

Import Libraries

```
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(2)

{"type": "dataframe", "variable_name": "df"}

df.tail(2)

{"type": "dataframe"}

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
```

```
{"type": "dataframe"}
```

Data Cleaning

```
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('int64')
```

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

```
<ipython-input-18-5c207e96e7cb>: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 format.

```
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'})
```

```
{"summary":{"\n  \"name\": \"df\",\n  \"rows\": 37514,\n  \"fields\":  
[\n    {\n      \"column\": \"index\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 36844,\n        \"min\": 0,\n        \"max\": 128891,\n        \"num_unique_values\": 37464,\n        \"samples\": [\n          68140,\n          71868,\n          104098\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Order ID\",\n      \"properties\": {\n        \"dtype\": \"string\",\n        \"num_unique_values\": 34664,\n        \"samples\": [\n          \"403-3148026-7474712\",\n          \"171-3343065-8758738\",\n          \"171-2686399-8780363\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Date\",\n      \"properties\": {\n        \"dtype\": \"date\",\n        \"min\": \"2022-03-31 00:00:00\",\n        \"max\": \"2022-06-29 00:00:00\",\n        \"num_unique_values\": 91,\n        \"samples\": [\n          \"2022-05-22 00:00:00\",\n          \"2022-04-08 00:00:00\",\n          \"2022-05-07 00:00:00\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Status\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 11,\n        \"samples\": [\n          \"Shipped - Out for  
Delivery\",\n          \"Cancelled\",\n          \"Pending - Waiting  
for Pick Up\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Fulfilment\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 1,\n        \"samples\": [\n          \"Merchant\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Sales Channel\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 1,\n        \"samples\": [\n          \"Amazon.in\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"ship-service-level\",
```

```

{"properties": {"dtype": "category",
"num_unique_values": 1,
"Standard": },
"description": },
"Category": {"properties": {"dtype": "category",
"category": "num_unique_values": 8,
"samples": ["Shirt"],
"semantic_type": },
"description": },
"Size": {"properties": {"dtype": "category",
"num_unique_values": 11,
"samples": ["XXL"],
"semantic_type": },
"description": },
"Courier Status": {"properties": {"dtype": "category",
"category": "num_unique_values": 3,
"samples": ["On the Way"],
"semantic_type": },
"description": },
"column": "Quantity",
"properties": {"dtype": "number",
"number": "std": 0,
"min": 0,
"max": 5,
"num_unique_values": 6,
"samples": [0],
"semantic_type": },
"description": },
"currency": {"properties": {"dtype": "category",
"category": "num_unique_values": 1,
"samples": ["INR"],
"semantic_type": },
"description": },
"Amount": {"properties": {"dtype": "number",
"std": 279.95241409947926,
"min": 0.0,
"max": 5495.0,
"num_unique_values": 1238,
"samples": [597.14],
"semantic_type": },
"description": },
"ship-city": {"properties": {"dtype": "category",
"category": "num_unique_values": 4698,
"samples": ["KUNKURI"],
"semantic_type": },
"description": },
"ship-state": {"properties": {"dtype": "category",
"num_unique_values": 58,
"samples": ["MAHARASHTRA"],
"semantic_type": },
"description": },
"ship-postal-code": {"properties": {"dtype": "number",
"std": 194550,
"min": 110001,
"max": 989898,
"num_unique_values": 6205,
"samples": [673613],
"semantic_type": },
"description": },
"ship-country": {"properties": {"dtype": "category",
"num_unique_values": 1,
"samples": ["IN"],
"semantic_type": },
"description": },
"column": "B2B",
"properties": {

```

```

n      \ "dtype\ ": \ "boolean\ ",\n      \ "num_unique_values\ ": 2,\n
\ "samples\ ": [\n      true\n      ],\n      \ "semantic_type\ ":
\ "\",\n      \ "description\ ": \ "\",\n      }\n      },\n      {\n
\ "column\ ": \ "fulfilled-by\ ",\n      \ "properties\ ": {\n
\ "dtype\ ": \ "category\ ",\n      \ "num_unique_values\ ": 1,\n
\ "samples\ ": [\n      \ "Easy Ship\ "\n      ],\n
\ "semantic_type\ ": \ "\",\n      \ "description\ ": \ "\",\n      }\n
n      }\n      ]\n      }", "type": "dataframe"}

```

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

```
df.describe()
```

```

{"summary": "{\n  \ "name\ ": \ "df\ ",\n  \ "rows\ ": 8,\n  \ "fields\ ": [\n
{\n    \ "column\ ": \ "index\ ",\n    \ "properties\ ": {\n
\ "dtype\ ": \ "number\ ",\n    \ "std\ ": 40260.3593865165,\n
\ "min\ ": 0.0,\n    \ "max\ ": 128891.0,\n
\ "num_unique_values\ ": 8,\n    \ "samples\ ": [\n
60953.80985765314,\n    91790.75,\n    37514.0\n
n    ],\n    \ "semantic_type\ ": \ "\",\n
\ "description\ ": \ "\",\n    }\n    },\n    {\n    \ "column\ ":
\ "Date\ ",\n    \ "properties\ ": {\n    \ "dtype\ ": \ "date\ ",\n
\ "min\ ": \ "1970-01-01 00:00:00.000037514\ ",\n    \ "max\ ": \ "2022-
06-29 00:00:00\ ",\n    \ "num_unique_values\ ": 7,\n
\ "samples\ ": [\n    \ "37514\ ",\n    \ "2022-05-11
07:56:47.303939840\ ",\n    \ "2022-06-01 00:00:00\ "\n    ],\n
\ "semantic_type\ ": \ "\",\n    \ "description\ ": \ "\",\n    }\n
n    },\n    {\n    \ "column\ ": \ "Qty\ ",\n    \ "properties\ ": {\n
\ "dtype\ ": \ "number\ ",\n    \ "std\ ": 13262.736226567698,\n
\ "min\ ": 0.0,\n    \ "max\ ": 37514.0,\n
\ "num_unique_values\ ": 6,\n    \ "samples\ ": [\n    37514.0,\n
n    0.8673828437383377,\n    0.354160004012791\n
n    ],\n    \ "semantic_type\ ": \ "\",\n
\ "description\ ": \ "\",\n    }\n    },\n    {\n    \ "column\ ":
\ "Amount\ ",\n    \ "properties\ ": {\n    \ "dtype\ ": \ "number\ ",\n
\ "std\ ": 12967.326077075897,\n    \ "min\ ": 0.0,\n    \ "max\ ":
37514.0,\n    \ "num_unique_values\ ": 8,\n    \ "samples\ ": [\n
646.5539598549875,\n    771.0,\n    37514.0\n    ],\n
\ "semantic_type\ ": \ "\",\n    \ "description\ ": \ "\",\n    }\n
n    },\n    {\n    \ "column\ ": \ "ship-postal-code\ ",\n
\ "properties\ ": {\n    \ "dtype\ ": \ "number\ ",\n    \ "std\ ":
307069.6861502951,\n    \ "min\ ": 37514.0,\n    \ "max\ ":
989898.0,\n    \ "num_unique_values\ ": 8,\n    \ "samples\ ": [\n
463291.55275363865,\n    600042.0,\n    37514.0\n    ],\n
\ "semantic_type\ ": \ "\",\n
\ "description\ ": \ "\",\n    }\n    }\n    ]\n    }", "type": "dataframe"}

```

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

```

{"summary":{"\n  \"name\": \"df\", \n  \"rows\": 4, \n  \"fields\": [\n    {\n      \"column\": \"Order ID\", \n      \"properties\": {\n        \"dtype\": \"string\", \n        \"num_unique_values\": 4, \n        \"samples\": [\n          34664, \n          \"12\", \n          \"37514\", \n          \"\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      } \n    }, \n    {\n      \"column\": \"Status\", \n      \"properties\": {\n        \"dtype\": \"string\", \n        \"num_unique_values\": 4, \n        \"samples\": [\n          11, \n          \"28741\", \n          \"37514\", \n          \"\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      } \n    }, \n    {\n      \"column\": \"Fulfilment\", \n      \"properties\": {\n        \"dtype\": \"string\", \n        \"num_unique_values\": 3, \n        \"samples\": [\n          1, \n          \"Merchant\", \n          \"\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      } \n    }, \n    {\n      \"column\": \"Sales Channel\", \n      \"properties\": {\n        \"dtype\": \"string\", \n        \"num_unique_values\": 3, \n        \"samples\": [\n          1, \n          \"Amazon.in\", \n          \"\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      } \n    }, \n    {\n      \"column\": \"ship-service-level\", \n      \"properties\": {\n        \"dtype\": \"string\", \n        \"num_unique_values\": 3, \n        \"samples\": [\n          1, \n          \"Standard\", \n          \"\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      } \n    }, \n    {\n      \"column\": \"Category\", \n      \"properties\": {\n        \"dtype\": \"string\", \n        \"num_unique_values\": 4, \n        \"samples\": [\n          8, \n          \"14062\", \n          \"37514\", \n          \"\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      } \n    }, \n    {\n      \"column\": \"Size\", \n      \"properties\": {\n        \"dtype\": \"string\", \n        \"num_unique_values\": 4, \n        \"samples\": [\n          11, \n          \"6806\", \n          \"37514\", \n          \"\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      } \n    }, \n    {\n      \"column\": \"Courier Status\", \n      \"properties\": {\n        \"dtype\": \"string\", \n        \"num_unique_values\": 4, \n        \"samples\": [\n          3, \n          \"31859\", \n          \"37514\", \n          \"\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      } \n    }, \n    {\n      \"column\": \"currency\", \n      \"properties\": {\n        \"dtype\": \"string\", \n        \"num_unique_values\": 3, \n        \"samples\": [\n          1, \n          \"INR\", \n          \"37514\", \n          \"\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      } \n    }, \n    {\n      \"column\": \"ship-city\", \n      \"properties\": {\n        \"dtype\": \"string\", \n        \"num_unique_values\": 4, \n        \"samples\": [\n          4698, \n          \"2839\", \n          \"37514\", \n          \"\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      } \n    }, \n    {\n      \"column\": \"ship-state\", \n      \"properties\": {\n        \"dtype\": \"string\", \n        \"num_unique_values\": 4, \n        \"samples\": [\n          4698, \n          \"2839\", \n          \"37514\", \n          \"\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      } \n    } \n  ] \n}

```

```

\"samples\": [\n          58,\n          \"6236\",
\"37514\",
],\n          \"semantic_type\": \"\",
\"description\": \"\",
\"column\":
\"ship-country\",
\"properties\": {\n          \"dtype\":
\"string\",
\"num_unique_values\": 3,\n          \"samples\":
[\n          \"37514\",
1,\n          \"IN\",
],\n          \"semantic_type\": \"\",
\"description\": \"\",
n      },\n      {\n          \"column\": \"fulfilled-by\",
\"properties\": {\n          \"dtype\": \"string\",
\"num_unique_values\": 3,\n          \"samples\": [\n          \"37514\",
1,\n          \"Easy Ship\",
],\n          \"semantic_type\": \"\",
\"description\": \"\",
n      }\n      ]\n      }, \"type\": \"dataframe\"}

```

```

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

```

```

{\"summary\": \"\n  \"name\": \"df[['Qty', 'Amount']]\",
  \"rows\": 8,\n  \"fields\": [\n    {\n      \"column\": \"Qty\",
\"properties\": {\n      \"dtype\": \"number\",
13262.736226567698,\n      \"min\": 0.0,\n      \"max\": 37514.0,\n      \"num_unique_values\": 6,\n      \"samples\": [\n      37514.0,\n      0.8673828437383377,\n      5.0\n      ],\n      \"semantic_type\": \"\",
\"description\": \"\",
n    },\n    {\n      \"column\": \"Amount\",
\"properties\":
{\n      \"dtype\": \"number\",
12967.326077075899,\n      \"min\": 0.0,\n      \"max\": 37514.0,\n      \"num_unique_values\": 8,\n      \"samples\": [\n      646.5539598549875,\n      629.0,\n      37514.0\n      ],\n      \"semantic_type\": \"\",
\"description\": \"\",
n    }\n    ]\n  }, \"type\": \"dataframe\"}

```

(EDA) 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, palette='viridis')
plt.xlabel('Size')

```

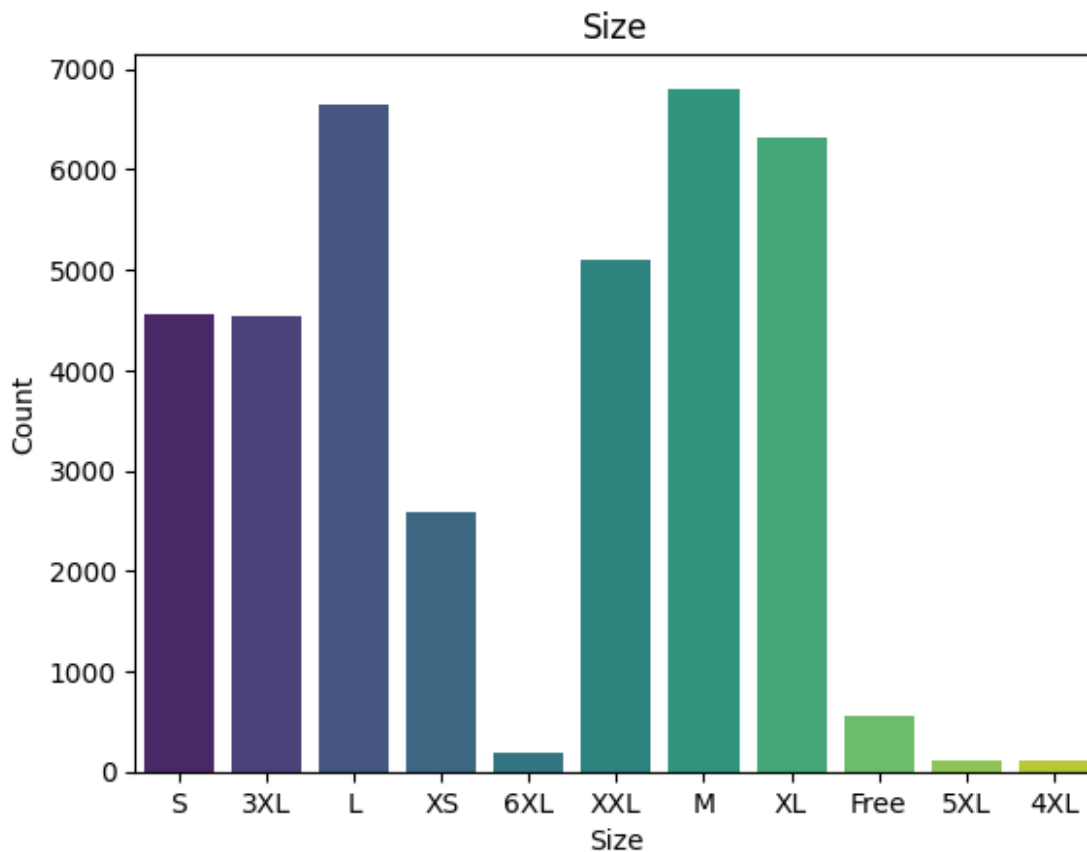


```
plt.ylabel('Count')
plt.title('Size')
plt.show()
```

<ipython-input-31-9fb6ac693d8a>:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

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



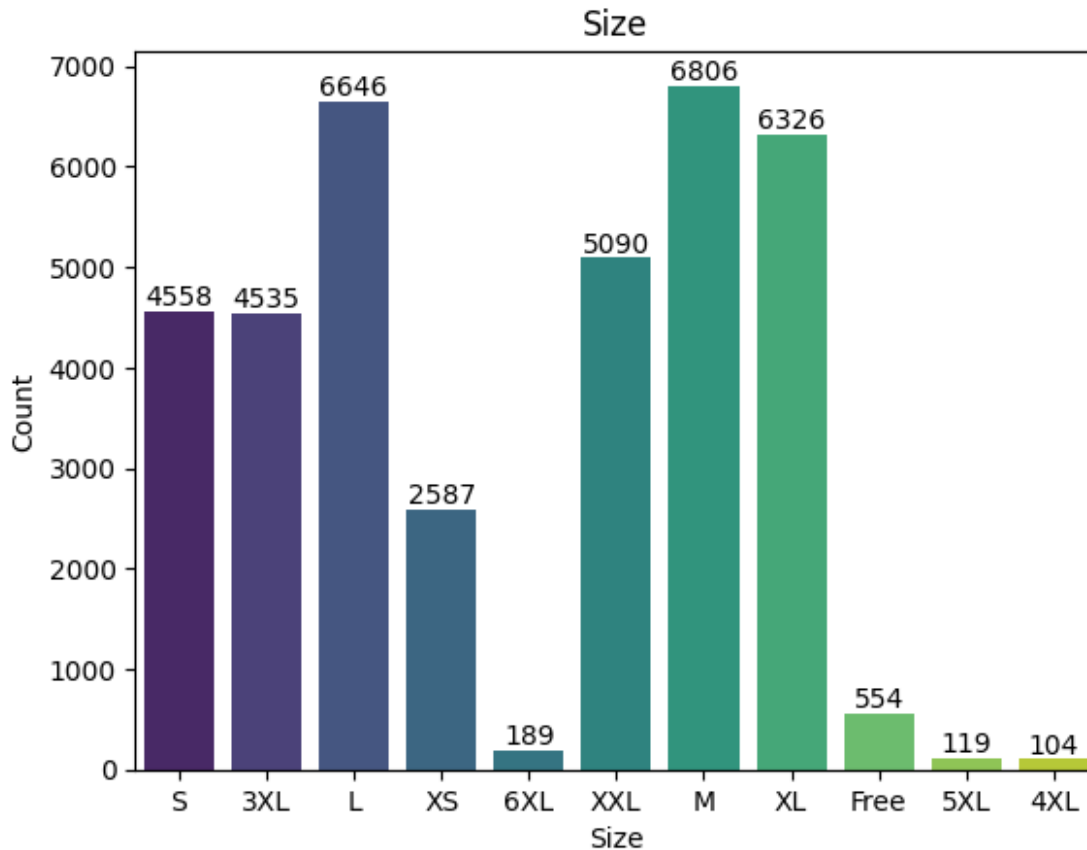
```
ax=sns.countplot(x='Size', data=df, palette='viridis')
for bars in ax.containers:
    ax.bar_label(bars)
plt.xlabel('Size')
plt.ylabel('Count')
plt.title('Size')
plt.show()
```

<ipython-input-35-3fa2991d4739>:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be

removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

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



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)

{"summary":{"name": "df", "rows": 11, "fields": [
  {
    "column": "Size",
    "properties": {
      "dtype": "string",
      "num_unique_values": 11,
      "samples": [
        "S", "M", "5XL"
      ],
      "semantic_type": ""
    }
  },
  {
    "column": "Qty",
    "properties": {
      "dtype": "number",
      "std": 2412,
      "min": 93,
      "max": 5905,
      "num_unique_values": 11,
      "samples": [
        3896,
```

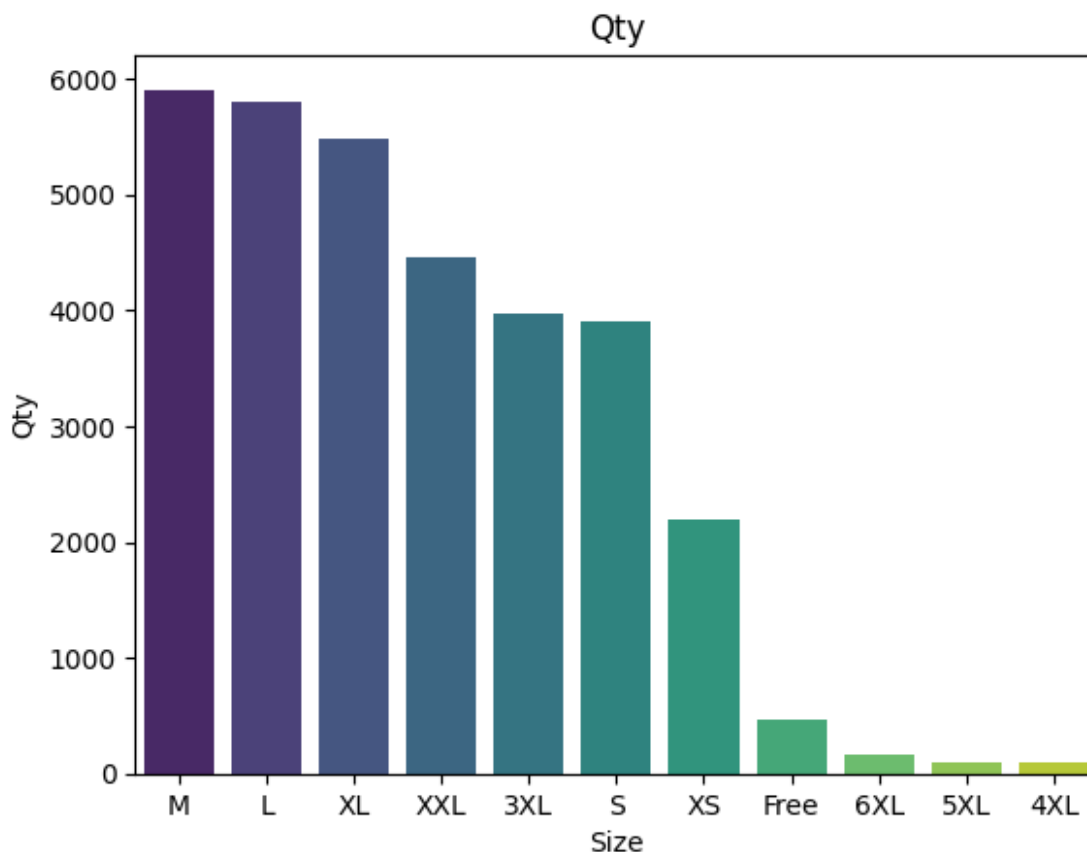
```
5905,\n          104\n          ],\n          \"semantic_type\": \"\", \n          \"description\": \"\" \n          }\n          }\n          ]\n          }\", \"type\": \"dataframe\"}
```

```
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, palette='viridis')
plt.xlabel('Size')
plt.ylabel('Qty')
plt.title('Qty')
plt.show()
```

<ipython-input-40-a9d82880199c>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

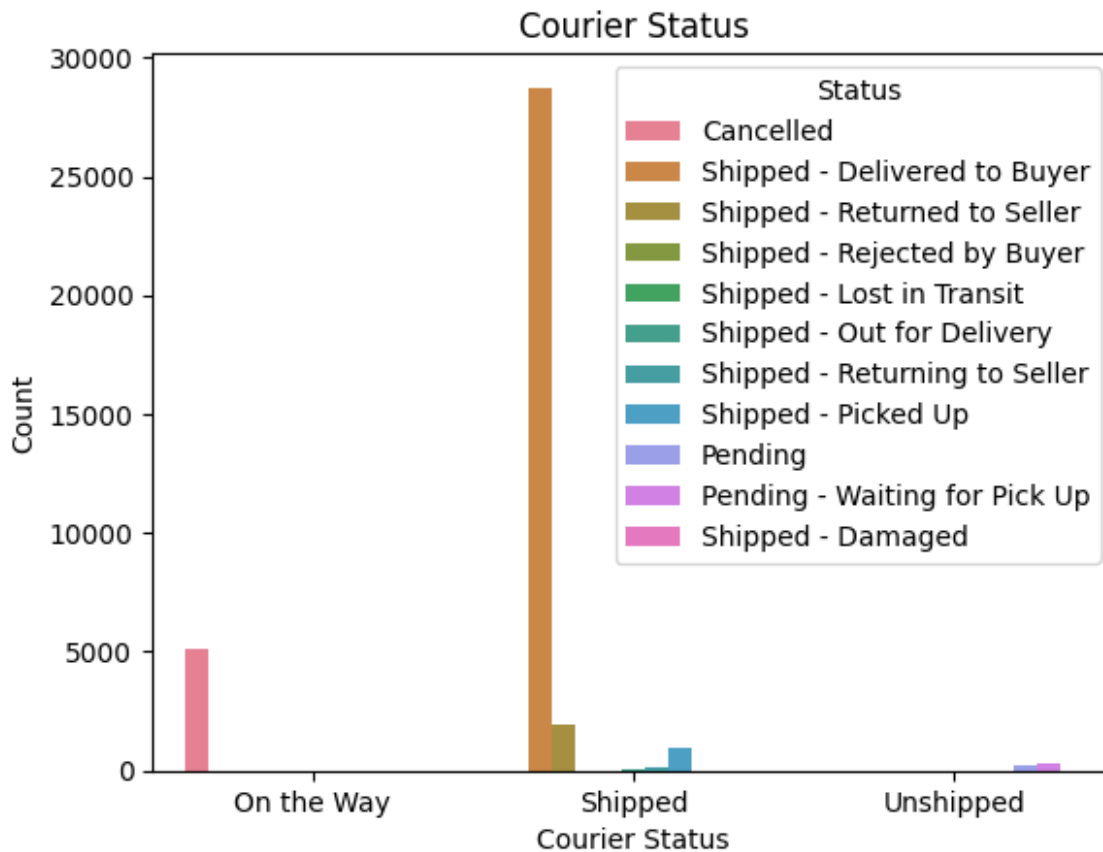
```
sns.barplot(x='Size', y='Qty', data=S_Qty, palette='viridis')
```



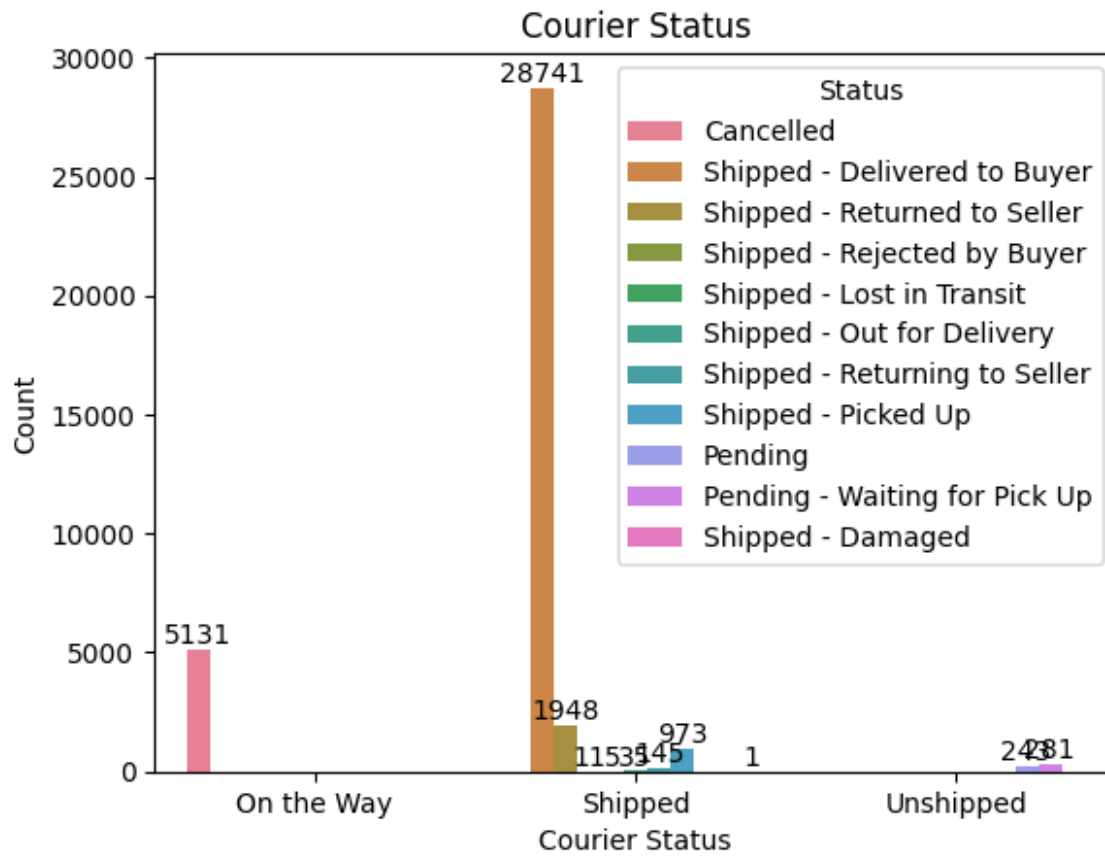
Note: From above Graph you can see that most of the Qty buys M-Size in the sales

Courier Status

```
sns.countplot(data=df, x='Courier Status', hue= 'Status')
plt.xlabel('Courier Status')
plt.ylabel('Count')
plt.title('Courier Status')
plt.show()
```

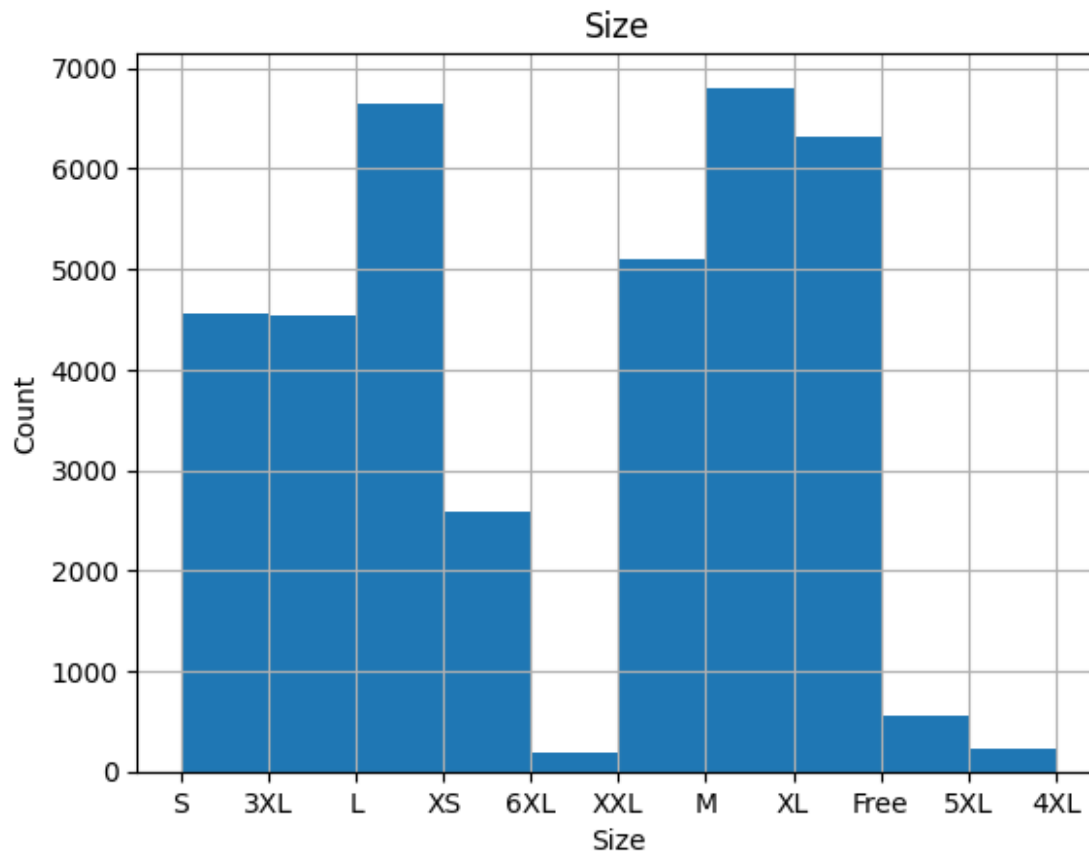


```
plt.figure()
ax=sns.countplot(data=df, x='Courier Status', hue= 'Status')
for bars in ax.containers:
    ax.bar_label(bars)
plt.xlabel('Courier Status')
plt.ylabel('Count')
plt.title('Courier Status')
plt.show()
```

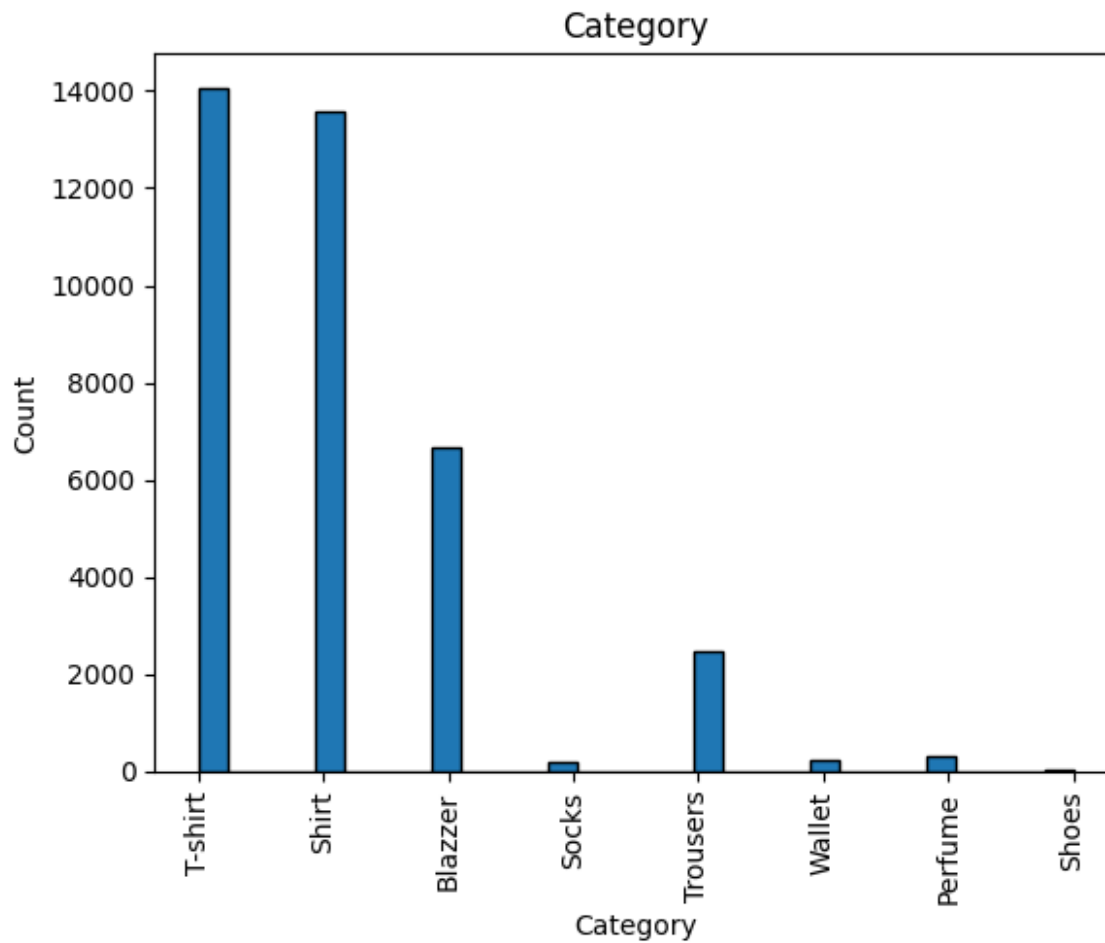


Note: From above Graph the majority of the orders are shipped through the courier.

```
#histogram
df['Size'].hist()
plt.xlabel('Size', color='black')
plt.ylabel('Count')
plt.title('Size')
plt.show()
```



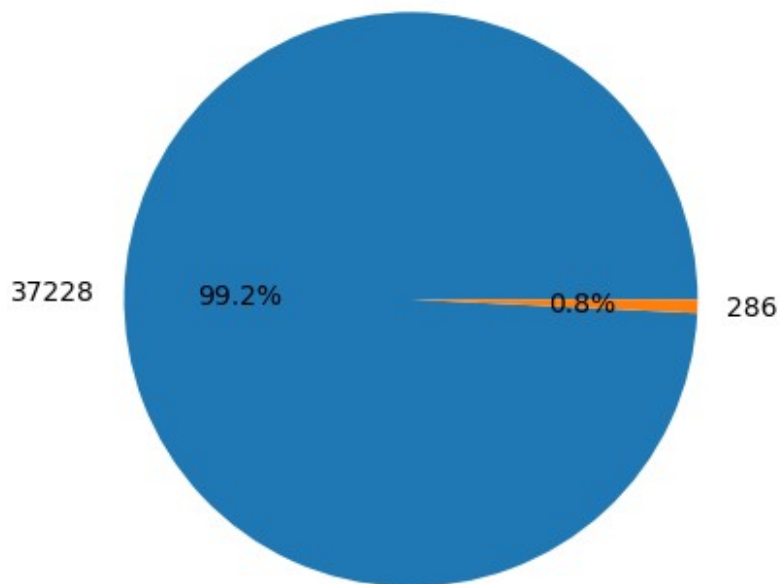
```
df['Category'] = df['Category'].astype(str)
column_data = df['Category']
plt.figure()
plt.hist(column_data, bins=30, edgecolor='Black')
plt.xticks(rotation=90)
plt.xlabel('Category')
plt.ylabel('Count')
plt.title('Category')
plt.show()
```



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

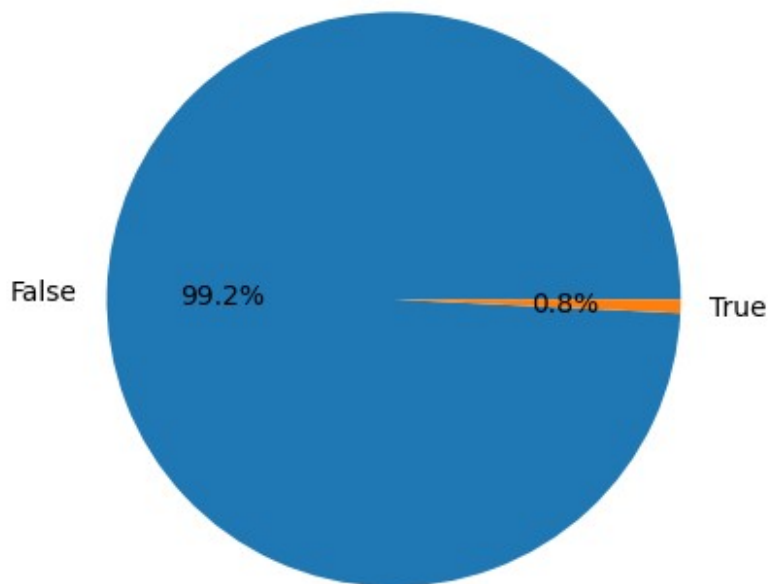
```
# Checking B2B Data by using pie chart
B2B_Check = df['B2B'].value_counts()

# 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()
```

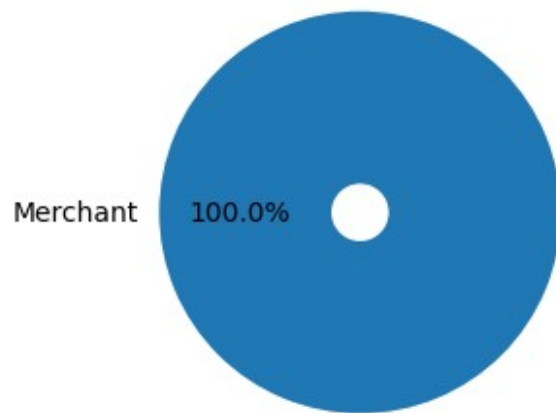
Note : From above chart we can see that maximum i.e. 99.3% of buyers are retailers and 0.7% are B2B buyers

```
# 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")

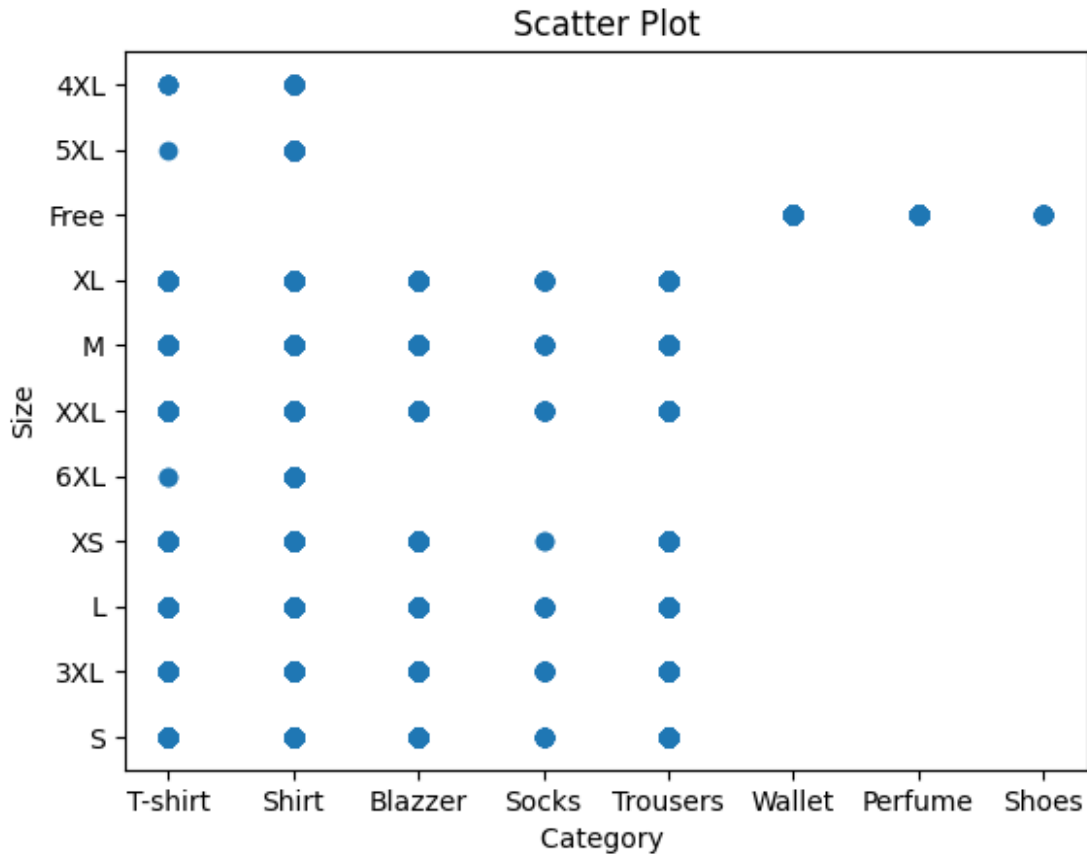
plt.show()
```



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', palette="viridis")
```

```
plt.xlabel('ship-state')
```

```
plt.ylabel('count')
```

```
plt.title('Distribution of State')
```

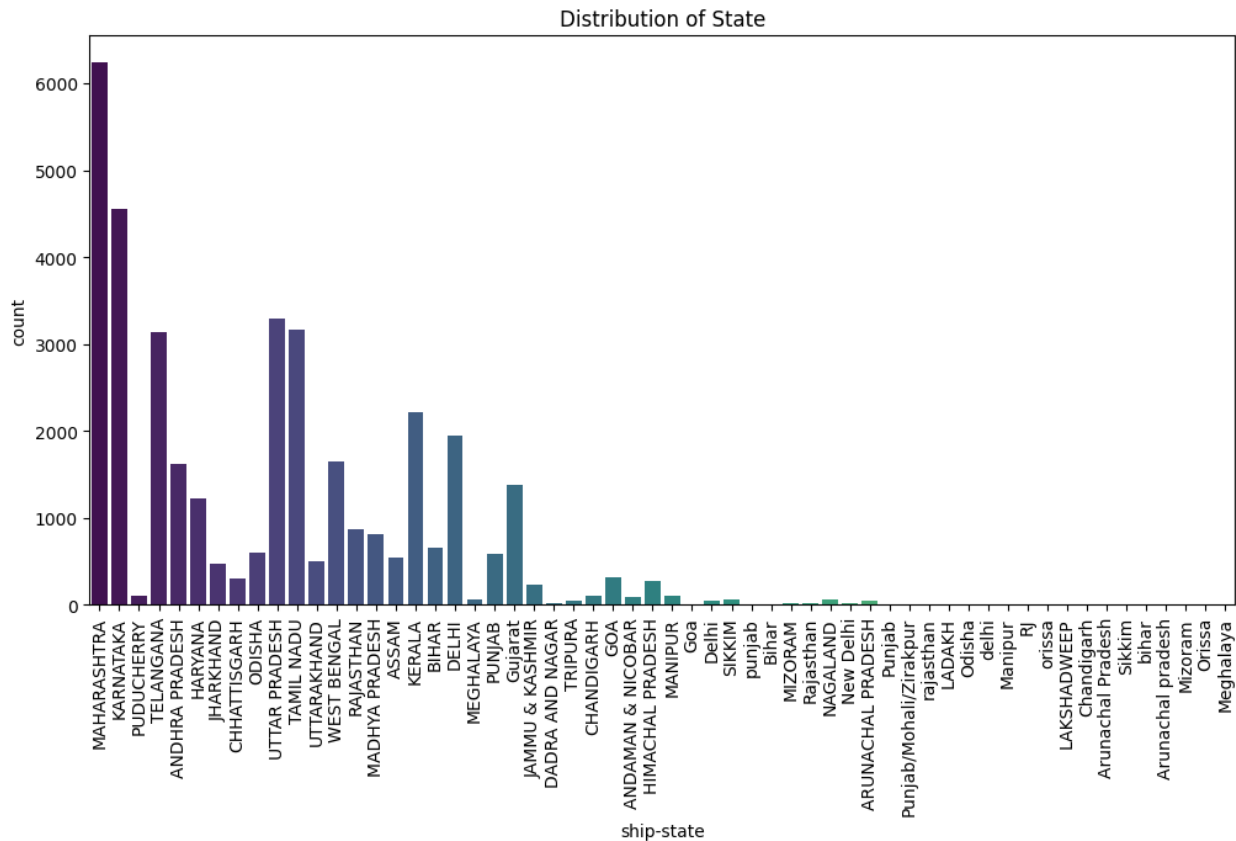
```
plt.xticks(rotation=90)
```

```
plt.show()
```

```
<ipython-input-75-5eee6ab98447>:3: FutureWarning:
```

```
Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.
```

```
sns.countplot(data=df, x='ship-state', palette="viridis")
```

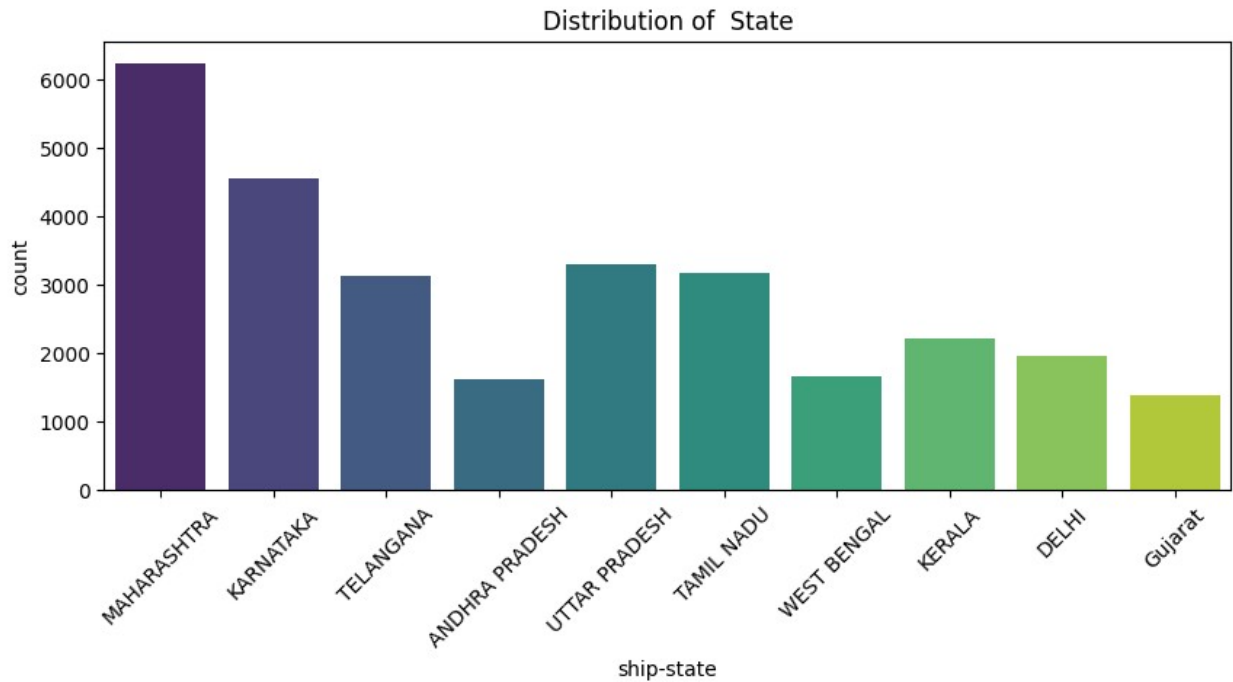


```
# top_10_States
top_10_state = df['ship-state'].value_counts().head(10)
# Plot count of cities by state
plt.figure(figsize=(10, 4))
sns.countplot(data=df[df['ship-state'].isin(top_10_state.index)],
x='ship-state', palette="viridis")
plt.xlabel('ship-state')
plt.ylabel('count')
plt.title('Distribution of State')
plt.xticks(rotation=45)
plt.show()
```

<ipython-input-76-a00c8f2c90f7>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.countplot(data=df[df['ship-state'].isin(top_10_state.index)],
x='ship-state', palette="viridis")
```



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.