amazon-data-analysis

October 30, 2024

1. IMPORTING LIBRARIES

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

2. IMPORT CSV FILES

```
[88]: df = pd.read_csv(r"H:\DA Python\Amazon Sales Analysis\Amazon Sale Report.
      ↪csv",encoding = 'unicode_escape')
```

3. DATA CLEANING

```
[89]: df.shape
```

```
[89]: (128976, 21)
```

```
[90]: df.head()
```

```
[90]:    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
```

1

```
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        NaN

[5 rows x 21 columns]
```

[91]: `df.tail()`

[91]:
```
              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]
```

[92]: `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
```

[93]: `#drop unrelated / blank columns`

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

[94]: `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
```

[95]: `#checking null value`
`pd.isnull(df)`

[95]:

	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	currency
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

	Amount	ship-city	ship-state	ship-postal-code	ship-country	B2B
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

         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]
```

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

```
[96]: 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
```

```
[97]: df.shape
```

```
[97]: (128976, 19)
```

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

```
[99]: df.shape
```

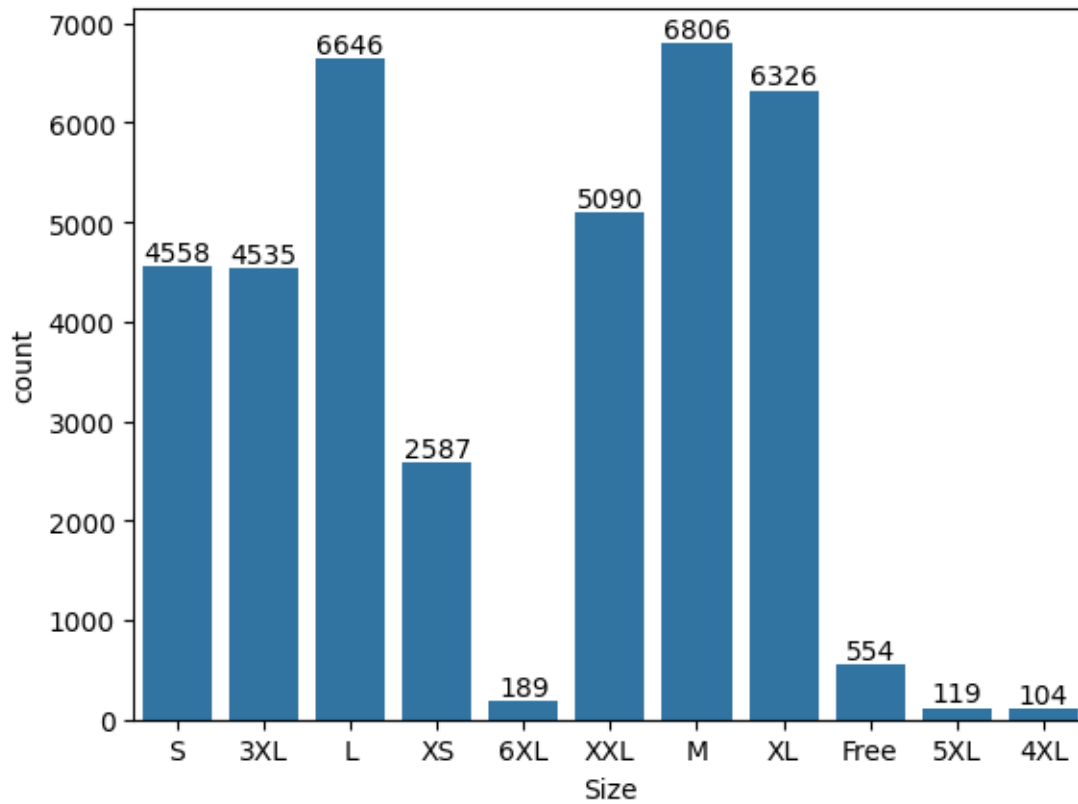```
[99]: (37514, 19)
```

```
[100]: df.columns
```

```
[100]: 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')
```

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

```
[102]: #checking data type
       df['ship-postal-code'].dtype
```

```
[102]: dtype('int64')
```

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

```
C:\Users\Ebad\AppData\Local\Temp\ipykernel_11744\3386729631.py: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'])
```

```
[104]: df['Date'].dtype
```

```
[104]: dtype('<M8[ns]')
```

4. Exploratory Data Analysis.

```
[105]: df.columns
```

```
[105]: 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

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

       for bars in ax.containers:
           ax.bar_label(bars)
```


Note: from above graph we can see that most of the people buy M-Size

Group By

```
[109]: df.groupby(['Size'], as_index=False)['Qty'].sum().
       ↪sort_values(by='Qty',ascending=False)
```
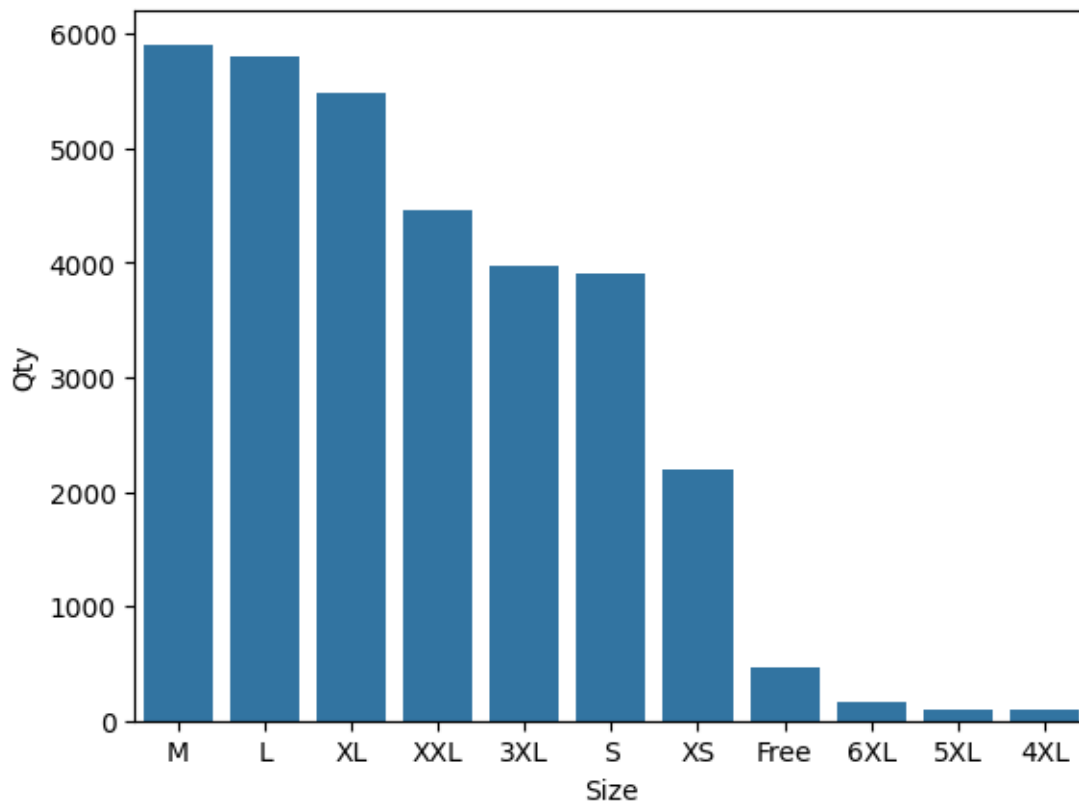
```
[109]:      Size   Qty
       6       M  5905
       5       L  5795
       8      XL  5481
       10    XXL  4465
       0     3XL  3972
       7       S  3896
       9      XS  2191
```

```
4    Free    467
3     6XL    170
2     5XL    104
1     4XL     93
```

[111]:
```python
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)
```
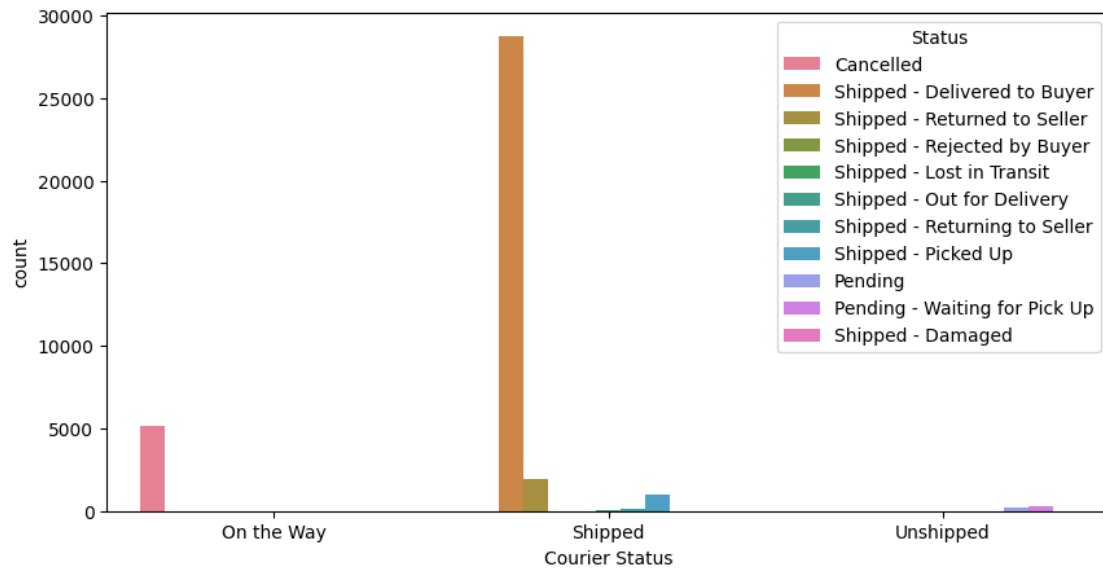
[111]:
```
<Axes: xlabel='Size', ylabel='Qty'>
```


Note: From the above graph we can see that most of the qty buys M_Size in the sale.

Courier Status

[116]:
```python
plt.figure(figsize=(10,5))

sns.countplot(data=df, x='Courier Status', hue='Status')

plt.show()
```
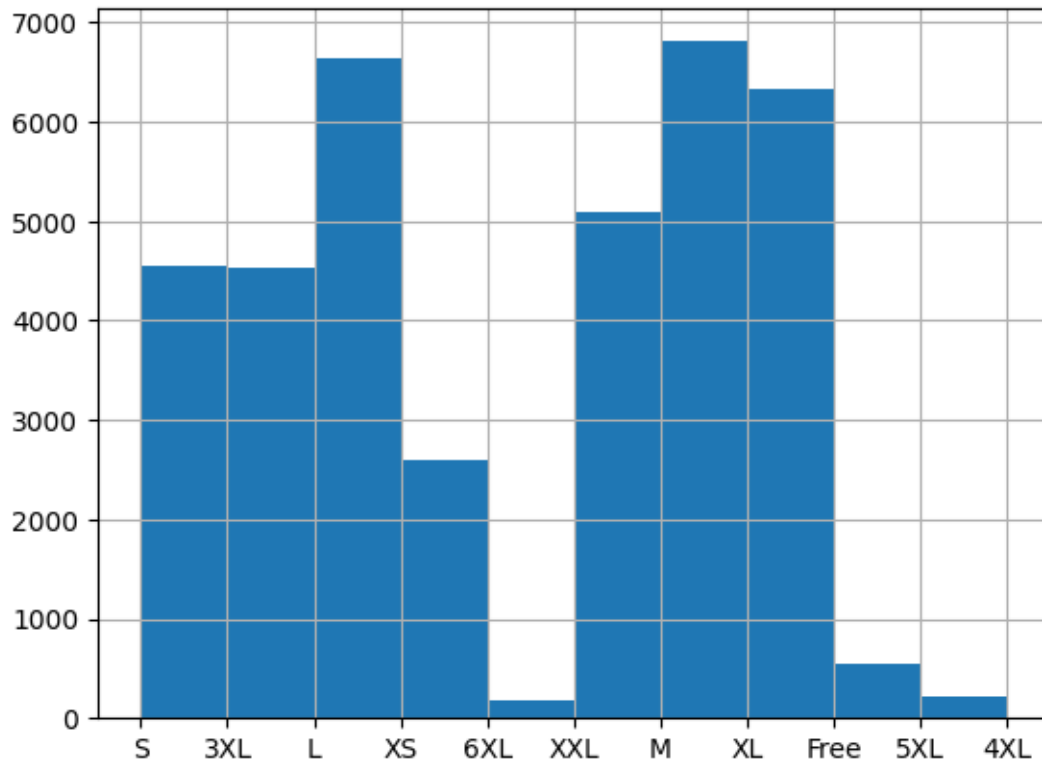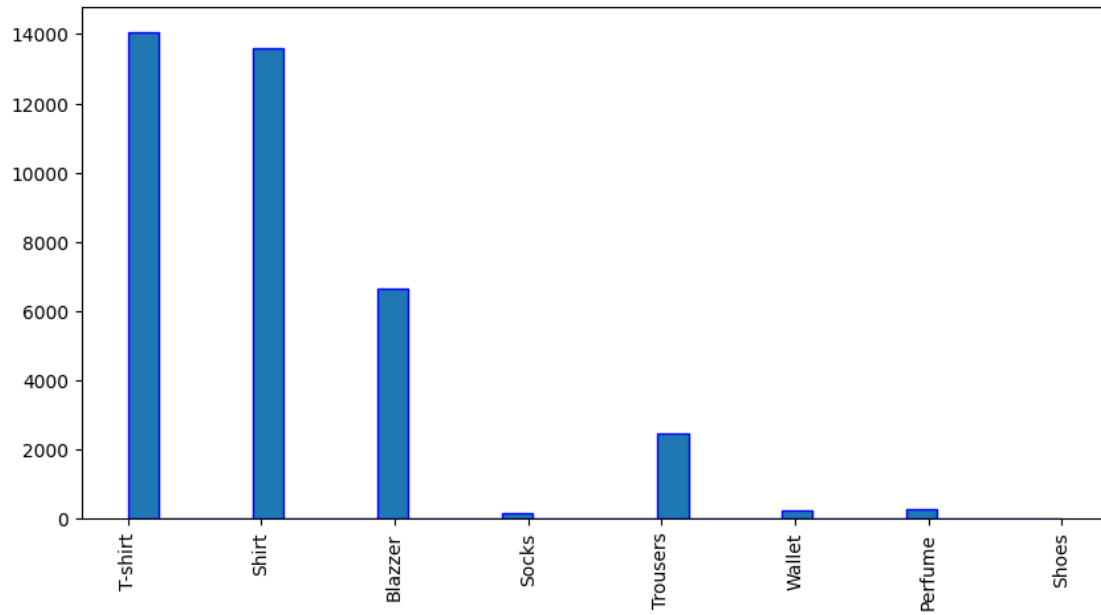
Note: From the above graph the majority of the orders are shipped through courier

```
[117]:  #histogram

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

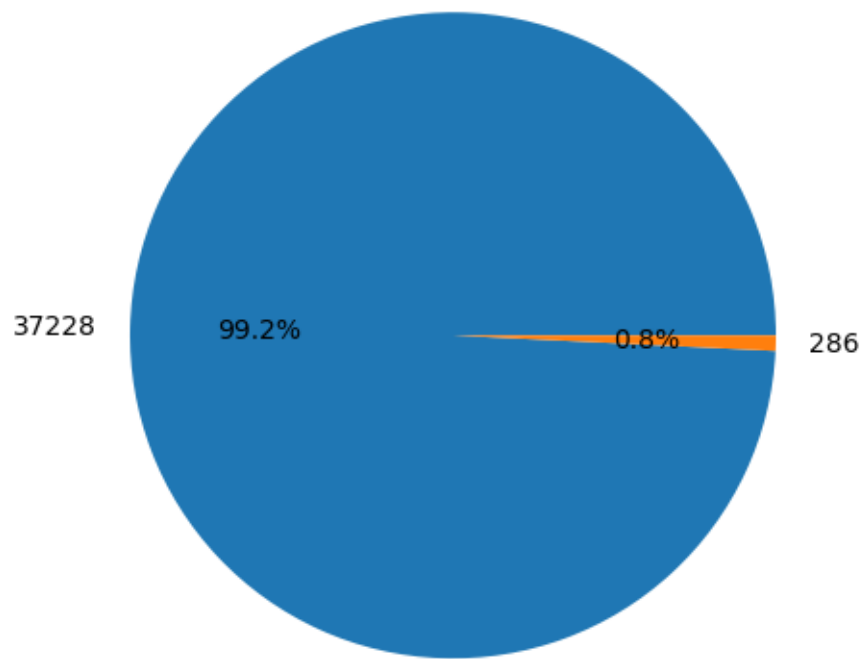[117]: <Axes: >

```
[123]: df['Category'] = df['Category'].astype(str)
       column_data = df['Category']
       plt.figure(figsize=(10,5))
       plt.hist(column_data, bins=30, edgecolor='Blue')
       plt.xticks(rotation=90)
       plt.show()
```
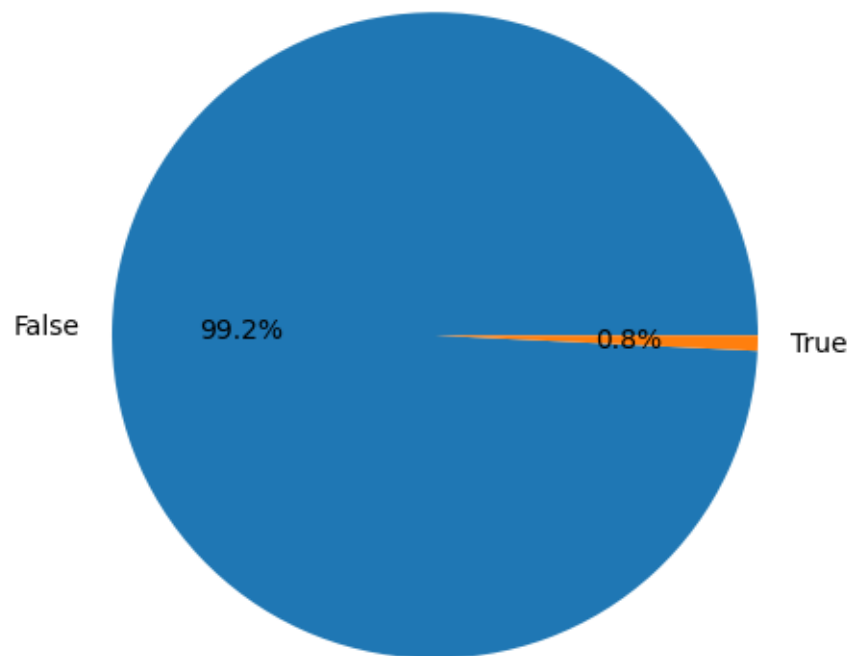
Note: From above graph we can se that most of the purchases are T shirt

```
[128]:  #Checking B2B data by usinig 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()
```

[129]:
```python
#Checking B2B data by usinig 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 the above chart we can see that maximum buyers are retailers 99.2% and 0.8 are B2B buyers

[130]:
```python
#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()
```

Scatter Plot

[136]:
```
#Plot count of cities by state

plt.figure(figsize=(20,5))
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()
```

Distribution of State


```
[138]: #top 10 states
       top_10_states = df['ship-state'].value_counts().head(10)

       #plot count of cities by state
       plt.figure(figsize=(15,5))
       sns.countplot(data=df[df['ship-state'].isin(top_10_states.index)],␣
         ↪x='ship-state')
       plt.xlabel('ship_state')
       plt.ylabel('count')
       plt.title('Distribution of State')
       plt.xticks(rotation=90)
       plt.show()
```

Distribution of State

Note: From the above graph we can see that most of the buyers are from Maharashtra state.

Conclusion

The data analysis reveals that the business has a significant base in maharashtra state, mainly serves retailers, fulfills orders through amazon, experience high demand for t shiort, and M sized as the preferred choice among buyers.