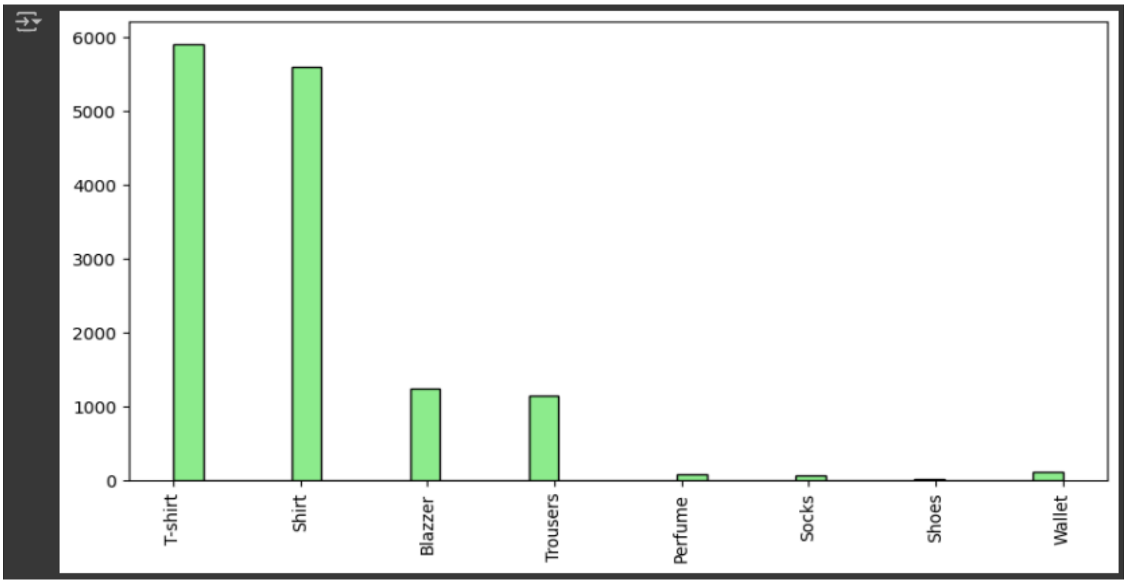
Note: From above Graph the majority of the orders are shipped through the courier.

```
# Histogram
df['Size'].hist(color="skyblue", edgecolor="black")
```



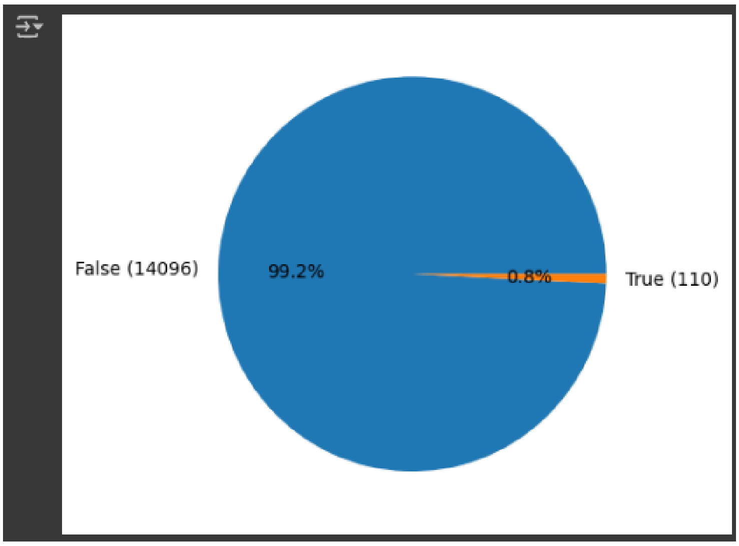
```
<Axes: >
```

```
# Category Distribution with Histogram
df['Category'] = df['Category'].astype(str)
column_data = df['Category']
plt.figure(figsize=(10, 5))
plt.hist(column_data, bins=30, edgecolor='Black', color="lightgreen")
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 a pie chart
B2B_Check = df['B2B'].value_counts()
# Create labels that include both the category names and their count values
labels = [f'{index} ({value})' for index, value in zip(B2B_Check.index, B2B_Check.values)]
plt.pie(B2B_Check, labels=labels, autopct='%1.1f%%',)
plt.show()
```



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

# Fulfilment Distribution with Pie Chart

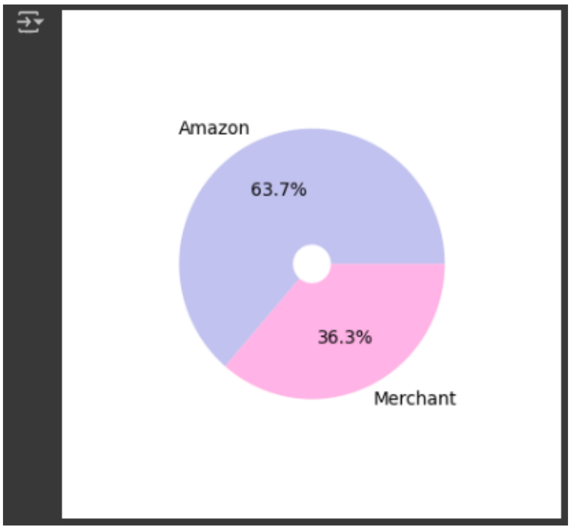
a1 = df['Fulfilment'].value_counts()

fig, ax = plt.subplots()

ax.pie(a1, labels=a1.index, autopct='%1.1f%%', radius=0.7, wedgeprops=dict(width=0.6), colors=["#c2c2f0", "#ffb3e6"])

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

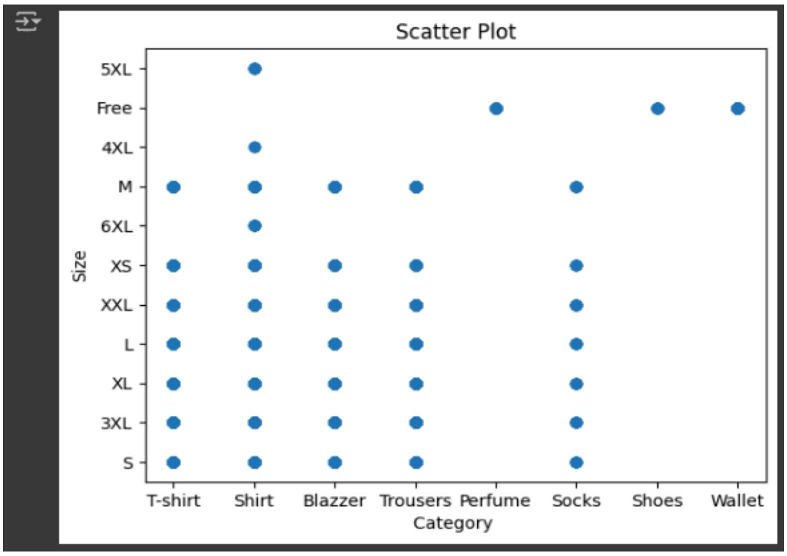
plt.scatter(x_data, y_data,)

plt.xlabel('Category ')

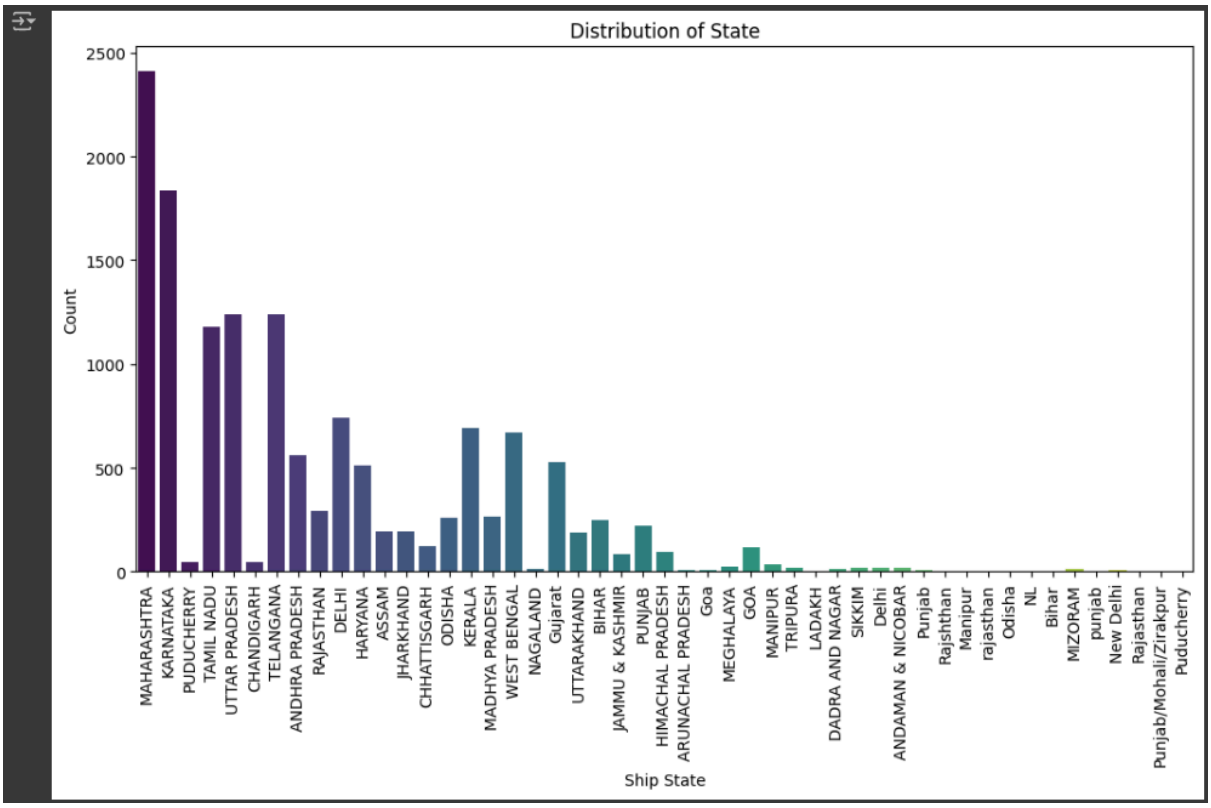
plt.ylabel('Size')

plt.title('Scatter Plot')

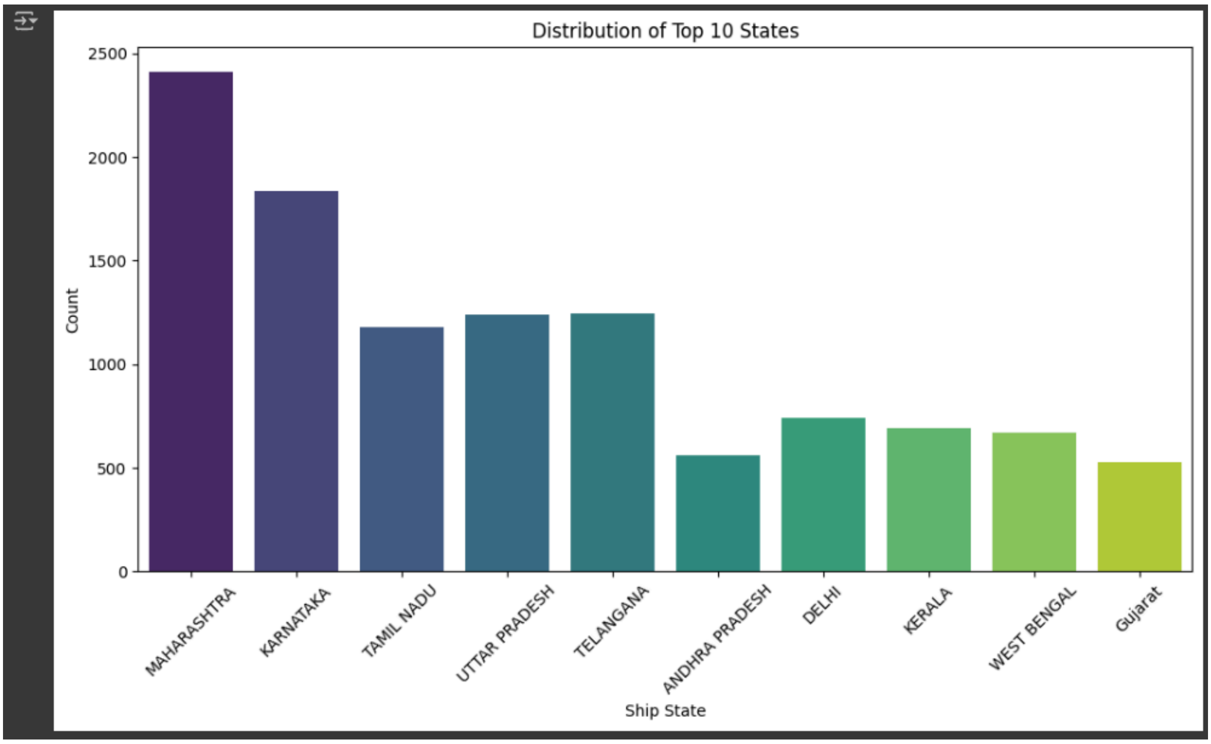
plt.show()



```
# Ship State Distribution with Count Plot
plt.figure(figsize=(12, 6))
sns.countplot(data=df, x='ship-state', hue='ship-state', palette="viridis", dodge=False)
plt.xlabel('Ship State')
plt.ylabel('Count')
plt.title('Distribution of State')
plt.xticks(rotation=90)
plt.legend([],[], frameon=False)
plt.show()
```



```
# Distribution of Top 10 States by Shipment Count
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',
palette="viridis", hue='ship-state')
plt.xlabel('Ship State')
plt.ylabel('Count')
plt.title('Distribution of Top 10 States')
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.