In [1]:

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

In [3]:

```
amazon=pd.read_csv(r"C:\Users\aks75\Downloads\dataset\e-commerce sales\Amazon Sale Report.csv")
amazon.head(5)
```

Out[3]:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Style	SKU	Category	...	Qty	currency	Amount	ship-ci
0	0	405-8078784-5731545	04-30-22	Cancelled	Merchant	Amazon.in	Standard	SET389	SET389-KR-NP-S	Set	...	0	INR	647.62	MUMB
1	1	171-9198151-1101146	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3781	JNE3781-KR-XXXL	kurta	...	1	INR	406.00	BENGALUR
2	2	404-0687676-7273146	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3371	JNE3371-KR-XL	kurta	...	1	INR	329.00	NAVI MUMB
3	3	403-9615377-8133951	04-30-22	Cancelled	Merchant	Amazon.in	Standard	J0341	J0341-DR-L	Western Dress	...	0	INR	753.33	PUDUCHERR
4	4	407-1069790-7240320	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3671	JNE3671-TU-XXXL	Top	...	1	INR	574.00	CHENN

5 rows × 23 columns

In [45]:

```
amazon.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 128975 entries, 0 to 128974
Data columns (total 23 columns):
 #   Column              Non-Null Count   Dtype  
---  ------              --------------   -----  
 0   index               128975 non-null  int64  
 1   Order ID            128975 non-null  object 
 2   Date                128975 non-null  object 
 3   Status              128975 non-null  object 
 4   Fulfilment          128975 non-null  object 
 5   Sales Channel       128975 non-null  object 
 6   ship-service-level  128975 non-null  object 
 7   Style               128975 non-null  object 
 8   SKU                 128975 non-null  object 
 9   Category            128975 non-null  object 
 10  Size                128975 non-null  object 
 11  ASIN                128975 non-null  object 
 12  Courier Status      122103 non-null  object 
 13  Qty                 128975 non-null  int64  
 14  currency            121180 non-null  object 
 15  Amount              121180 non-null  float64
 16  ship-city           128942 non-null  object 
 17  ship-state          128942 non-null  object 
 18  ship-postal-code    128942 non-null  float64
 19  ship-country        128942 non-null  object 
 20  promotion-ids       79822 non-null   object 
 21  B2B                 128975 non-null  bool   
 22  fulfilled-by        39277 non-null   object 
dtypes: bool(1), float64(2), int64(2), object(18)
memory usage: 21.8+ MB
```

There are total 128975 rows and 23 columns.

In [67]:

```
amazon["Date"]=pd.to_datetime(amazon["Date"])
print(amazon["Date"].dtypes)
```

```
datetime64[ns]
```

Date column has been changed to datetime datatype.

In [46]:

```
amazon.nunique()
```

Out[46]:

```
index                 128975
Order ID              120378
Date                      91
Status                    13
Fulfilment                 2
Sales Channel              2
ship-service-level         2
Style                   1377
SKU                     7195
Category                   9
Size                      11
ASIN                    7190
Courier Status             3
Qty                       10
currency                   1
Amount                  1410
ship-city               8955
ship-state                69
ship-postal-code        9459
ship-country               1
promotion-ids           5787
B2B                        2
fulfilled-by               1
dtype: int64
```

1- We can clearly see that columns like Fulfilment, sales Channel, ship-service-level, Courier Status are categorical columns and can be used for visualization.

Solving ship-state Multiple values.

In [7]:

```
print(amazon["ship-state"])
```

```
0          MAHARASHTRA
1            KARNATAKA
2          MAHARASHTRA
3           PUDUCHERRY
4           TAMIL NADU
              ...     
128970       TELANGANA
128971         HARYANA
128972       TELANGANA
128973         Gujarat
128974    CHHATTISGARH
Name: ship-state, Length: 128975, dtype: object
```

1. As we can see that some of the states name are in small letter, we need to change to all in capital letters.
2. Some of states name are mention in short-forms like PJ,RJ etc. This also needs to be treated.

In [47]:

```
amazon["ship-state"]=amazon["ship-state"].str.upper()
```

In [14]:

```
x=amazon["ship-state"].drop_duplicates()
x=pd.DataFrame(x).reset_index()

file_name = "x1.csv"  # Specify the file name

# Concatenate folder path and file name
file_path = folder_path + "\\" + file_name

# Export DataFrame to CSV
x.to_csv(file_path, index=True)
```

In [48]:

```
amazon.tail(5)
```

Out[48]:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Style	SKU	Category	...	Qty	currency	Amount	shi
128970	128970	406-6001380-7673107	05-31-22	Shipped	Amazon	Amazon.in	Expedited	JNE3697	JNE3697-KR-XL	kurta	...	1	INR	517.0	HYDER
128971	128971	402-9551604-7544318	05-31-22	Shipped	Amazon	Amazon.in	Expedited	SET401	SET401-KR-NP-M	Set	...	1	INR	999.0	GURU
128972	128972	407-9547469-3152358	05-31-22	Shipped	Amazon	Amazon.in	Expedited	J0157	J0157-DR-XXL	Western Dress	...	1	INR	690.0	HYDER
128973	128973	402-6184140-0545956	05-31-22	Shipped	Amazon	Amazon.in	Expedited	J0012	J0012-SKD-XS	Set	...	1	INR	1199.0	
128974	128974	408-7436540-8728312	05-31-22	Shipped	Amazon	Amazon.in	Expedited	J0003	J0003-SET-S	Set	...	1	INR	696.0	

5 rows × 23 columns

In [49]:
```
amazon["ship-state"]=amazon["ship-state"].replace({'RJ':'RAJASTHAN','NEW DELHI':'DELHI','RAJSHTHAN'
:'RAJASTHAN','RAJSTHAN':'RAJASTHAN','PB':'PUNJAB','PUNJAB/MOHALI/ZIRAKPUR':'PUNJAB','NL':'NAGALAND','AR':'ARUNA
'PONDICHERRY':'PUDUCHERRY','ORISSA':'ODISHA'})
```

In [50]:
```
amazon.nunique()
```

Out[50]:
```
index                 128975
Order ID              120378
Date                      91
Status                    13
Fulfilment                 2
Sales Channel              2
ship-service-level         2
Style                   1377
SKU                     7195
Category                   9
Size                      11
ASIN                    7190
Courier Status             3
Qty                       10
currency                   1
Amount                  1410
ship-city               8955
ship-state                37
ship-postal-code        9459
ship-country               1
promotion-ids           5787
B2B                        2
fulfilled-by               1
dtype: int64
```

Now our dataset have 37 unique ship-state values, which also includes union-territories. Therefore it seems correct.

Handling null values.

In [51]:
```
((amazon.isnull().sum()/len(amazon))*100).round(2)
```

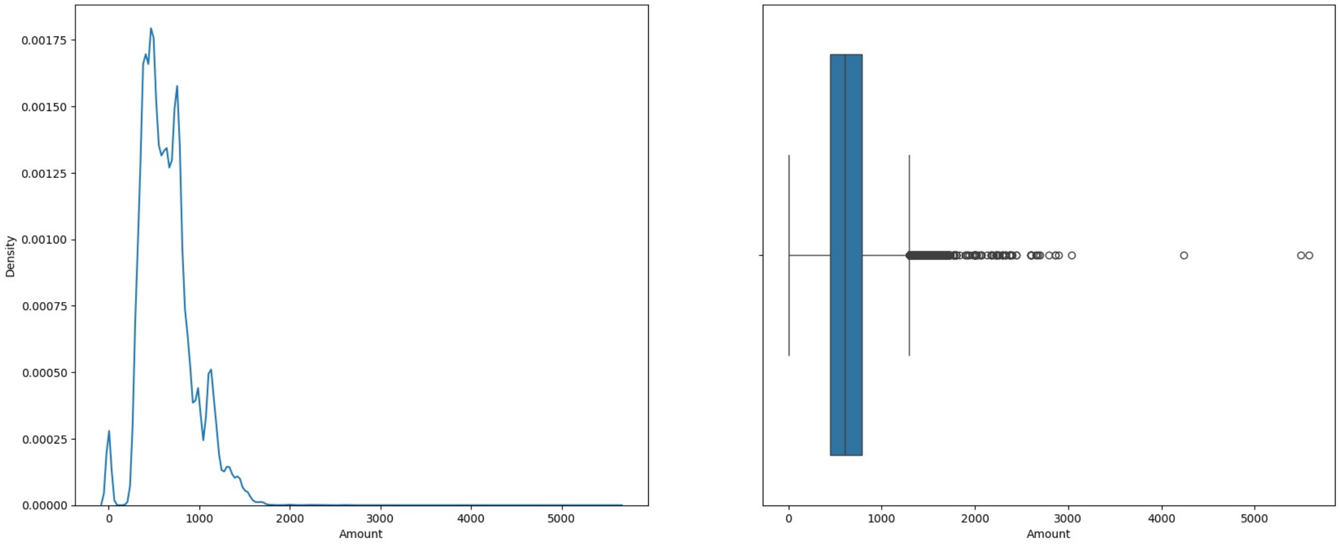
Out[51]:

```
index                  0.00
Order ID               0.00
Date                   0.00
Status                 0.00
Fulfilment             0.00
Sales Channel          0.00
ship-service-level     0.00
Style                  0.00
SKU                    0.00
Category               0.00
Size                   0.00
ASIN                   0.00
Courier Status         5.33
Qty                    0.00
currency               6.04
Amount                 6.04
ship-city              0.03
ship-state             0.03
ship-postal-code       0.03
ship-country           0.03
promotion-ids         38.11
B2B                    0.00
fulfilled-by          69.55
dtype: float64
```

1. Columns like Courier Status, currency, Amount, ship-city,state,country have null values.
2. Promotion-ids and fulfilled-by have large amount of null values.
3. Around 70% of values are null in fulfilled-by, so we will drop it.

In [52]:

```python
plt.figure(figsize=(20,8))
plt.subplot(1,2,1)
sns.kdeplot(data=amazon,x=amazon["Amount"])
plt.subplot(1,2,2)
sns.boxplot(data=amazon,x="Amount")
plt.show()
```



In [53]:

```python
x=amazon["Amount"].median()
x1=amazon["Amount"].mean()
print("The mean is ",x)
print("The median is ",round(x1,2))
```

```
The mean is  605.0
The median is  648.56
```

Replacing null values with mean and midian.

In [54]:

```python
amazon["Amount_median"]=amazon["Amount"].fillna(x)
amazon.head(5)
```

Out[54]:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Style	SKU	Category	...	currency	Amount	ship-city	
0	0	405-8078784-5731545	04-30-22	Cancelled	Merchant	Amazon.in	Standard	SET389	SET389-KR-NP-S	Set	...	INR	647.62	MUMBAI	M
1	1	171-9198151-1101146	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3781	JNE3781-KR-XXXL	kurta	...	INR	406.00	BENGALURU	
2	2	404-0687676-7273146	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3371	JNE3371-KR-XL	kurta	...	INR	329.00	NAVI MUMBAI	M
3	3	403-9615377-8133951	04-30-22	Cancelled	Merchant	Amazon.in	Standard	J0341	J0341-DR-L	Western Dress	...	INR	753.33	PUDUCHERRY	I
4	4	407-1069790-7240320	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3671	JNE3671-TU-XXXL	Top	...	INR	574.00	CHENNAI	

5 rows × 24 columns

In [55]:

```
amazon["Amount_mean"]=amazon["Amount"].fillna(x1)
amazon.head(5)
```

Out[55]:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Style	SKU	Category	...	Amount	ship-city	ship-s
0	0	405-8078784-5731545	04-30-22	Cancelled	Merchant	Amazon.in	Standard	SET389	SET389-KR-NP-S	Set	...	647.62	MUMBAI	MAHARASHT
1	1	171-9198151-1101146	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3781	JNE3781-KR-XXXL	kurta	...	406.00	BENGALURU	KARNATA
2	2	404-0687676-7273146	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3371	JNE3371-KR-XL	kurta	...	329.00	NAVI MUMBAI	MAHARASHT
3	3	403-9615377-8133951	04-30-22	Cancelled	Merchant	Amazon.in	Standard	J0341	J0341-DR-L	Western Dress	...	753.33	PUDUCHERRY	PUDUCHEF
4	4	407-1069790-7240320	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3671	JNE3671-TU-XXXL	Top	...	574.00	CHENNAI	TAMIL NA

5 rows × 25 columns

In [56]:

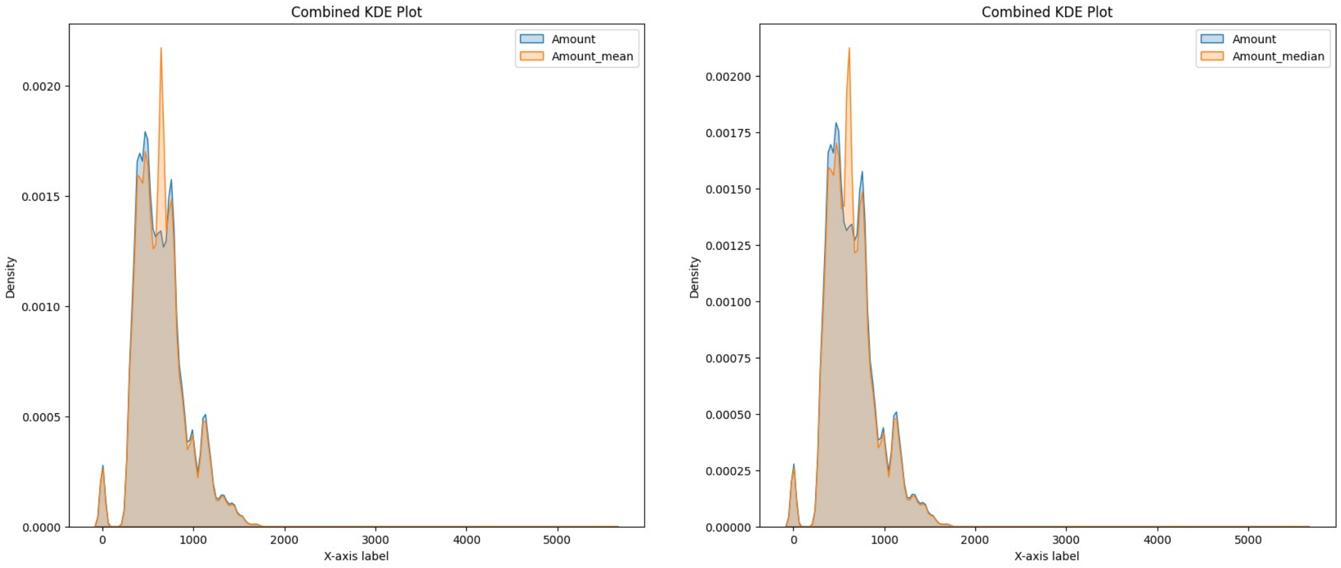
```
plt.figure(figsize=(20,8))
plt.subplot(1,2,1)
sns.kdeplot(data=amazon,x=amazon["Amount"], fill=True, label='Amount')

# Plot KDE for data2 on the same axis
sns.kdeplot(data=amazon,x=amazon["Amount_mean"], fill=True, label='Amount_mean')
# Add labels and title
plt.xlabel('X-axis label')
plt.ylabel('Density')
plt.title('Combined KDE Plot')

# Show legend
plt.legend()
plt.subplot(1,2,2)
sns.kdeplot(data=amazon,x=amazon["Amount"], fill=True, label='Amount')
# Plot KDE for data2 on the same axis
sns.kdeplot(data=amazon,x=amazon["Amount_median"], fill=True, label='Amount_median')
# Add labels and title
plt.xlabel('X-axis label')
plt.ylabel('Density')
plt.title('Combined KDE Plot')
```

```
# Show legend
plt.legend()

# Show plot
plt.show()
```



1. The kdeplot for both mean and median replaced null values differ from original one which null values.
2. So, we cannot replace with it. will try it forward fill.

Filling null values using forward fill.

In [57]:

```
amazon["Amount_ffill"]=amazon["Amount"].fillna(method="ffill")
```

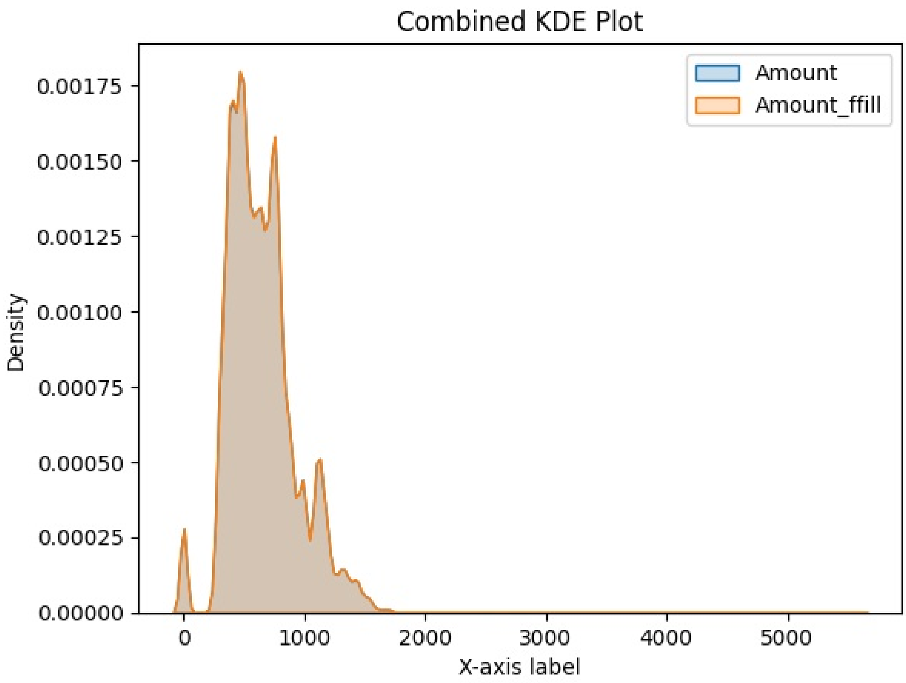
In [58]:

```
sns.kdeplot(data=amazon,x=amazon["Amount"], fill=True, label='Amount')
sns.kdeplot(data=amazon,x=amazon["Amount_ffill"], fill=True, label='Amount_ffill')

# Add labels and title
plt.xlabel('X-axis label')
plt.ylabel('Density')
plt.title('Combined KDE Plot')

# Show legend
plt.legend()

# Show plot
plt.show()
```



The kdeplot is totally overlapping each other, the one with null values and one without. So, this technique works.

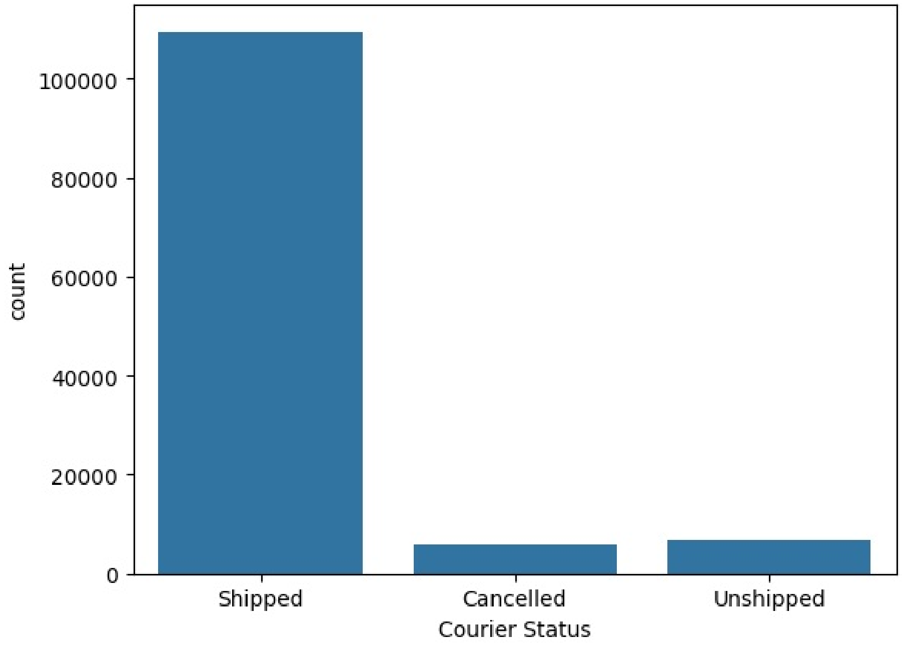
In [59]:

```
print("var of amount is",round((amazon["Amount"].var()),2))
print("var of amount is",round((amazon["Amount_ffill"].var()),2))
```

```
var of amount is 79080.01
var of amount is 79000.99
```

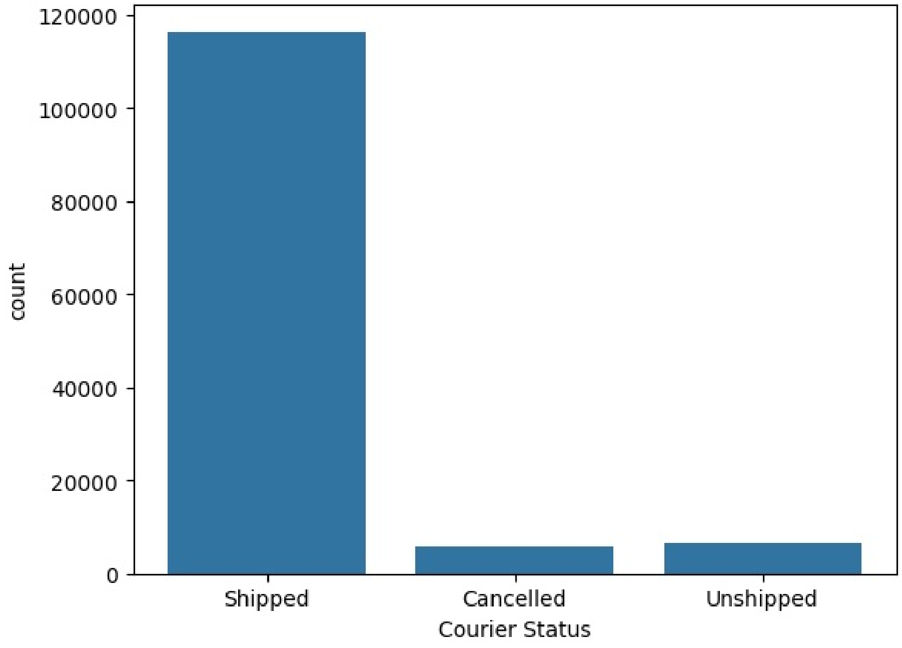
Variance is also nearly same, therefore this method is acceptable.

In [60]:
```
sns.countplot(data=amazon,x="Courier Status")
plt.show()
```



In [61]:
```
amazon["Courier Status"]=amazon["Courier Status"].fillna("Shipped")
```

In [25]:
```
sns.countplot(data=amazon,x="Courier Status")
plt.plot()
plt.show()
```



In [76]:
```
amazon.dropna(subset=['ship-state'],inplace=True)
```

In [74]:
```
amazon['promotion-ids'].fillna("Not Available", inplace=True)
```

In [77]:
```
amazon['currency'].fillna("INR", inplace=True)
```

In [ ]:
```
amazon.drop(columns=["fulfilled-by","Amount_median","Amount_mean","Amount"],inplace=True)
```

In [28]:
```
amazon.drop(columns="fulfilled-by",inplace=True)
```

In [79]:
```
amazon.rename(columns={"Amount_ffill":"Amount"},inplace=True)
amazon.head(3)
```

Out[79]:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Style	SKU	Category	...	Courier Status	Qty	currency	ship-city
0	0	405-8078784-5731545	2022-04-30	Cancelled	Merchant	Amazon.in	Standard	SET389	SET389-KR-NP-S	Set	...	Shipped	0	INR	MUMBA
1	1	171-9198151-1101146	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3781	JNE3781-KR-XXXL	kurta	...	Shipped	1	INR	BENGALURU
2	2	404-0687676-7273146	2022-04-30	Shipped	Amazon	Amazon.in	Expedited	JNE3371	JNE3371-KR-XL	kurta	...	Shipped	1	INR	NAV MUMBA

3 rows × 22 columns

1. Couririer-status null-values are filled with the mode.
2. Null-values of Promotion-ids is replaced with "Not Available".
3. Column Fulfilled-by is dropped.
4. For the same rows, ship-state,ship-city,postal-code,ship-country is missing. As, no information is given, we dropped all 3 rows.

In [64]:

```
amazon['currency'].value_counts()
```

Out[64]:

```
INR    121180
Name: currency, dtype: int64
```

In [80]:

```
((amazon.isnull().sum()/len(amazon))*100).round(2)
```

Out[80]:

```
index                 0.0
Order ID              0.0
Date                  0.0
Status                0.0
Fulfilment            0.0
Sales Channel         0.0
ship-service-level    0.0
Style                 0.0
SKU                   0.0
Category              0.0
Size                  0.0
ASIN                  0.0
Courier Status        0.0
Qty                   0.0
currency              0.0
ship-city             0.0
ship-state            0.0
ship-postal-code      0.0
ship-country          0.0
promotion-ids         0.0
B2B                   0.0
Amount                0.0
dtype: float64
```

Transferring our data into csv file.

In [84]:

```
amazon_updated=amazon.copy()
filename="amazon_updated1.xlsx"
folder_path = r"C:\Users\aks75\Downloads"
file_path = folder_path + "\\" + filename
amazon_updated.to_csv(file_path)
```

Finally, our updated data isready for visualization.

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js