zb8sqfqwx

December 4, 2024

1 InnoByte Services - Data Analyst Internship Project

1.1 Problem Statement:

- Analyze and Provide Insights on Amazon Sales Report.

1.2 Problem Description:

- The provided dataset contains information about sales transactions on Amazon, including details such as order ID, date, status, fulfilment method, sales channel, product category, size, quantity, amount, shipping details, and more. The objective is to conduct a comprehensive analysis of the data and extract actionable insights to support business decision-making.

1.3 Import libraries

[ ]:
```
import pandas as pd
import numpy as np
import warnings
import re
import matplotlib.pyplot as plt
import seaborn as sns
warnings.filterwarnings("ignore")
```

1.4 Read the dataset

[ ]:
```
amazons1 = pd.read_csv('/content/Amazon Sale Report.csv')
amazons1
```

[ ]:

	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	
…	…	…	…	…	
128971	128970	406-6001380-7673107	05-31-22	Shipped	
128972	128971	402-9551604-7544318	05-31-22	Shipped	
128973	128972	407-9547469-3152358	05-31-22	Shipped	

```
128974  128973  402-6184140-0545956  05-31-22                      Shipped
128975  128974  408-7436540-8728312  05-31-22                      Shipped

       Fulfilment Sales Channel ship-service-level  Category Size  \
0        Merchant     Amazon.in           Standard   T-shirt    S
1        Merchant     Amazon.in           Standard     Shirt  3XL
2          Amazon     Amazon.in          Expedited     Shirt   XL
3        Merchant     Amazon.in           Standard   Blazzer    L
4          Amazon     Amazon.in          Expedited  Trousers  3XL
…           …           …                 …       …  …
128971     Amazon     Amazon.in          Expedited     Shirt   XL
128972     Amazon     Amazon.in          Expedited   T-shirt    M
128973     Amazon     Amazon.in          Expedited   Blazzer  XXL
128974     Amazon     Amazon.in          Expedited   T-shirt   XS
128975     Amazon     Amazon.in          Expedited   T-shirt    S

       Courier Status  …  currency   Amount    ship-city    ship-state  \
0          On the Way  …       INR   647.62       MUMBAI   MAHARASHTRA
1             Shipped  …       INR   406.00    BENGALURU     KARNATAKA
2             Shipped  …       INR   329.00  NAVI MUMBAI   MAHARASHTRA
3          On the Way  …       INR   753.33   PUDUCHERRY    PUDUCHERRY
4             Shipped  …       INR   574.00      CHENNAI    TAMIL NADU
…               …  …       …       …           …            …
128971        Shipped  …       INR   517.00    HYDERABAD     TELANGANA
128972        Shipped  …       INR   999.00     GURUGRAM       HARYANA
128973        Shipped  …       INR   690.00    HYDERABAD     TELANGANA
128974        Shipped  …       INR  1199.00        Halol       Gujarat
128975        Shipped  …       INR   696.00       Raipur  CHHATTISGARH

       ship-postal-code  ship-country    B2B  fulfilled-by New  PendingS
0              400081.0            IN  False     Easy Ship NaN       NaN
1              560085.0            IN  False     Easy Ship NaN       NaN
2              410210.0            IN   True           NaN NaN       NaN
3              605008.0            IN  False     Easy Ship NaN       NaN
4              600073.0            IN  False           NaN NaN       NaN
…                 …           …    …           …  ..       …
128971         500013.0            IN  False           NaN NaN       NaN
128972         122004.0            IN  False           NaN NaN       NaN
128973         500049.0            IN  False           NaN NaN       NaN
128974         389350.0            IN  False           NaN NaN       NaN
128975         492014.0            IN  False           NaN NaN       NaN

[128976 rows x 21 columns]
```

1.5 Check No. Of Rows and Columns

[ ]:
```
amazons1.shape
```

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

- There are 128976 rows and 21 columns.

1.6 Data Checks to Perform

- Check missing values
- Check Duplicated values
- Check Data type
- Check the number of unique values of each columns
- Check Statistics of Dataset
- Check various categories Present in the different Categories Columns

[ ]:
```
## check missing values
amazons1.isnull().sum()
```

[ ]:
```
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
New                128976
PendingS           128976
dtype: int64
```

- There are many missing values in the dataset.
- Some Columns that have missing values are 'currency', 'Amount', 'ship-city', 'ship-state', 'ship-postal-code', 'ship-country', 'fulfilled-by', 'New', 'PendingS'.

[ ]:
```
## check duplicated values
amazons1.duplicated().sum()
```

[ ]:
```
168
```

- There are 168 duplicated records in the dataset.

[ ]:
```
## check the datatype
amazons1.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
```

- There are 1 column that have boolean data type, 4 columns float data type, 2 columns integer data type and 14 columns string data type.

[ ]:
```
## check number of unique values in each columns
amazons1.nunique()
```

[ ]:
```
index               128808
Order ID            120229
Date                    91
Status                  13
Fulfilment               2
Sales Channel            2
```

```
ship-service-level         2
Category                   9
Size                      11
Courier Status             4
Qty                       10
currency                   1
Amount                  1408
ship-city               8948
ship-state                69
ship-postal-code        9454
ship-country               1
B2B                        2
fulfilled-by               1
New                        0
PendingS                   0
dtype: int64
```

[ ]:
```
## check the Statistical summary of dataset
amazons1.describe()
```

[ ]:
```
                index            Qty         Amount  ship-postal-code  New  \
count  128976.000000  128976.000000  121176.000000     128941.000000  0.0
mean    64486.130427       0.904401     648.562176     463945.677744  NaN
std     37232.897832       0.313368     281.185041     191458.488954  NaN
min         0.000000       0.000000       0.000000     110001.000000  NaN
25%     32242.750000       1.000000     449.000000     382421.000000  NaN
50%     64486.500000       1.000000     605.000000     500033.000000  NaN
75%     96730.250000       1.000000     788.000000     600024.000000  NaN
max    128974.000000      15.000000    5584.000000     989898.000000  NaN

       PendingS
count       0.0
mean        NaN
std         NaN
min         NaN
25%         NaN
50%         NaN
75%         NaN
max         NaN
```

- Index: Uniformly distributed from 0 to 128,974, indicating a sequential identifier.
- Qty (Quantity): Highly concentrated around the value 1, with occasional higher values up to 15, showing a skewed distribution towards lower quantities.
- Amount: Exhibits significant variability, with most values below 788 but some outliers reaching up to 5,584, indicating a right-skewed distribution.
- Ship Postal Code: Spans a wide range from 110,001 to 989,898, suggesting the dataset covers a broad geographic area.

- New and PendingS: Both columns have no data, indicating they might be empty or not processed properly.

1.7 Data Cleaning

- The best Practice before to clean the dataset make a another copy of the dataset. Because If a mistake is made during the cleaning process, you can easily revert to the original dataset without having to reload or re-fetch the data.

[ ]:
```
amazons1_copy = amazons1.copy()
```

[ ]:
```
amazons1_copy.head(2)
```

[ ]:
```
   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

  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

   …  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

   ship-country    B2B  fulfilled-by New  PendingS
0            IN  False     Easy Ship NaN       NaN
1            IN  False     Easy Ship NaN       NaN

[2 rows x 21 columns]
```

[ ]:
```
amazons1_copy.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
```

- Firstly, remove the columns that have no data.
- Remove the index column also because they have no any use

[ ]:
```
amazons1_copy.drop(['New', 'PendingS','index'], axis = 1, inplace = True)
```

[ ]:
```
amazons1_copy
```

[ ]:
```
                   Order ID      Date                        Status  \
0       405-8078784-5731545  04-30-22                     Cancelled
1       171-9198151-1101146  04-30-22  Shipped - Delivered to Buyer
2       404-0687676-7273146  04-30-22                       Shipped
3       403-9615377-8133951  04-30-22                     Cancelled
4       407-1069790-7240320  04-30-22                       Shipped
…                       …       …                           …
128971  406-6001380-7673107  05-31-22                       Shipped
128972  402-9551604-7544318  05-31-22                       Shipped
128973  407-9547469-3152358  05-31-22                       Shipped
128974  402-6184140-0545956  05-31-22                       Shipped
128975  408-7436540-8728312  05-31-22                       Shipped

       Fulfilment Sales Channel ship-service-level  Category Size  \
0        Merchant     Amazon.in           Standard   T-shirt    S
1        Merchant     Amazon.in           Standard     Shirt  3XL
2          Amazon     Amazon.in          Expedited     Shirt   XL
3        Merchant     Amazon.in           Standard   Blazzer    L
4          Amazon     Amazon.in          Expedited  Trousers  3XL
…           …           …                 …       …  …
128971     Amazon     Amazon.in          Expedited     Shirt   XL
128972     Amazon     Amazon.in          Expedited   T-shirt    M
128973     Amazon     Amazon.in          Expedited   Blazzer  XXL
128974     Amazon     Amazon.in          Expedited   T-shirt   XS
128975     Amazon     Amazon.in          Expedited   T-shirt    S

       Courier Status  Qty currency   Amount    ship-city    ship-state  \
```

```
0          On the Way    0      INR   647.62       MUMBAI   MAHARASHTRA
1             Shipped    1      INR   406.00    BENGALURU    KARNATAKA
2             Shipped    1      INR   329.00  NAVI MUMBAI   MAHARASHTRA
3          On the Way    0      INR   753.33   PUDUCHERRY    PUDUCHERRY
4             Shipped    1      INR   574.00      CHENNAI    TAMIL NADU
…                …  …     …       …          …           …
128971        Shipped    1      INR   517.00    HYDERABAD     TELANGANA
128972        Shipped    1      INR   999.00     GURUGRAM       HARYANA
128973        Shipped    1      INR   690.00    HYDERABAD     TELANGANA
128974        Shipped    1      INR  1199.00        Halol       Gujarat
128975        Shipped    1      INR   696.00       Raipur  CHHATTISGARH

        ship-postal-code ship-country    B2B fulfilled-by
0               400081.0           IN  False    Easy Ship
1               560085.0           IN  False    Easy Ship
2               410210.0           IN   True          NaN
3               605008.0           IN  False    Easy Ship
4               600073.0           IN  False          NaN
…                  …          …     …          …
128971          500013.0           IN  False          NaN
128972          122004.0           IN  False          NaN
128973          500049.0           IN  False          NaN
128974          389350.0           IN  False          NaN
128975          492014.0           IN  False          NaN

[128976 rows x 18 columns]
```

- Remove the duplicated data

[ ]:
```
##check the duplicated records
amazons1_copy.duplicated().sum()
```

[ ]:
```
959
```

- After drop the some columns we achieve 959 records that have duplicate.

[ ]:
```
amazons1_copy = amazons1_copy.drop_duplicates(subset='Order ID', keep='first')
```

[ ]:
```
amazons1_copy.duplicated().sum()
```

[ ]:
```
0
```

- There is no duplicated records.

[ ]:
```
amazons1_copy.shape
```

[ ]:
```
(120229, 18)
```

- After removing the duplicated records there are 120229 rows and 18 columns.

1.8 Finding Some errors and Remove it by Checking each Columns in the Dataset.

[ ]:
```
amazons1_copy.Date.dtypes
```

[ ]:
```
dtype('O')
```

- The date column is object type. convert into datetime format.

[ ]:
```
amazons1_copy['Date'] = pd.to_datetime(amazons1_copy['Date'])
```

[ ]:
```
amazons1_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 120229 entries, 0 to 128975
Data columns (total 18 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   Order ID            120229 non-null  object
 1   Date                120229 non-null  datetime64[ns]
 2   Status              120229 non-null  object
 3   Fulfilment          120229 non-null  object
 4   Sales Channel       120229 non-null  object
 5   ship-service-level  120229 non-null  object
 6   Category            120229 non-null  object
 7   Size                120229 non-null  object
 8   Courier Status      120229 non-null  object
 9   Qty                 120229 non-null  int64
 10  currency            112834 non-null  object
 11  Amount              112834 non-null  float64
 12  ship-city           120201 non-null  object
 13  ship-state          120201 non-null  object
 14  ship-postal-code    120201 non-null  float64
 15  ship-country        120201 non-null  object
 16  B2B                 120229 non-null  bool
 17  fulfilled-by        36323 non-null   object
dtypes: bool(1), datetime64[ns](1), float64(2), int64(1), object(13)
memory usage: 16.6+ MB
```

[ ]:
```
amazons1_copy['Status'].unique()
```

[ ]:
```
array(['Cancelled', 'Shipped - Delivered to Buyer', 'Shipped',
       'Shipped - Returned to Seller', 'Shipped - Rejected by Buyer',
       'Shipped - Lost in Transit', 'Shipped - Out for Delivery',
       'Shipped - Returning to Seller', 'Shipped - Picked Up', 'Pending',
       'Pending - Waiting for Pick Up', 'Shipped - Damaged', 'Shipping'],
```

```
       dtype=object)
```

[ ]:
```
amazons1_copy['status'] = amazons1_copy['Status'].str.split('-').str[0]
amazons1_copy['order status'] = amazons1_copy['Status'].str.split('-').str[1]
```

[ ]:
```
amazons1_copy['status'].unique()
```

[ ]:
```
array(['Cancelled', 'Shipped ', 'Shipped', 'Pending', 'Pending ',
       'Shipping'], dtype=object)
```

[ ]:
```
amazons1_copy['order status'].unique()
```

[ ]:
```
array([nan, ' Delivered to Buyer', ' Returned to Seller',
       ' Rejected by Buyer', ' Lost in Transit', ' Out for Delivery',
       ' Returning to Seller', ' Picked Up', ' Waiting for Pick Up',
       ' Damaged'], dtype=object)
```

[ ]:
```
amazons1_copy['order status'].isnull().sum()
```

[ ]:
```
90500
```

[ ]:
```
amazons1_copy['order status'] = amazons1_copy['order status'].replace(np.nan,␣
↪"loading")
```

[ ]:
```
amazons1_copy['order status'].unique()
```

[ ]:
```
array(['loading', ' Delivered to Buyer', ' Returned to Seller',
       ' Rejected by Buyer', ' Lost in Transit', ' Out for Delivery',
       ' Returning to Seller', ' Picked Up', ' Waiting for Pick Up',
       ' Damaged'], dtype=object)
```

[ ]:
```
amazons1_copy['order status'].isnull().sum()
```

[ ]:
```
0
```

[ ]:
```
amazons1_copy.head(2)
```

[ ]:
```
              Order ID       Date                        Status Fulfilment  \
0  405-8078784-5731545 2022-04-30                     Cancelled   Merchant
1  171-9198151-1101146 2022-04-30  Shipped - Delivered to Buyer   Merchant

  Sales Channel ship-service-level Category Size Courier Status  Qty currency  \
0     Amazon.in           Standard  T-shirt    S     On the Way    0      INR
1     Amazon.in           Standard    Shirt  3XL        Shipped    1      INR

   Amount  ship-city   ship-state  ship-postal-code ship-country    B2B  \
0  647.62     MUMBAI  MAHARASHTRA          400081.0           IN  False
```

```
1  406.00  BENGALURU    KARNATAKA          560085.0           IN  False

  fulfilled-by     status         order status
0    Easy Ship  Cancelled              loading
1    Easy Ship   Shipped    Delivered to Buyer
```

[ ]:
```
## handle the Fulfilment column
amazons1_copy['Fulfilment'].unique()
```

[ ]:
```
array(['Merchant', 'Amazon'], dtype=object)
```

- There is no any error in the Fulfilment column.

[ ]:
```
## handle the Sales Channel
amazons1_copy['Sales Channel'].unique()
```

[ ]:
```
array(['Amazon.in', 'Non-Amazon'], dtype=object)
```

- There is one error in the 'Sales Channel' i.e '.in'.
- We have to remove the error

[ ]:
```
amazons1_copy['Sales Channel'] = amazons1_copy['Sales Channel'].replace('Amazon.
↪in','Amazon')
```

[ ]:
```
amazons1_copy['Sales Channel'].unique()
```

[ ]:
```
array(['Amazon', 'Non-Amazon'], dtype=object)
```

[ ]:
```
amazons1_copy.head(2)
```

[ ]:
```
              Order ID       Date                        Status Fulfilment  \
0  405-8078784-5731545 2022-04-30                     Cancelled   Merchant
1  171-9198151-1101146 2022-04-30  Shipped - Delivered to Buyer   Merchant

  Sales Channel ship-service-level Category Size Courier Status  Qty currency  \
0        Amazon           Standard  T-shirt    S     On the Way    0      INR
1        Amazon           Standard    Shirt  3XL        Shipped    1      INR

   Amount  ship-city   ship-state  ship-postal-code ship-country    B2B  \
0  647.62     MUMBAI  MAHARASHTRA          400081.0           IN  False
1  406.00  BENGALURU    KARNATAKA          560085.0           IN  False

  fulfilled-by     status         order status
0    Easy Ship  Cancelled              loading
1    Easy Ship   Shipped    Delivered to Buyer
```

[ ]:
```
## handle ship-service-level
amazons1_copy['ship-service-level'].unique()
```

[ ]:
```
array(['Standard', 'Expedited'], dtype=object)
```

- There is no errors in the 'ship-service-level' column.

[ ]:
```
## category column
amazons1_copy['Category'].unique()
```

[ ]:
```
array(['T-shirt', 'Shirt', 'Blazzer', 'Trousers', 'Perfume', 'Socks',
       'Shoes', 'Wallet', 'Watch'], dtype=object)
```

- There is no error in the 'Category' column.

[ ]:
```
## size column
amazons1_copy['Size'].unique()
```

[ ]:
```
array(['S', '3XL', 'XL', 'L', 'XXL', 'XS', '6XL', 'M', '4XL', 'Free',
       '5XL'], dtype=object)
```

- There is no error in the 'Size' column.

[ ]:
```
## Courier Status
amazons1_copy['Courier Status'].unique()
```

[ ]:
```
array(['On the Way', 'Shipped', 'Cancelled', 'Unshipped'], dtype=object)
```

- There is no error in the 'Courier Status' column.

[ ]:
```
## Qty column
amazons1_copy['Qty'].unique()
```

[ ]:
```
array([ 0,  1,  2, 15,  3,  9, 13,  5,  4])
```

- There is no error in the 'Qty' column.

[ ]:
```
## currency column
amazons1_copy['currency'].unique()
```

[ ]:
```
array(['INR', nan], dtype=object)
```

- There is no error in the 'currency' column.

[ ]:
```
## ship-city column
np.set_printoptions(threshold=np.inf)
```

[ ]:
```
amazons1_copy['ship-city'] = amazons1_copy['ship-city'].str.upper()
```

[ ]:
```
def clean_text(text):
    if isinstance(text, str):
        # Remove special characters and numbers, keeping only letters
```

```
        return re.sub(r'[^A-Za-z\s]', '', text)
    else:
        return text  # Return the original value if it's not a string
```

[ ]:
```
amazons1_copy['ship-city'] = amazons1_copy['ship-city'].apply(clean_text)
```

[ ]:
```
amazons1_copy['ship-city'].isnull().sum()
```

[ ]:
```
28
```

[ ]:
```
amazons1_copy['ship-city'] = amazons1_copy['ship-city'].replace(np.nan, "NA")
```

[ ]:
```
amazons1_copy['ship-city'].isnull().sum()
```

[ ]:
```
0
```

[ ]:
```
amazons1_copy['ship-state'] = amazons1_copy['ship-state'].str.upper()
```

[ ]:
```
amazons1_copy['ship-state'].unique()
```

[ ]:
```
array(['MAHARASHTRA', 'KARNATAKA', 'PUDUCHERRY', 'TAMIL NADU',
       'UTTAR PRADESH', 'CHANDIGARH', 'TELANGANA', 'ANDHRA PRADESH',
       'RAJASTHAN', 'DELHI', 'HARYANA', 'ASSAM', 'JHARKHAND',
       'CHHATTISGARH', 'ODISHA', 'KERALA', 'MADHYA PRADESH',
       'WEST BENGAL', 'NAGALAND', 'GUJARAT', 'UTTARAKHAND', 'BIHAR',
       'JAMMU & KASHMIR', 'PUNJAB', 'HIMACHAL PRADESH',
       'ARUNACHAL PRADESH', 'GOA', 'MEGHALAYA', 'MANIPUR', 'TRIPURA',
       'LADAKH', 'DADRA AND NAGAR', 'SIKKIM', nan, 'ANDAMAN & NICOBAR',
       'RAJSHTHAN', 'NL', 'MIZORAM', 'NEW DELHI',
       'PUNJAB/MOHALI/ZIRAKPUR', 'RJ', 'ORISSA', 'LAKSHADWEEP', 'PB',
       'APO', 'AR', 'PONDICHERRY', 'RAJSTHAN'], dtype=object)
```

[ ]:
```
amazons1_copy['ship-state'] = amazons1_copy['ship-state'].
 ↪replace("NA","NAGALAND")
amazons1_copy['ship-state'] = amazons1_copy['ship-state'].
 ↪replace("NL","NAGALAND")
amazons1_copy['ship-state'] = amazons1_copy['ship-state'].
 ↪replace("RJ","RAJSTHAN")
amazons1_copy['ship-state'] = amazons1_copy['ship-state'].replace("PB","PUNJAB")
amazons1_copy['ship-state'] = amazons1_copy['ship-state'].replace("APO","ANDHRA␣
 ↪PRADESH")
amazons1_copy['ship-state'] = amazons1_copy['ship-state'].
 ↪replace("AR","ARUNACHAL PRADESH")
amazons1_copy['ship-state'] = amazons1_copy['ship-state'].replace("DELHI","NEW␣
 ↪DELHI")
```

```
amazons1_copy['ship-state'] = amazons1_copy['ship-state'].
 ↪replace("RAJSTHAN","RAJASTHAN")
amazons1_copy['ship-state'] = amazons1_copy['ship-state'].
 ↪replace("RAJSHTHAN","RAJASTHAN")
amazons1_copy['ship-state'] = amazons1_copy['ship-state'].replace("PUNJAB/
 ↪MOHALI/ZIRAKPUR","PUNJAB")
```

[ ]:
```
amazons1_copy['ship-state'].unique()
```

[ ]:
```
array(['MAHARASHTRA', 'KARNATAKA', 'PUDUCHERRY', 'TAMIL NADU',
       'UTTAR PRADESH', 'CHANDIGARH', 'TELANGANA', 'ANDHRA PRADESH',
       'RAJASTHAN', 'NEW DELHI', 'HARYANA', 'ASSAM', 'JHARKHAND',
       'CHHATTISGARH', 'ODISHA', 'KERALA', 'MADHYA PRADESH',
       'WEST BENGAL', 'NAGALAND', 'GUJARAT', 'UTTARAKHAND', 'BIHAR',
       'JAMMU & KASHMIR', 'PUNJAB', 'HIMACHAL PRADESH',
       'ARUNACHAL PRADESH', 'GOA', 'MEGHALAYA', 'MANIPUR', 'TRIPURA',
       'LADAKH', 'DADRA AND NAGAR', 'SIKKIM', nan, 'ANDAMAN & NICOBAR',
       'MIZORAM', 'ORISSA', 'LAKSHADWEEP', 'PONDICHERRY'], dtype=object)
```

[ ]:
```
amazons1_copy['ship-state'].isnull().sum()
```

[ ]:
```
28
```

[ ]:
```
amazons1_copy['ship-state'] = amazons1_copy['ship-state'].replace(np.nan,"NA")
```

[ ]:
```
amazons1_copy['ship-state'].isnull().sum()
```

[ ]:
```
0
```

[ ]:
```
## ship-country
amazons1_copy['ship-country'].unique()
```

[ ]:
```
array(['IN', nan], dtype=object)
```

[ ]:
```
amazons1_copy['ship-country'] = amazons1_copy['ship-country'].map({'IN':␣
 ↪'INDIA', np.nan : 'INDIA'})
```

[ ]:
```
amazons1_copy['ship-country'].unique()
```

[ ]:
```
array(['INDIA'], dtype=object)
```

[ ]:
```
## B2B
amazons1_copy['B2B'].unique()
```

[ ]:
```
array([False,  True])
```

```
[ ]: ## fulfilled-by
     amazons1_copy['fulfilled-by'].unique()
```

```
[ ]: array(['Easy Ship', nan], dtype=object)
```

```
[ ]: amazons1_copy['fulfilled-by'].isnull().sum()
```

```
[ ]: 83906
```

```
[ ]: amazons1_copy['fulfilled-by'] = amazons1_copy['fulfilled-by'].replace(np.nan,␣
      ↪"NA")
```

```
[ ]: amazons1_copy['fulfilled-by'].isnull().sum()
```

```
[ ]: 0
```

```
[ ]: amazons1_copy['currency'].unique()
```

```
[ ]: array(['INR', nan], dtype=object)
```

```
[ ]: amazons1_copy['currency'] = amazons1_copy['currency'].replace(np.nan,"INR")
```

```
[ ]: amazons1_copy['currency'].unique()
```

```
[ ]: array(['INR'], dtype=object)
```

```
[ ]: amazons1_copy.isnull().sum()
```

```
[ ]: 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                0
     Amount               7395
     ship-city               0
     ship-state              0
     ship-postal-code       28
     ship-country            0
     B2B                     0
     fulfilled-by            0
     status                  0
```

```
order status            0
dtype: int64
```

[ ]:
```
amazons1_copy['ship-postal-code'] = amazons1_copy['ship-postal-code'].
↪replace(np.nan,'None')
```

[ ]:
```
amazons1_copy.isnull().sum()
```

[ ]:
```
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                0
Amount               7395
ship-city               0
ship-state              0
ship-postal-code        0
ship-country            0
B2B                     0
fulfilled-by            0
status                  0
order status            0
dtype: int64
```

[ ]:
```
amazons1_copy.dropna(inplace=True)
```

[ ]:
```
amazons1_copy.isnull().sum()
```

[ ]:
```
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              0
Amount                0
ship-city             0
```

```
ship-state            0
ship-postal-code      0
ship-country          0
B2B                   0
fulfilled-by          0
status                0
order status          0
dtype: int64
```

- There is no null values in the data

1.9 Explore More Data

[ ]:
```
amazons1_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 112834 entries, 0 to 128975
Data columns (total 20 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   Order ID            112834 non-null  object
 1   Date                112834 non-null  datetime64[ns]
 2   Status              112834 non-null  object
 3   Fulfilment          112834 non-null  object
 4   Sales Channel       112834 non-null  object
 5   ship-service-level  112834 non-null  object
 6   Category            112834 non-null  object
 7   Size                112834 non-null  object
 8   Courier Status      112834 non-null  object
 9   Qty                 112834 non-null  int64
 10  currency            112834 non-null  object
 11  Amount              112834 non-null  float64
 12  ship-city           112834 non-null  object
 13  ship-state          112834 non-null  object
 14  ship-postal-code    112834 non-null  object
 15  ship-country        112834 non-null  object
 16  B2B                 112834 non-null  bool
 17  fulfilled-by        112834 non-null  object
 18  status              112834 non-null  object
 19  order status        112834 non-null  object
dtypes: bool(1), datetime64[ns](1), float64(1), int64(1), object(16)
memory usage: 17.3+ MB
```

[ ]:
```
amazons1_copy.head(5)
```

[ ]:
```
              Order ID       Date                     Status Fulfilment  \
 0  405-8078784-5731545 2022-04-30                  Cancelled   Merchant
```

```
1  171-9198151-1101146 2022-04-30  Shipped - Delivered to Buyer   Merchant
2  404-0687676-7273146 2022-04-30                       Shipped     Amazon
3  403-9615377-8133951 2022-04-30                     Cancelled   Merchant
4  407-1069790-7240320 2022-04-30                       Shipped     Amazon

  Sales Channel ship-service-level  Category Size Courier Status  Qty  \
0        Amazon           Standard   T-shirt    S     On the Way    0
1        Amazon           Standard     Shirt  3XL        Shipped    1
2        Amazon          Expedited     Shirt   XL        Shipped    1
3        Amazon           Standard   Blazzer    L     On the Way    0
4        Amazon          Expedited  Trousers  3XL        Shipped    1

  currency  Amount    ship-city   ship-state ship-postal-code ship-country  \
0      INR  647.62       MUMBAI  MAHARASHTRA         400081.0        INDIA
1      INR  406.00    BENGALURU    KARNATAKA         560085.0        INDIA
2      INR  329.00  NAVI MUMBAI  MAHARASHTRA         410210.0        INDIA
3      INR  753.33   PUDUCHERRY   PUDUCHERRY         605008.0        INDIA
4      INR  574.00      CHENNAI   TAMIL NADU         600073.0        INDIA

    B2B fulfilled-by     status         order status
0  False    Easy Ship  Cancelled              loading
1  False    Easy Ship   Shipped    Delivered to Buyer
2   True           NA   Shipped              loading
3  False    Easy Ship  Cancelled              loading
4  False           NA   Shipped              loading
```

[ ]:
```
## create a new column total amount
amazons1_copy['Total Amount'] = amazons1_copy['Qty'] * amazons1_copy['Amount']
```

[ ]:
```
amazons1_copy.head(5)
```

[ ]:
```
             Order ID       Date                         Status Fulfilment  \
0  405-8078784-5731545 2022-04-30                      Cancelled   Merchant
1  171-9198151-1101146 2022-04-30  Shipped - Delivered to Buyer   Merchant
2  404-0687676-7273146 2022-04-30                       Shipped     Amazon
3  403-9615377-8133951 2022-04-30                     Cancelled   Merchant
4  407-1069790-7240320 2022-04-30                       Shipped     Amazon

  Sales Channel ship-service-level  Category Size Courier Status  Qty  …  \
0        Amazon           Standard   T-shirt    S     On the Way    0  …
1        Amazon           Standard     Shirt  3XL        Shipped    1  …
2        Amazon          Expedited     Shirt   XL        Shipped    1  …
3        Amazon           Standard   Blazzer    L     On the Way    0  …
4        Amazon          Expedited  Trousers  3XL        Shipped    1  …

   Amount    ship-city   ship-state ship-postal-code ship-country    B2B  \
0  647.62       MUMBAI  MAHARASHTRA         400081.0        INDIA  False
```

```
1  406.00    BENGALURU    KARNATAKA         560085.0        INDIA  False
2  329.00  NAVI MUMBAI  MAHARASHTRA         410210.0        INDIA   True
3  753.33   PUDUCHERRY   PUDUCHERRY         605008.0        INDIA  False
4  574.00      CHENNAI   TAMIL NADU         600073.0        INDIA  False

   fulfilled-by     status         order status Total Amount
0     Easy Ship  Cancelled              loading          0.0
1     Easy Ship   Shipped    Delivered to Buyer        406.0
2            NA    Shipped              loading        329.0
3     Easy Ship  Cancelled              loading          0.0
4            NA    Shipped              loading        574.0

[5 rows x 21 columns]
```

[ ]:
```
amazons1_copy.shape
```

[ ]:
```
(112834, 21)
```

[ ]:
```
amazons1_copy[(amazons1_copy['Qty'] == 0) & (amazons1_copy['Amount'] != 0)].
↪head(5)
```

[ ]:
```
                Order ID       Date     Status Fulfilment Sales Channel  \
0    405-8078784-5731545 2022-04-30  Cancelled   Merchant        Amazon
3    403-9615377-8133951 2022-04-30  Cancelled   Merchant        Amazon
23   404-6019946-2909948 2022-04-30  Cancelled   Merchant        Amazon
83   404-6522553-9345930 2022-04-30  Cancelled   Merchant        Amazon
178  171-1224053-5752314 2022-04-30  Cancelled   Merchant        Amazon

    ship-service-level  Category Size Courier Status  Qty  …   Amount  \
0             Standard   T-shirt    S     On the Way    0  …   647.62
3             Standard   Blazzer    L     On the Way    0  …   753.33
23            Standard   T-shirt    M     On the Way    0  …   570.48
83            Standard   T-shirt    M     On the Way    0  …  1105.36
178           Standard  Trousers    L     On the Way    0  …   463.81

      ship-city   ship-state ship-postal-code ship-country    B2B  \
0        MUMBAI  MAHARASHTRA         400081.0        INDIA  False
3    PUDUCHERRY   PUDUCHERRY         605008.0        INDIA  False
23         PUNE  MAHARASHTRA         411044.0        INDIA  False
83     DEHRADUN  UTTARAKHAND         248001.0        INDIA  False
178   BENGALURU    KARNATAKA         560087.0        INDIA  False

     fulfilled-by     status order status Total Amount
0       Easy Ship  Cancelled      loading          0.0
3       Easy Ship  Cancelled      loading          0.0
23      Easy Ship  Cancelled      loading          0.0
83      Easy Ship  Cancelled      loading          0.0
```

```
178     Easy Ship  Cancelled      loading          0.0

[5 rows x 21 columns]
```

[ ]:
```
## change column name
change_column = { 'ship-service-level' : 'Service',
                  'ship-city' : 'City',
                  'ship-state' : 'State',
                  'ship-postal-code' : 'Postal Code',
                  'ship-country' : 'Country'
}
```

[ ]:
```
amazons1_copy = amazons1_copy.rename(columns= change_column)
```

[ ]:
```
amazons1_copy.head(5)
```

[ ]:
```
               Order ID       Date                        Status Fulfilment  \
0  405-8078784-5731545 2022-04-30                     Cancelled   Merchant
1  171-9198151-1101146 2022-04-30  Shipped - Delivered to Buyer   Merchant
2  404-0687676-7273146 2022-04-30                       Shipped     Amazon
3  403-9615377-8133951 2022-04-30                     Cancelled   Merchant
4  407-1069790-7240320 2022-04-30                       Shipped     Amazon

  Sales Channel    Service  Category Size Courier Status  Qty  …  Amount  \
0        Amazon   Standard   T-shirt    S     On the Way    0  …  647.62
1        Amazon   Standard     Shirt  3XL        Shipped    1  …  406.00
2        Amazon  Expedited     Shirt   XL        Shipped    1  …  329.00
3        Amazon   Standard   Blazzer    L     On the Way    0  …  753.33
4        Amazon  Expedited  Trousers  3XL        Shipped    1  …  574.00

          City        State Postal Code Country    B2B  fulfilled-by  \
0       MUMBAI  MAHARASHTRA    400081.0   INDIA  False     Easy Ship
1    BENGALURU    KARNATAKA    560085.0   INDIA  False     Easy Ship
2  NAVI MUMBAI  MAHARASHTRA    410210.0   INDIA   True            NA
3   PUDUCHERRY   PUDUCHERRY    605008.0   INDIA  False     Easy Ship
4      CHENNAI   TAMIL NADU    600073.0   INDIA  False            NA

      status         order status Total Amount
0  Cancelled              loading          0.0
1   Shipped    Delivered to Buyer        406.0
2    Shipped              loading        329.0
3  Cancelled              loading          0.0
4    Shipped              loading        574.0

[5 rows x 21 columns]
```

[ ]:
```
## drop Status Column
amazons1_copy.drop('Status', axis = 1, inplace = True)
```

[ ]:
```
amazons1_copy.head(5)
```

[ ]:
```
              Order ID       Date Fulfilment Sales Channel    Service  \
0  405-8078784-5731545 2022-04-30   Merchant        Amazon   Standard
1  171-9198151-1101146 2022-04-30   Merchant        Amazon   Standard
2  404-0687676-7273146 2022-04-30     Amazon        Amazon  Expedited
3  403-9615377-8133951 2022-04-30   Merchant        Amazon   Standard
4  407-1069790-7240320 2022-04-30     Amazon        Amazon  Expedited

   Category Size Courier Status  Qty currency  Amount         City  \
0   T-shirt    S     On the Way    0      INR  647.62       MUMBAI
1     Shirt  3XL        Shipped    1      INR  406.00    BENGALURU
2     Shirt   XL        Shipped    1      INR  329.00  NAVI MUMBAI
3   Blazzer    L     On the Way    0      INR  753.33   PUDUCHERRY
4  Trousers  3XL        Shipped    1      INR  574.00      CHENNAI

          State Postal Code Country    B2B fulfilled-by     status  \
0  MAHARASHTRA    400081.0   INDIA  False    Easy Ship  Cancelled
1    KARNATAKA    560085.0   INDIA  False    Easy Ship   Shipped
2  MAHARASHTRA    410210.0   INDIA   True           NA    Shipped
3   PUDUCHERRY    605008.0   INDIA  False    Easy Ship  Cancelled
4   TAMIL NADU    600073.0   INDIA  False           NA    Shipped

           order status  Total Amount
0               loading           0.0
1   Delivered to Buyer         406.0
2               loading         329.0
3               loading           0.0
4               loading         574.0
```

[ ]:
```
## seperate year, month, day
amazons1_copy['Day'] = amazons1_copy['Date'].dt.day
amazons1_copy['Month'] = amazons1_copy['Date'].dt.month
amazons1_copy['Year'] = amazons1_copy['Date'].dt.year
```

[ ]:
```
amazons1_copy.head(5)
```

[ ]:
```
              Order ID       Date Fulfilment Sales Channel    Service  \
0  405-8078784-5731545 2022-04-30   Merchant        Amazon   Standard
1  171-9198151-1101146 2022-04-30   Merchant        Amazon   Standard
2  404-0687676-7273146 2022-04-30     Amazon        Amazon  Expedited
3  403-9615377-8133951 2022-04-30   Merchant        Amazon   Standard
4  407-1069790-7240320 2022-04-30     Amazon        Amazon  Expedited
```

```
   Category Size Courier Status  Qty currency  …  Postal Code Country  \
0   T-shirt    S     On the Way    0      INR  …     400081.0   INDIA
1     Shirt  3XL        Shipped    1      INR  …     560085.0   INDIA
2     Shirt   XL        Shipped    1      INR  …     410210.0   INDIA
3   Blazzer    L     On the Way    0      INR  …     605008.0   INDIA
4  Trousers  3XL        Shipped    1      INR  …     600073.0   INDIA

     B2B fulfilled-by     status         order status Total Amount Day Month  \
0  False    Easy Ship  Cancelled              loading          0.0  30     4
1  False    Easy Ship   Shipped    Delivered to Buyer        406.0  30     4
2   True           NA    Shipped              loading        329.0  30     4
3  False    Easy Ship  Cancelled              loading          0.0  30     4
4  False           NA    Shipped              loading        574.0  30     4

   Year
0  2022
1  2022
2  2022
3  2022
4  2022

[5 rows x 23 columns]
```

[ ]:
```
## month column
amazons1_copy['Month'].unique()
```

[ ]:
```
array([4, 3, 5, 6], dtype=int32)
```

[ ]:
```
## year column
amazons1_copy['Year'].unique()
```

[ ]:
```
array([2022], dtype=int32)
```

[ ]:
```
## extract month name from date column
amazons1_copy['month_name'] = amazons1_copy['Date'].dt.month_name()
```

[ ]:
```
amazons1_copy.head(5)
```

[ ]:
```
              Order ID       Date Fulfilment Sales Channel    Service  \
0  405-8078784-5731545 2022-04-30   Merchant        Amazon   Standard
1  171-9198151-1101146 2022-04-30   Merchant        Amazon   Standard
2  404-0687676-7273146 2022-04-30     Amazon        Amazon  Expedited
3  403-9615377-8133951 2022-04-30   Merchant        Amazon   Standard
4  407-1069790-7240320 2022-04-30     Amazon        Amazon  Expedited

   Category Size Courier Status  Qty currency  …  Country    B2B  \
0   T-shirt    S     On the Way    0      INR  …    INDIA  False
```

```
1     Shirt  3XL        Shipped    1      INR  …    INDIA  False
2     Shirt   XL        Shipped    1      INR  …    INDIA   True
3   Blazzer    L     On the Way    0      INR  …    INDIA  False
4  Trousers  3XL        Shipped    1      INR  …    INDIA  False

  fulfilled-by     status         order status  Total Amount Day Month  Year  \
0    Easy Ship  Cancelled              loading           0.0  30     4  2022
1    Easy Ship   Shipped    Delivered to Buyer         406.0  30     4  2022
2           NA    Shipped              loading         329.0  30     4  2022
3    Easy Ship  Cancelled              loading           0.0  30     4  2022
4           NA    Shipped              loading         574.0  30     4  2022

   month_name
0       April
1       April
2       April
3       April
4       April

[5 rows x 24 columns]
```

[ ]:
```
## check one last time the dataset before save the data file
amazons1_copy.duplicated().sum()
```

[ ]:
```
0
```

[ ]:
```
amazons1_copy.isnull().sum()
```

[ ]:
```
Order ID          0
Date              0
Fulfilment        0
Sales Channel     0
Service           0
Category          0
Size              0
Courier Status    0
Qty               0
currency          0
Amount            0
City              0
State             0
Postal Code       0
Country           0
B2B               0
fulfilled-by      0
status            0
order status      0
```

```
Total Amount      0
Day               0
Month             0
Year              0
month_name        0
dtype: int64
```

[ ]:
```
amazons1_copy.head(4)
```

[ ]:
```
              Order ID       Date Fulfilment Sales Channel    Service  \
0  405-8078784-5731545 2022-04-30   Merchant        Amazon   Standard
1  171-9198151-1101146 2022-04-30   Merchant        Amazon   Standard
2  404-0687676-7273146 2022-04-30     Amazon        Amazon  Expedited
3  403-9615377-8133951 2022-04-30   Merchant        Amazon   Standard

  Category Size Courier Status  Qty currency  …  Country    B2B  \
0  T-shirt    S     On the Way    0      INR  …    INDIA  False
1    Shirt  3XL        Shipped    1      INR  …    INDIA  False
2    Shirt   XL        Shipped    1      INR  …    INDIA   True
3  Blazzer    L     On the Way    0      INR  …    INDIA  False

  fulfilled-by     status         order status  Total Amount Day Month  Year  \
0    Easy Ship  Cancelled              loading           0.0  30     4  2022
1    Easy Ship   Shipped    Delivered to Buyer         406.0  30     4  2022
2           NA    Shipped              loading         329.0  30     4  2022
3    Easy Ship  Cancelled              loading           0.0  30     4  2022

   month_name
0       April
1       April
2       April
3       April

[4 rows x 24 columns]
```

1.10 EDA And Visualization

- Perform data analysis and do visualization

1.11 1. Sales Overview: Understand the overall sales performance, trends, and patterns over time.

[ ]:
```
## preprocess the data
amazons1_copy = amazons1_copy.sort_values(by='Date')
```

[ ]:
```
amazons1_copy.head(5)
```

[ ]:
```
                 Order ID       Date Fulfilment Sales Channel    Service  \
48971  404-1445673-1345134 2022-03-31   Merchant        Amazon   Standard
48997  402-0339394-3540335 2022-03-31     Amazon        Amazon  Expedited
48998  408-6597776-7485121 2022-03-31     Amazon        Amazon  Expedited
48999  404-1415044-3213110 2022-03-31     Amazon        Amazon  Expedited
49000  171-3572716-7638764 2022-03-31   Merchant        Amazon   Standard

       Category Size Courier Status  Qty currency  …  Country    B2B  \
48971     Shirt    L        Shipped    1      INR  …    INDIA  False
48997   T-shirt   XL        Shipped    1      INR  …    INDIA  False
48998  Trousers   XS        Shipped    1      INR  …    INDIA  False
48999   T-shirt  3XL        Shipped    1      INR  …    INDIA  False
49000     Shirt  XXL        Shipped    1      INR  …    INDIA  False

      fulfilled-by    status         order status  Total Amount Day Month  \
48971    Easy Ship  Shipped    Delivered to Buyer         495.0  31     3
48997           NA   Shipped              loading         688.0  31     3
48998           NA   Shipped              loading         354.0  31     3
48999           NA   Shipped              loading         698.0  31     3
49000    Easy Ship  Shipped    Delivered to Buyer         248.0  31     3

       Year  month_name
48971  2022       March
48997  2022       March
48998  2022       March
48999  2022       March
49000  2022       March

[5 rows x 24 columns]
```

[ ]:
```
amazons1_copy = amazons1_copy.drop(['Day', 'Month', 'Year', 'month_name'], axis
 ↪= 1)
```

[ ]:
```
amazons1_copy.head(4)
```

[ ]:
```
                 Order ID       Date Fulfilment Sales Channel    Service  \
48971  404-1445673-1345134 2022-03-31   Merchant        Amazon   Standard
48997  402-0339394-3540335 2022-03-31     Amazon        Amazon  Expedited
48998  408-6597776-7485121 2022-03-31     Amazon        Amazon  Expedited
48999  404-1415044-3213110 2022-03-31     Amazon        Amazon  Expedited

       Category Size Courier Status  Qty currency  Amount       City  \
48971     Shirt    L        Shipped    1      INR   495.0    KOLKATA
48997   T-shirt   XL        Shipped    1      INR   688.0  GHAZIABAD
48998  Trousers   XS        Shipped    1      INR   354.0      DELHI
48999   T-shirt  3XL        Shipped    1      INR   698.0      SIKAR
```

```
                State Postal Code Country    B2B fulfilled-by    status  \
48971    WEST BENGAL    700124.0   INDIA  False    Easy Ship  Shipped
48997  UTTAR PRADESH    201005.0   INDIA  False           NA   Shipped
48998      NEW DELHI    110016.0   INDIA  False           NA   Shipped
48999      RAJASTHAN    332001.0   INDIA  False           NA   Shipped

              order status  Total Amount
48971   Delivered to Buyer         495.0
48997              loading         688.0
48998              loading         354.0
48999              loading         698.0
```

[ ]:
```
amazons1_copy = amazons1_copy.rename(columns={'Total Amount' : 'Total Sales'})
```

[ ]:
```
amazons1_copy['Total Sales'].unique()
```

[ ]:
```
array([  495.,   688.,   354.,   698.,   248.,   322.,   599.,     0.,
         542.,   519.,  1165.,   358.,   417.,   449.,   481.,   877.,
         437.,   730.,   899.,   744.,   729.,  1147.,   499.,   493.,
         998.,   353.,   657.,   453.,  1138.,   399.,   798.,   627.,
         259.,   567.,   490.,   791.,  1112.,   385.,   549.,   761.,
        1099.,   561.,   927.,   799.,   648.,   999.,  1199.,  1271.,
         474.,   446.,   472.,   379.,   984.,  1201.,   315.,   394.,
         364.,   293.,   317.,   313.,  1496.,  1799.,   427.,   955.,
         280.,   727.,   990.,   324.,   849.,  1033.,   330.,  1245.,
         452.,   597.,  1166.,  1130.,   424.,   434.,   530.,   969.,
         750.,   672.,   469.,   383.,   528.,   396.,   448.,  1481.,
         674.,  1354.,   521.,   547.,   527.,   689.,   715.,   642.,
        1450.,   321.,   426.,   349.,   869.,   360.,  1129.,   646.,
         859.,   329.,   589.,   775.,   299.,   647.,   699.,   484.,
         930.,   885.,   841.,   958.,   421.,  1298.,  1426.,   726.,
         277.,   563.,   339.,  1542.,  1576.,  1174.,   279.,   941.,
         573.,   496.,   845.,   817.,  1324.,  1085.,   556.,   432.,
        1449.,  1229.,   825.,   569.,   423.,   898.,   865.,  1079.,
        1196.,   443.,  1125.,   836.,   450.,   365.,   377.,   480.,
         758.,   912.,   311.,   460.,   616.,  1299.,  1249.,   269.,
        1154.,   347.,   714.,   371.,   683.,   618.,   518.,   774.,
         747.,   749.,  2792.,   425.,   333.,   442.,   387.,   677.,
         546.,   795.,   625.,   412.,  1812.,   764.,   696.,   342.,
        1078.,   650.,  1053.,   871.,  1127.,   537.,   949.,  1695.,
        1228.,   372.,  1120.,   420.,   594.,   925.,   693.,   345.,
         743.,   389.,   464.,  3051.,  3348.,  1075.,   297.,   568.,
         267.,   566.,   678.,  4132.,  1399.,  6048.,   870.,   338.,
         584.,   956.,   622.,   473.,   391.,   939.,  2196.,   352.,
         264.,   466.,   640.,  1170.,   328.,   343.,   488.,   510.,
         586.,   505.,   292.,  1648.,  1020.,   284.,   654.,   458.,
         786.,   465.,  7191.,  1250.,   630.,   684.,  1213.,   359.,
```

```
 873.,   241.,  1668.,   526.,   590.,   819.,   405.,  1800.,
 790.,   598.,   376.,  1254.,  1233.,   428.,   676.,   516.,
 620.,   534.,   935.,   588.,  1396.,   355.,  1220.,   378.,
1691.,   783.,   947.,   675.,   368.,  1030.,  1039.,  3832.,
2472.,   745.,  1386.,   400.,   772.,   574.,   712.,   363.,
1704.,   327.,   690.,   459.,   517.,   475.,   888.,   666.,
 487.,   476.,  1158.,   824.,  1173.,   721.,   968.,   788.,
1186.,   612.,   759.,   571.,   291.,  1523.,   435.,   348.,
 635.,   653.,   318.,  1338.,   751.,  1149.,   660.,   545.,
2584.,   771.,   716.,   388.,  1133.,   658.,   438.,  1270.,
 477.,   631.,   662.,   413.,  1281.,   847.,   357.,   419.,
 995.,   967.,  1256.,   852.,   319.,  1126.,   560.,   685.,
 737.,  1593.,   295.,   629.,   680.,   575.,   665.,   613.,
 393.,   579.,   471.,   582.,  1237.,   316.,   591.,   478.,
 362.,  1238.,   682.,   595.,   911.,   950.,   548.,   645.,
 501.,   522.,   350.,   920.,   916.,   736.,   636., 11440.,
 229.,   381.,   283.,  3152.,  1146.,   881.,  1596.,   429.,
1556.,   455.,   199.,   539.,   717.,   583.,  1671.,   874.,
 833.,   637.,  2796.,   587.,   523.,   544.,   380.,   271.,
 249.,  1438.,  2961.,   789.,   298.,   886.,   818.,  1369.,
 895.,   659.,  1163.,   486.,   463.,   540.,  1388.,  1432.,
 725.,  1260.,  1065.,  1115.,   626.,  1692.,   801.,   835.,
1092.,   525.,   491.,   511.,   807.,   856.,  1140.,  1204.,
 406.,   581.,   533.,  1463.,   792.,   309.,   692.,   922.,
 864.,   462.,   562.,   531.,   512.,  1200.,  1504.,  1624.,
1669.,  1272.,   733.,   375.,   671.,   909.,   551.,   294.,
1287.,   641.,   782.,  1676.,  1316.,  1440.,  2388.,   441.,
 382.,  2784.,   414.,  1900.,  2640.,   816.,  6120.,   632.,
1472.,  1836.,  1696.,  3868.,   763.,  1221.,  1136.,  2044.,
6525.,  3296.,   667.,   273.,   974.,   286.,   820.,   468.,
 848.,  2900.,  5728.,  2068.,  3596.,   832.,  1948.,  4460.,
3384.,   814.,   422.,  1448.,   529.,  1768.,  1884.,  2272.,
 370.,   719.,  1729.,  3312.,   524.,  2072.,  1972.,  2296.,
 344.,   433.,   541.,  2976.,  2176.,  3978.,  2299.,   643.,
2616.,  4448.,  5264.,   989.,  1999.,  2476.,  1477.,  1740.,
4396.,   638.,  1548.,  2160.,   838., 27475.,   929.,  1036.,
 436.,  1728.,   889.,  3552.,  1499.,   619.,   839.,  1944.,
 884.,  1980.,   431.,   390.,  4260.,  3036.,   655.,  1111.,
 724.,   965.,  1257.,  1093.,  1159.,  4584.,  1776.,   513.,
1389.,   765.,   857.,  1066.,  1205.,   694.,   351.,   634.,
1137.,  2536.,   320.,  2896., 12144.,  1670.,   310.,   664.,
2380.,   461.,   497.,  2972.,  2620.,   808.,  4744.,   301.,
 307.,  1315.,   921.,   603.,  1083.,  1096.,   776.,   709.,
1008.,  1068.,   962.,   565.,   649.,   828.,   854.,  1258.,
1122.,   837.,  1309.,   988.,   720.,   605.,   850.,   553.,
1364.,  1072.,   479.,   356.,   633.,  1098.,   718.,   752.,
 373.,   872.,  1279.,  1996.,  1323.,  1088.,   489.,   492.,
```

```
 1031.,  1352.,  1164.,   785.,  1698.,  3300.,   346.,   711.,
  668.,   457.,   614.,   882.,   703.,  1192.,   498.,   704.,
  855.,   606.,   386.,   507.,   308.,   754.,   416.,   602.,
  314.,   762.,   892.,   543.,   323.,   842.,  1442.,   550.,
  306.,   979.,   325.,   467.,   335.,   735.,   369.,   787.,
  760.,  1018.,   607.,   919.,  1082.,  1319.,   931.,   846.,
 1108.,   397.,  1043.,  1041.,  1202.,   940.,   336.,  1672.,
  366.,   702.,   331.,   661.,   803.,   985.,  1152.,   756.,
  933.,   564.,   572.,   596.,   418.,  1409.,  1531.,  1549.,
 1063.,   812.,   809.,   502.,  1325.,   902.,   728.,  1559.,
 2452.,   802.,  1398.,   796.,   558.,   880.,   876.,   827.,
 1333.,   767.,   305.,   361.,  1473.,  1168.,   430., 11184.,
 5292.,   831.,  1073.,   494.,  4716.,   797.,  1419.,   669.,
 1362.,   883.,  1231.,   738.,   811.,  1268.,   611.,   663.,
  332.,  2442.,   398.,   535.,   410.,  3744.,   554.,   593.,
  384.,   559.,  1297.,  2808.,   800.,  1349.,  2564.,  2448.,
  570.,  2836.,   777.,  1284.,  1649.,  5508.,  1089.,   769.,
  973.,   766.,   773.,   936.,  3996.,   341.,   755.,  6174.,
  753.,  5352.,   741.,   686.,  1132.,  2396.,  2264.,   552.,
 1492.,   621.,  3540.,  2052.,  3084.,  2500.,   878.,   937.,
 2598.,   604.,  1181.,   451.,   337.,   976.,  2660.,  5396.,
 1629.,  3056.,  2592.,  1528.,  1189.,  2524.,  3108.,  2456.,
 4653.,  5166.,   844.,   367.,  7965.,   615.,   679.,   440.,
 2316.,  4596.,  5769.,  1984.,   639.,  2488.,   411.,  2940.,
 1210.,  1610.,  1329.,   879.,  1076.,  1044.,   483.,  1377.,
 1613.,  1091.,  1176.,   794.,   908.,   948.,  1294.,  1603.,
  404.,  4488.,  1447.,  1198.,   748.,  1248.,  3483.,  1187.,
 2628.,  1013.,  1566.,  6016.,  1139.,  1301.,  3148.,  1403.,
 1772.,   826.,  5112.,  4131.,  2248.,   793., 10656.,  5788.,
  536.,  1148.,  1234.,   402.,  1380.,   652.,  1094.,  2916.,
  707.,  1332.,  2540.,  1828.,  2944.,  4364.,  3032.,  2148.,
  580.,  2752.,  6960., 10875.,  1832.,  2768.,   987.,  4500.,
 3428.,  2276.,  2580.,  3952.,   312.,  2656.,   515.,   409.,
 2352.,  3168.,  1872.,  2028.,   695.,  3324.,   891.,   334.,
 1664.,  1412.,  2260.,   723.,  2996.,  1269.,  1964.,  3196.,
  779.,  2772.,  1904.,  4652.,  3344.])
```

- Aggregate the sales data to analyze trends over different time periods like daily, monthly, quarterly, or yearly.

[ ]:
```
# Convert 'Total Amount' and 'B2B' columns to numeric, coercing errors
amazons1_copy['Total Amount'] = pd.to_numeric(amazons1_copy['Total Sales'],␣
 ↪errors='coerce')
amazons1_copy['B2B'] = pd.to_numeric(amazons1_copy['B2B'], errors='coerce')

# Check for NaN values after conversion
```

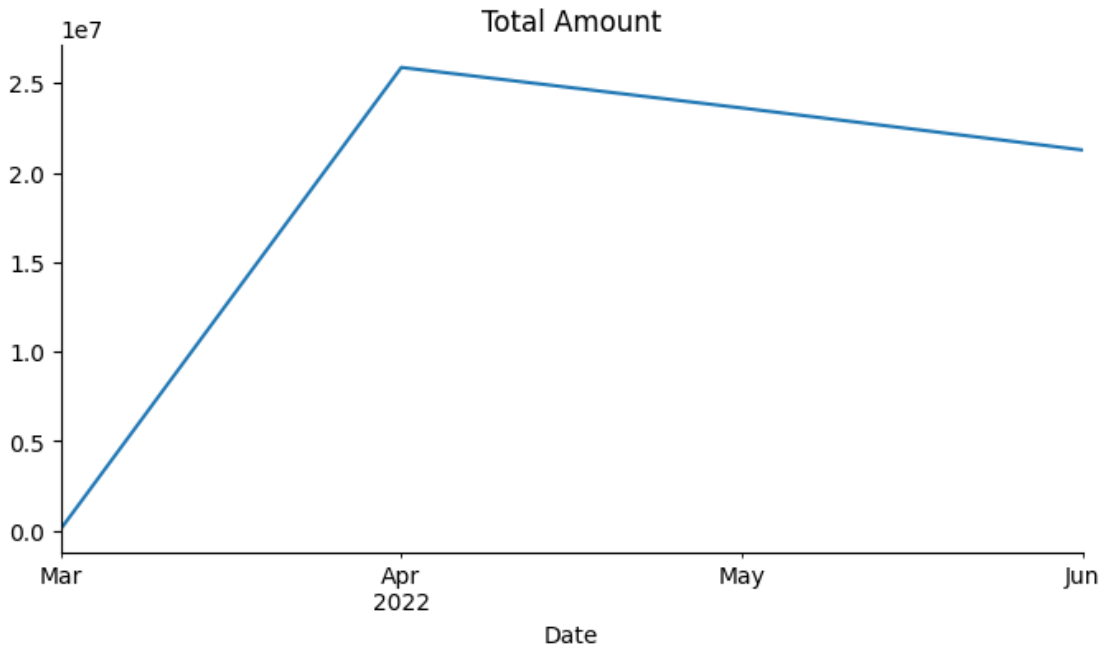
```
print("NaN values in 'Total Amount':", amazons1_copy['Total Amount'].isna().
  ↪sum())
print("NaN values in 'B2B':", amazons1_copy['B2B'].isna().sum())

# Optionally handle NaNs (e.g., fill with 0 or drop rows)
amazons1_copy = amazons1_copy.dropna(subset=['Total Amount', 'B2B'])

# Define numeric_columns
numeric_columns = amazons1_copy.select_dtypes(include=np.number).columns #␣
  ↪Selecting numeric columns

# Perform resampling and summation
monthly_sales = amazons1_copy.resample('M', on='Date')[numeric_columns].sum() #␣
  ↪Resample and select numeric columns


# Check the result
print(monthly_sales.head())
```

```
NaN values in 'Total Amount': 0
NaN values in 'B2B': 0
              Qty       Amount  Total Sales  Total Amount
Date
2022-03-31    144     95971.85      92549.0       92549.0
2022-04-30  41035  26867098.20   25886087.0    25886087.0
2022-05-31  35403  24472622.32   23619231.0    23619231.0
2022-06-30  31888  21887776.50   21270794.0    21270794.0
```

[ ]:
```
monthly_sales
```

[ ]:
```
              Qty       Amount  Total Sales  Total Amount
Date
2022-03-31    144     95971.85      92549.0       92549.0
2022-04-30  41035  26867098.20   25886087.0    25886087.0
2022-05-31  35403  24472622.32   23619231.0    23619231.0
2022-06-30  31888  21887776.50   21270794.0    21270794.0
```

[ ]:
```
# @title Total Amount

from matplotlib import pyplot as plt
monthly_sales['Total Amount'].plot(kind='line', figsize=(8, 4), title='Total␣
  ↪Amount')
plt.gca().spines[['top', 'right']].set_visible(False)
```



1.12 Insights and Observations

- March 2022: 0.093 million in total sales, minimal activity.
- April 2022: 25.89 million in total sales, peak performance.
- May 2022: 23.62 million in total sales, slight decline.
- June 2022: 21.27 million in total sales, continued downward trend.
- Overall, sales peaked in April and then gradually declined over the following months.

[ ]:
```
yearly_sales = amazons1_copy.resample('Y', on = 'Date')[numeric_columns].sum()
```

[ ]:
```
yearly_sales
```

[ ]:
```
               Qty       Amount  Total Sales  Total Amount
Date
2022-12-31  108470  73323468.87   70868661.0    70868661.0
```

```
<google.colab._quickchart_helpers.SectionTitle at 0x7f9eeaecece0>
```

```
from matplotlib import pyplot as plt
import seaborn as sns
def _plot_series(series, series_name, series_index=0):
  palette = list(sns.palettes.mpl_palette('Dark2'))
  counted = (series['Date']
               .value_counts()
             .reset_index(name='counts')
             .rename({'index': 'Date'}, axis=1)
```

```
              .sort_values('Date', ascending=True))
  xs = counted['Date']
  ys = counted['counts']
  plt.plot(xs, ys, label=series_name, color=palette[series_index % len(palette)])

fig, ax = plt.subplots(figsize=(10, 5.2), layout='constrained')
df_sorted = _df_0.sort_values('Date', ascending=True)
_plot_series(df_sorted, '')
sns.despine(fig=fig, ax=ax)
plt.xlabel('Date')
_ = plt.ylabel('count()')
```

```
from matplotlib import pyplot as plt
import seaborn as sns
def _plot_series(series, series_name, series_index=0):
  palette = list(sns.palettes.mpl_palette('Dark2'))
  counted = (series['Qty']
               .value_counts()
              .reset_index(name='counts')
              .rename({'index': 'Qty'}, axis=1)
              .sort_values('Qty', ascending=True))
  xs = counted['Qty']
  ys = counted['counts']
  plt.plot(xs, ys, label=series_name, color=palette[series_index % len(palette)])

fig, ax = plt.subplots(figsize=(10, 5.2), layout='constrained')
df_sorted = _df_1.sort_values('Qty', ascending=True)
_plot_series(df_sorted, '')
sns.despine(fig=fig, ax=ax)
plt.xlabel('Qty')
_ = plt.ylabel('count()')
```

```
from matplotlib import pyplot as plt
import seaborn as sns
def _plot_series(series, series_name, series_index=0):
  palette = list(sns.palettes.mpl_palette('Dark2'))
  counted = (series['Amount']
               .value_counts()
              .reset_index(name='counts')
              .rename({'index': 'Amount'}, axis=1)
              .sort_values('Amount', ascending=True))
  xs = counted['Amount']
  ys = counted['counts']
  plt.plot(xs, ys, label=series_name, color=palette[series_index % len(palette)])

fig, ax = plt.subplots(figsize=(10, 5.2), layout='constrained')
df_sorted = _df_2.sort_values('Amount', ascending=True)
_plot_series(df_sorted, '')
sns.despine(fig=fig, ax=ax)
```

```
plt.xlabel('Amount')
_ = plt.ylabel('count()')
```

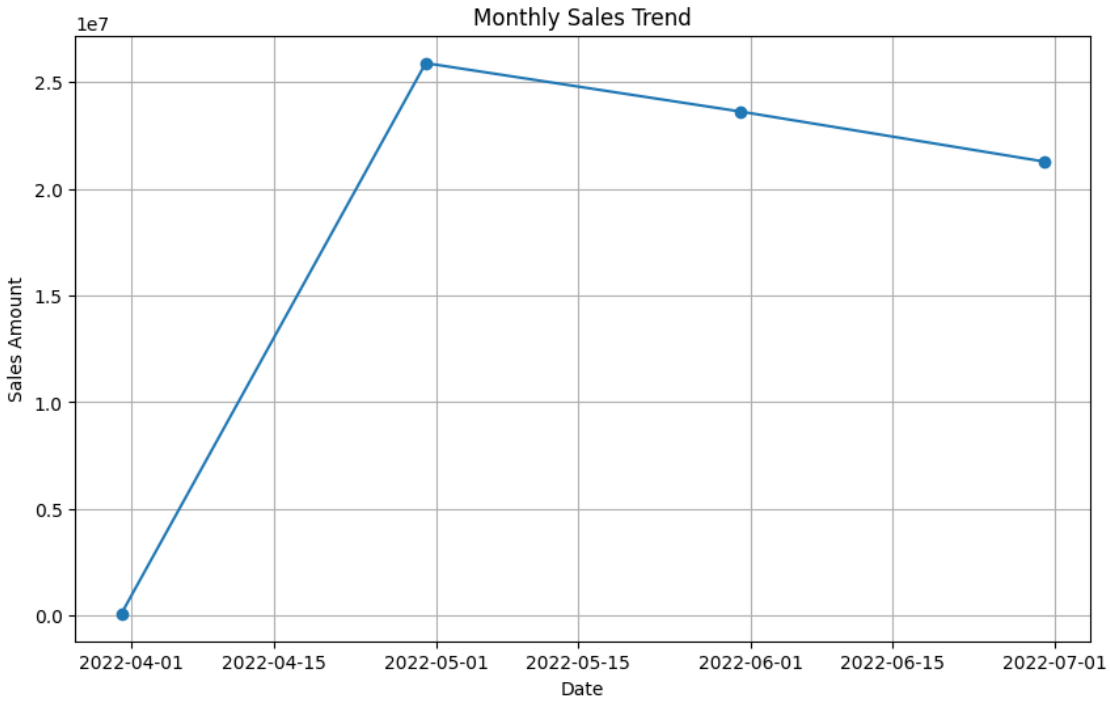
```
from matplotlib import pyplot as plt
import seaborn as sns
def _plot_series(series, series_name, series_index=0):
  palette = list(sns.palettes.mpl_palette('Dark2'))
  counted = (series['Total Sales']
               .value_counts()
             .reset_index(name='counts')
             .rename({'index': 'Total Sales'}, axis=1)
             .sort_values('Total Sales', ascending=True))
  xs = counted['Total Sales']
  ys = counted['counts']
  plt.plot(xs, ys, label=series_name, color=palette[series_index % len(palette)])

fig, ax = plt.subplots(figsize=(10, 5.2), layout='constrained')
df_sorted = _df_3.sort_values('Total Sales', ascending=True)
_plot_series(df_sorted, '')
sns.despine(fig=fig, ax=ax)
plt.xlabel('Total Sales')
_ = plt.ylabel('count()')
```

1.13 Insights and Observations

- year 2022: 70.87 million in total sales, with 108,470 units sold. The sales include 0.0008 million in B2B transactions.

1.14 Visualize Sales Trends

[ ]:
```
plt.figure(figsize = (10,6))
plt.plot(monthly_sales.index, monthly_sales['Total Amount'], marker = 'o')
plt.title('Monthly Sales Trend')
plt.xlabel('Date')
plt.ylabel('Sales Amount')
plt.grid(True)
plt.show()
```



- Overall, sales peaked in April and then gradually declined over the following months.

1.15 Analyze Sales by Categories

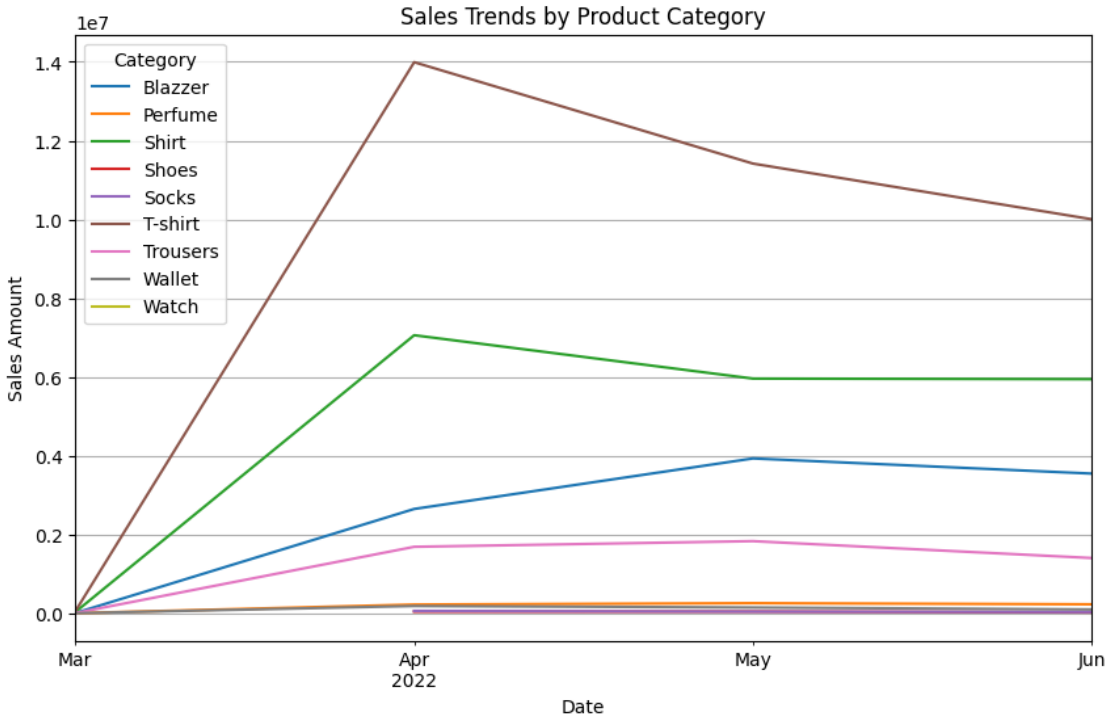
[ ]:
```
## Group Sales by product category
Category_sales = amazons1_copy.groupby(['Category', pd.Grouper(key = 'Date',␣
↪freq= 'M')])['Total Amount'].sum().unstack()
```

[ ]:
```
Category_sales
```

[ ]:
```
Date      2022-03-31  2022-04-30  2022-05-31  2022-06-30
Category
Blazzer       5479.0   2647107.0   3930309.0   3545296.0
Perfume       1099.0    220401.0    255662.0    226770.0
Shirt        30343.0   7058269.0   5957438.0   5942321.0
Shoes            NaN     41128.0     43218.0     23993.0
Socks            NaN     58656.0     41206.0     31842.0
T-shirt      50837.0  13990667.0  11416137.0  10002619.0
Trousers      4511.0   1686374.0   1830115.0   1401518.0
Wallet         280.0    183485.0    145146.0     96130.0
Watch            NaN         NaN         NaN       305.0
```

[ ]:
```
# Plotting sales trends by category
Category_sales.T.plot(kind='line', figsize=(10, 6))
```

```
plt.title('Sales Trends by Product Category')
plt.xlabel('Date')
plt.ylabel('Sales Amount')
plt.grid(True)
plt.show()
```



1.16 Insights and Observations

- T-shirts and Shirts consistently generated the highest sales, with T-shirts peaking in April.
- Blazzer and Trousers also saw significant sales but with more fluctuation.
- Perfume, Shoes, Socks, Wallet, and Watch had lower sales overall, with some categories like Shoes and Socks having no sales in March.

1.17 2. Product Analysis: Analyze the distribution of product categories, sizes, and quantities sold to identify popular products.

- Analyzing Distribution of product categories

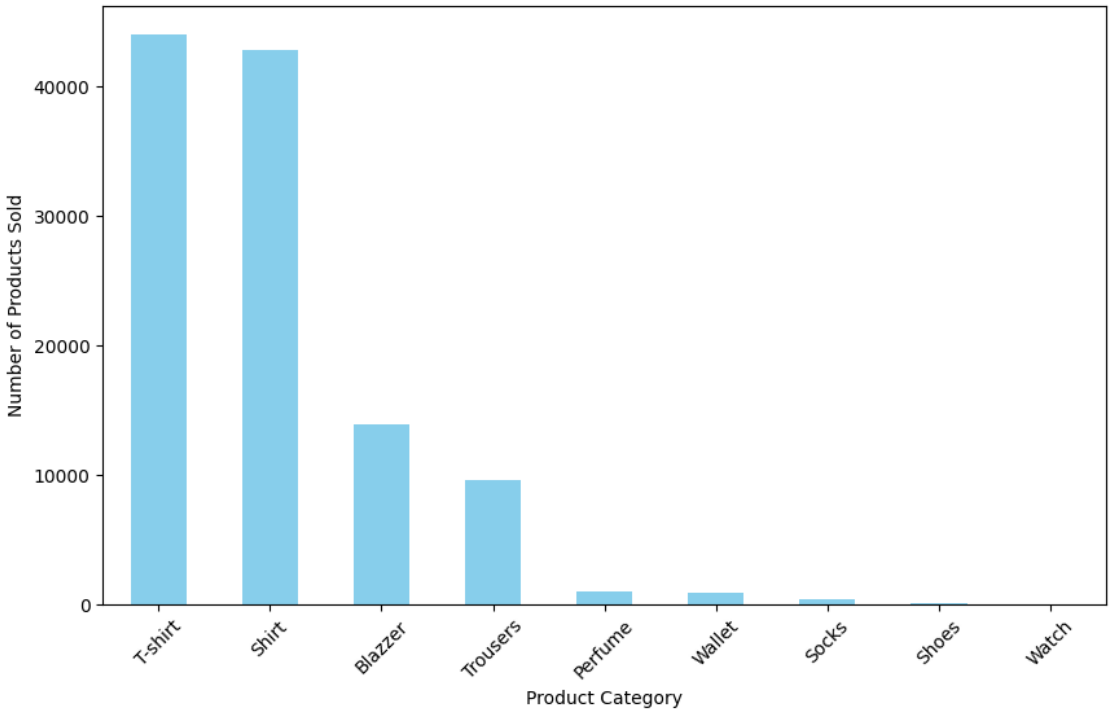
[ ]:
```
category_distribution = amazons1_copy['Category'].value_counts()
```

[ ]:
```
category_distribution
```

[ ]:
```
Category
T-shirt     44052
```

```
Shirt       42888
Blazzer     13938
Trousers     9572
Perfume      1017
Wallet        846
Socks         386
Shoes         134
Watch           1
Name: count, dtype: int64
```

[ ]:
```
## plotting the distribution of product category
plt.figure(figsize = (10,6))
category_distribution.plot(kind = 'bar', color = 'skyblue')
plt.xlabel('Product Category')
plt.ylabel('Number of Products Sold')
plt.xticks(rotation=45)
plt.show()
```



1.18 Insights and Observations

- T-shirts and Shirts are the top-selling categories, with T-shirts leading in quantity sold.
- Blazzer and Trousers follow, with significantly lower quantities compared to T-shirts and Shirts.
- Perfume and Wallet have modest sales, while Socks, Shoes, and Watch have very low quantities

sold.

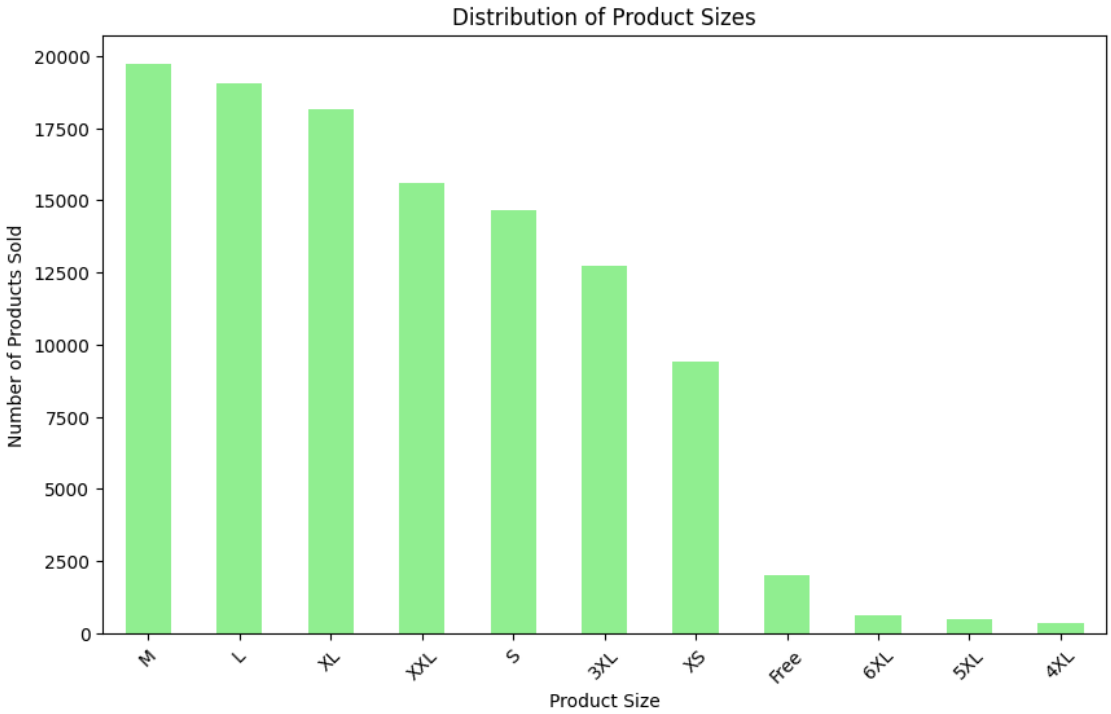
- Analyze Distribution of Product Sizes

[ ]:
```
size_distribution = amazons1_copy['Size'].value_counts()
```

[ ]:
```
size_distribution
```

[ ]:
```
Size
M       19731
L       19072
XL      18146
XXL     15595
S       14647
3XL     12717
XS       9418
Free     1998
6XL       643
5XL       496
4XL       371
Name: count, dtype: int64
```

[ ]:
```
## plotting the size distribution
plt.figure(figsize = (10,6))
size_distribution.plot(kind = 'bar' , color = 'lightgreen' )
plt.title('Distribution of Product Sizes')
plt.xlabel('Product Size')
plt.ylabel('Number of Products Sold')
plt.xticks(rotation=45)
plt.show()
```



1.19 Insights and Observation

- Medium (M) and Large (L) are the most popular sizes, with nearly equal high quantities sold.
- Extra Large (XL) and Double Extra Large (XXL) follow, showing strong sales.
- Small (S) and Triple Extra Large (3XL) also have substantial quantities sold.
- Extra Small (XS) and Free Size have lower sales compared to the other sizes.
- 6XL, 5XL, and 4XL are the least sold sizes, with significantly lower quantities.

- Analyze Quantities sold by product category

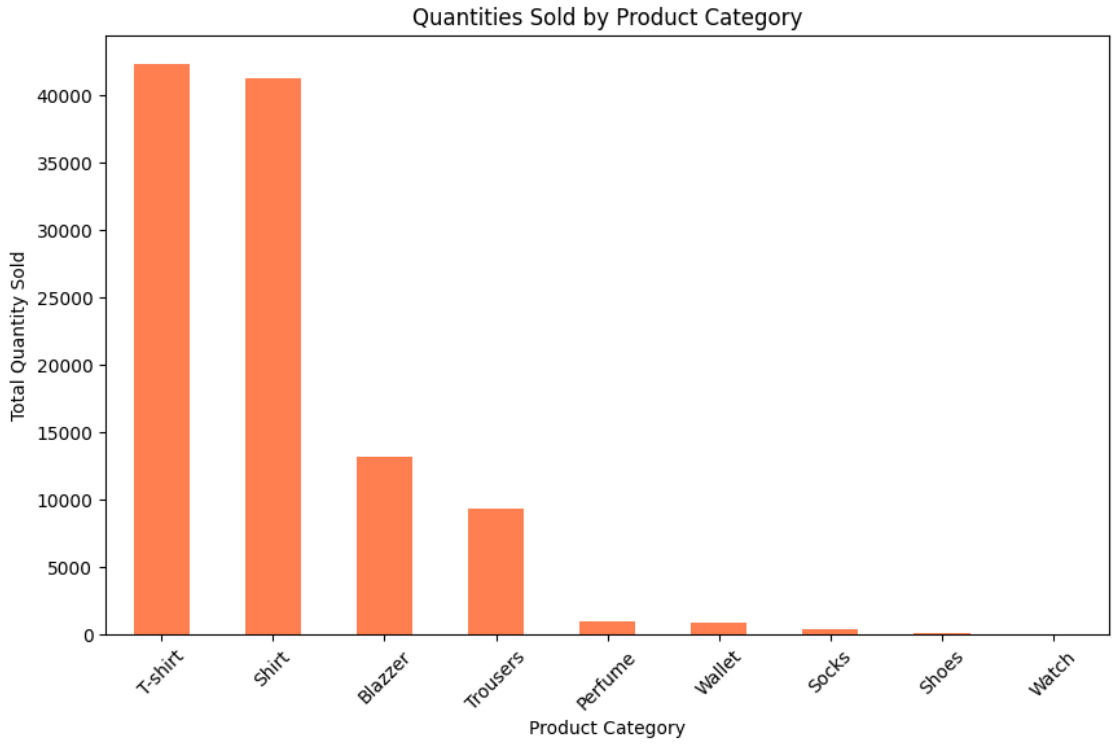
[ ]:
```
quantity_by_category = amazons1_copy.groupby('Category')['Qty'].sum().
↪sort_values(ascending = False)
```

[ ]:
```
quantity_by_category
```

[ ]:
```
Category
T-shirt     42364
Shirt       41290
Blazzer     13208
Trousers     9320
Perfume       979
Wallet        811
Socks         367
Shoes         130
```

```
Watch           1
Name: Qty, dtype: int64
```

[ ]:
```
## plotting quantity sold by product category
plt.figure(figsize = (10,6))
quantity_by_category.plot(kind = 'bar', color = 'coral')
plt.title('Quantities Sold by Product Category')
plt.xlabel('Product Category')
plt.ylabel('Total Quantity Sold')
plt.xticks(rotation = 45)
plt.show()
```



1.20 Insights and Observation

- T-shirts and Shirts are the top-selling categories, with T-shirts leading in quantity sold.
- Blazzer follows, with a significantly lower quantity compared to T-shirts and Shirts.
- Trousers also show a notable amount of sales.
- Perfume and Wallet have moderate quantities sold.
- Socks and Shoes have low sales, with Shoes having the least quantity sold.
- Watch has minimal sales with only 1 unit sold.

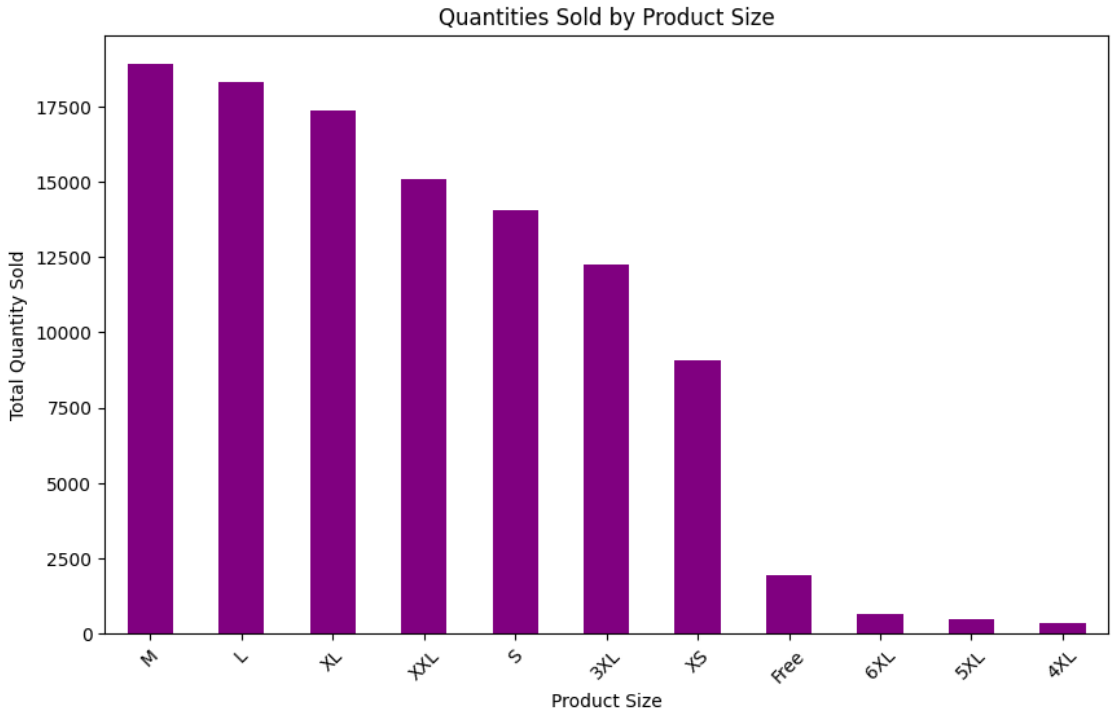
- Analyze Quantities Sold by Product Size

[ ]:
```
quantity_by_size = amazons1_copy.groupby('Size')['Qty'].sum().
↪sort_values(ascending = False)
```

[ ]:
```
quantity_by_size
```

[ ]:
```
Size
M       18922
L       18325
XL      17393
XXL     15083
S       14048
3XL     12234
XS       9076
Free     1921
6XL       624
5XL       483
4XL       361
Name: Qty, dtype: int64
```

[ ]:
```
## plotting the quantities sold by product size
plt.figure(figsize = (10,6))
quantity_by_size.plot(kind = 'bar', color = 'purple')
plt.title('Quantities Sold by Product Size')
plt.xlabel('Product Size')
plt.ylabel('Total Quantity Sold')
plt.xticks(rotation = 45)
plt.show()
```



1.21 Insights and Observations

- Medium (M) and Large (L) sizes are the most popular, with Medium slightly ahead in quantity sold.
- Extra Large (XL) and Double Extra Large (XXL) follow, with substantial sales.
- Small (S) and Triple Extra Large (3XL) have strong sales, though less than the larger sizes.
- Extra Small (XS) and Free Size have lower sales volumes.
- 6XL, 5XL, and 4XL are the least sold sizes, with significantly fewer units sold.

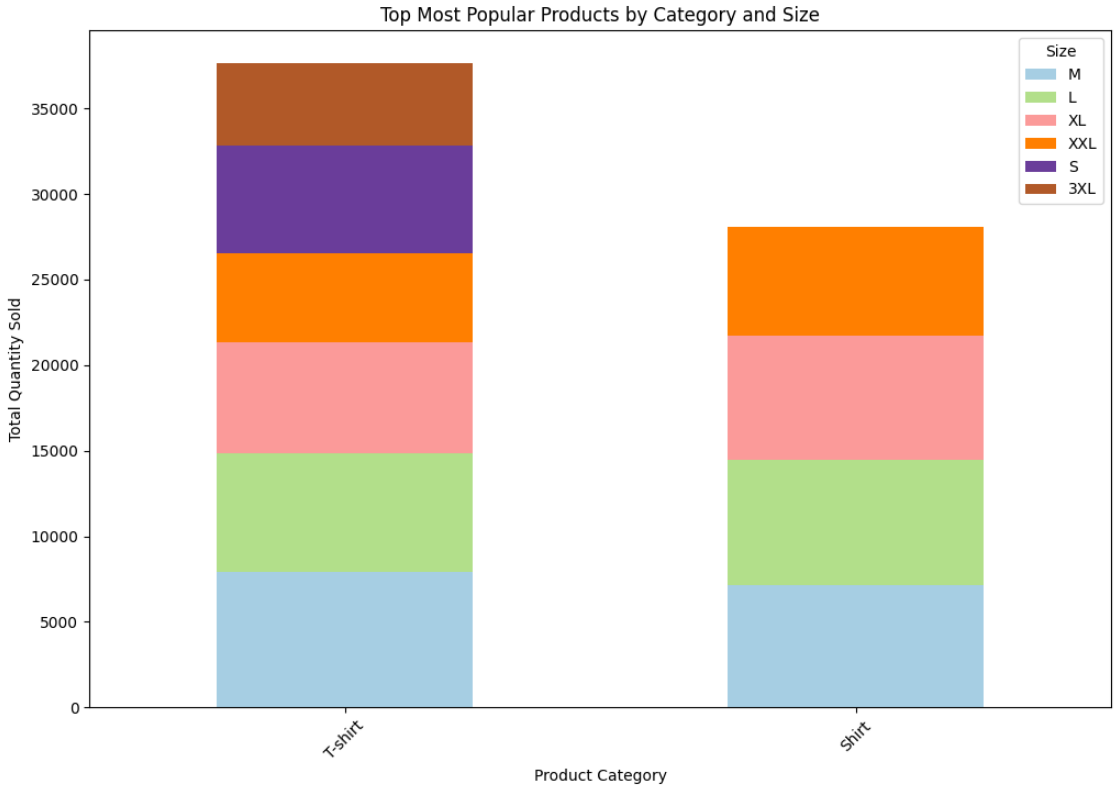
- Identify the most Popular Products

[ ]:
```
popular_products = amazons1_copy.groupby(['Category', 'Size'])['Qty'].sum().
↪sort_values(ascending = False)
popular_products.head(10) ## Top Products
```

[ ]:
```
Category  Size
T-shirt   M       7911
Shirt     L       7358
          XL      7223
          M       7142
T-shirt   L       6938
          XL      6466
Shirt     XXL     6368
T-shirt   S       6293
```

```
XXL     5210
3XL     4852
Name: Qty, dtype: int64
```

[ ]:
```
## visualize top popular Products
popular_products.head(10).unstack().plot(kind = 'bar', stacked = True, figsize␣
↪= (12,8), colormap = 'Paired' )
plt.title('Top Most Popular Products by Category and Size')
plt.xlabel('Product Category')
plt.ylabel('Total Quantity Sold')
plt.xticks(rotation = 45)
plt.show()
```



1.22 Insights and Observations

- T-shirts are sold in all sizes, with Medium and Large being the top-selling sizes.
- Shirts also have substantial sales across multiple sizes, with Large and Extra Large leading in quantity sold.
- Both T-shirts and Shirts show strong performance in sizes M and L, while T-shirts have notable sales in Small and XXL as well.

1.23 Correlation Trends

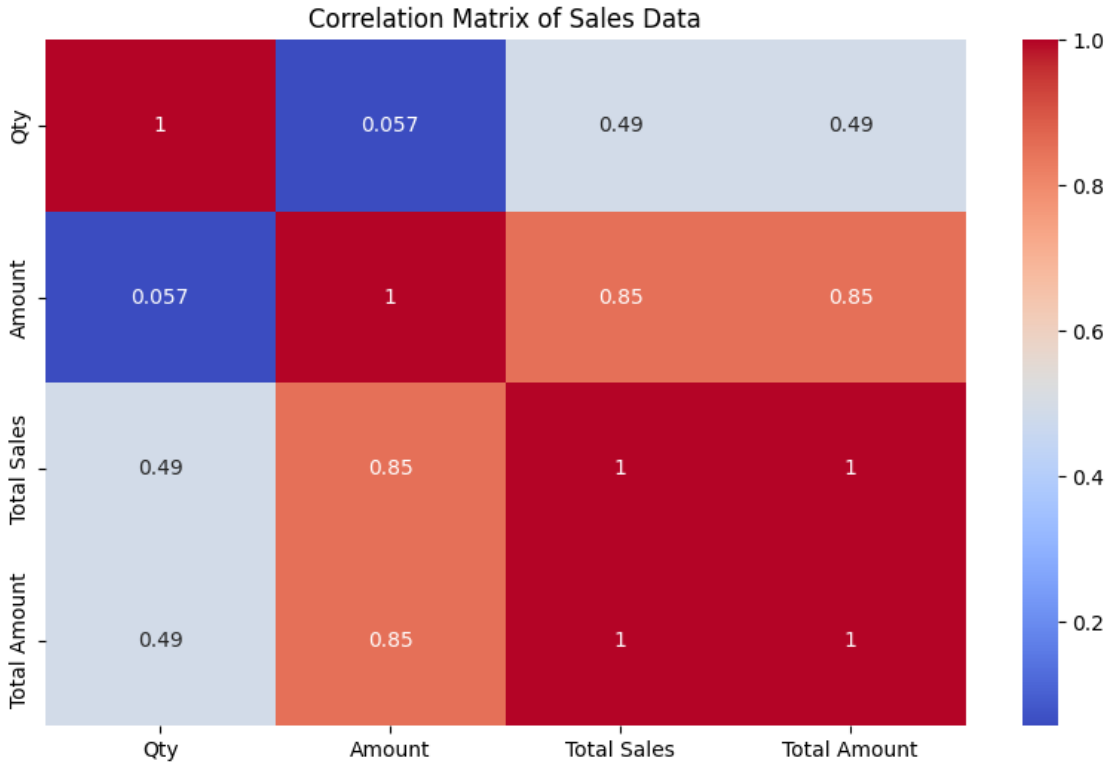
```
[ ]: # Select only numeric columns for correlation calculation
numeric_columns = amazons1_copy.select_dtypes(include=np.number).columns
correlation_matrix = amazons1_copy[numeric_columns].corr()
```

```
[ ]: correlation_matrix
```

```
[ ]:
```

	Qty	Amount	Total Sales	Total Amount
Qty	1.000000	0.056792	0.488467	0.488467
Amount	0.056792	1.000000	0.849273	0.849273
Total Sales	0.488467	0.849273	1.000000	1.000000
Total Amount	0.488467	0.849273	1.000000	1.000000

```
[ ]: plt.figure(figsize=(10, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix of Sales Data')
plt.show()
```



1.24 Insights and Observation

- Total Sales and Amount have the strongest positive correlation, indicating that the total amount generated is closely related to Total Sales.

- Quantity Sold has a moderate correlation with Total Sales, suggesting that higher quantities tend to drive higher total sales.
- B2B Sales have very weak correlations with the other variables, indicating that B2B transactions have minimal impact on the overall metrics in this dataset.

1.25 3. Fulfillment Analysis: Investigate the fulfillment methods used and their effectiveness in delivering orders.

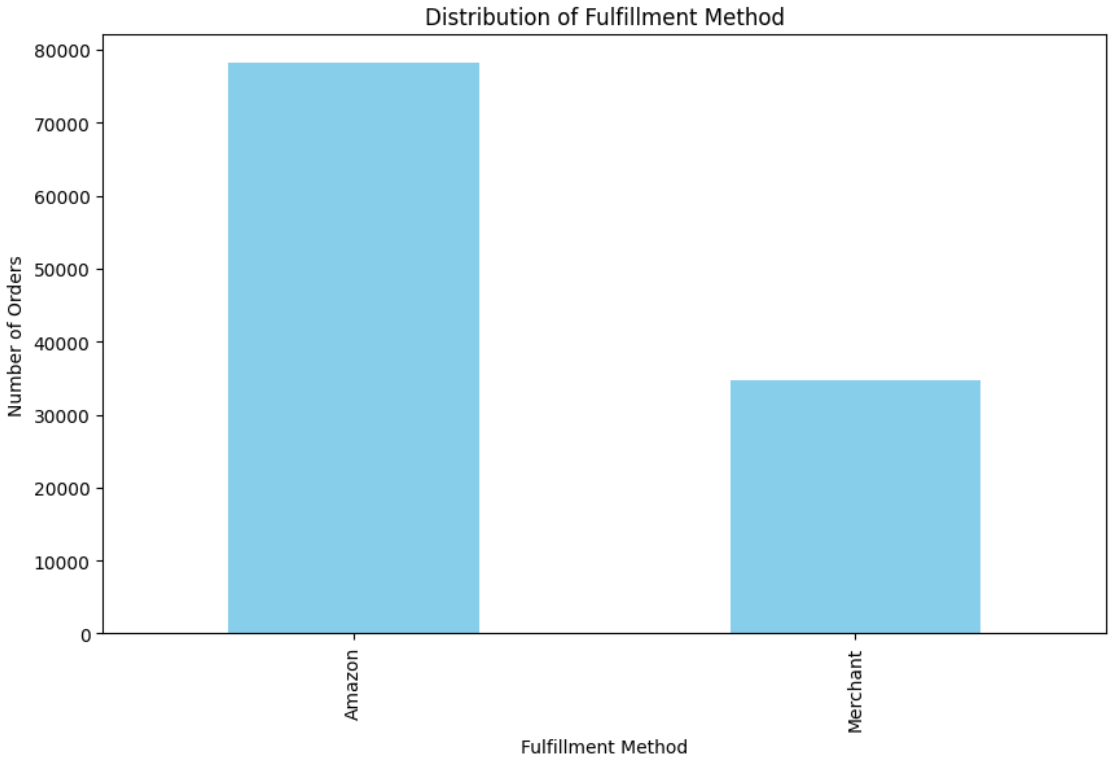
- Analyze Fulfillment methods Distribution

[ ]:
```
fulfillment_distribution = amazons1_copy['Fulfilment'].value_counts()
```

[ ]:
```
fulfillment_distribution
```

[ ]:
```
Fulfilment
Amazon      78164
Merchant    34670
Name: count, dtype: int64
```

[ ]:
```
## plotting the fulfillment distribution
plt.figure(figsize = (10,6))
fulfillment_distribution.plot(kind = 'bar', color = 'skyblue')
plt.title('Distribution of Fulfillment Method')
plt.xlabel('Fulfillment Method')
plt.ylabel('Number of Orders')
plt.show()
```



1.26 Insights and Observation

- Amazon is the dominant fulfillment channel, with significantly higher sales compared to Merchant.
- Merchant fulfillment accounts for a smaller portion of the total sales, indicating that Amazon is the primary channel for these sales.

[ ]:
```
amazons1_copy.head(3)
```

[ ]:
```
                  Order ID       Date Fulfilment Sales Channel    Service  \
48971  404-1445673-1345134 2022-03-31   Merchant        Amazon   Standard
48997  402-0339394-3540335 2022-03-31     Amazon        Amazon  Expedited
48998  408-6597776-7485121 2022-03-31     Amazon        Amazon  Expedited

       Category Size Courier Status  Qty currency  …       City  \
48971     Shirt    L        Shipped    1      INR  …    KOLKATA
48997   T-shirt   XL        Shipped    1      INR  …  GHAZIABAD
48998  Trousers   XS        Shipped    1      INR  …      DELHI

               State Postal Code Country    B2B  fulfilled-by    status  \
48971    WEST BENGAL    700124.0   INDIA  False     Easy Ship  Shipped
48997  UTTAR PRADESH    201005.0   INDIA  False            NA   Shipped
48998      NEW DELHI    110016.0   INDIA  False            NA   Shipped

             order status Total Sales  Total Amount
48971   Delivered to Buyer       495.0         495.0
48997              loading       688.0         688.0
48998              loading       354.0         354.0

[3 rows x 21 columns]
```

[ ]:
```
amazons1_copy['Courier Status'].unique()
```

[ ]:
```
array(['Shipped', 'On the Way', 'Unshipped'], dtype=object)
```

[ ]:
```
amazons1_copy['status'].unique()
```

[ ]:
```
array(['Shipped ', 'Shipped', 'Cancelled', 'Pending', 'Pending '],
      dtype=object)
```

[ ]:
```
amazons1_copy['order status'].unique()
```

[ ]:
```
array([' Delivered to Buyer', 'loading', ' Returned to Seller',
       ' Rejected by Buyer', ' Picked Up', ' Returning to Seller',
       ' Out for Delivery', ' Lost in Transit', ' Damaged',
```

```
       ' Waiting for Pick Up'], dtype=object)
```

```
[ ]: amazons1_copy = amazons1_copy.rename(columns= {'status': 'Pre order status',␣
      ↪'order status': 'Post order status'})
```

```
[ ]: amazons1_copy.head(2)
```

```
[ ]:                  Order ID       Date Fulfilment Sales Channel    Service  \
     48971  404-1445673-1345134 2022-03-31   Merchant        Amazon   Standard
     48997  402-0339394-3540335 2022-03-31     Amazon        Amazon  Expedited

           Category Size Courier Status  Qty currency  …       City  \
     48971    Shirt    L        Shipped    1      INR  …    KOLKATA
     48997  T-shirt   XL        Shipped    1      INR  …  GHAZIABAD

                    State Postal Code Country    B2B  fulfilled-by  \
     48971    WEST BENGAL    700124.0   INDIA  False     Easy Ship
     48997  UTTAR PRADESH    201005.0   INDIA  False            NA

           Pre order status    Post order status Total Sales  Total Amount
     48971         Shipped    Delivered to Buyer       495.0         495.0
     48997          Shipped              loading       688.0         688.0

     [2 rows x 21 columns]
```

```
[ ]: amazons1_copy = amazons1_copy.rename(columns = {'Post order status': 'order␣
      ↪status'})
     amazons1_copy = amazons1_copy.drop('Pre order status', axis = 1)
```

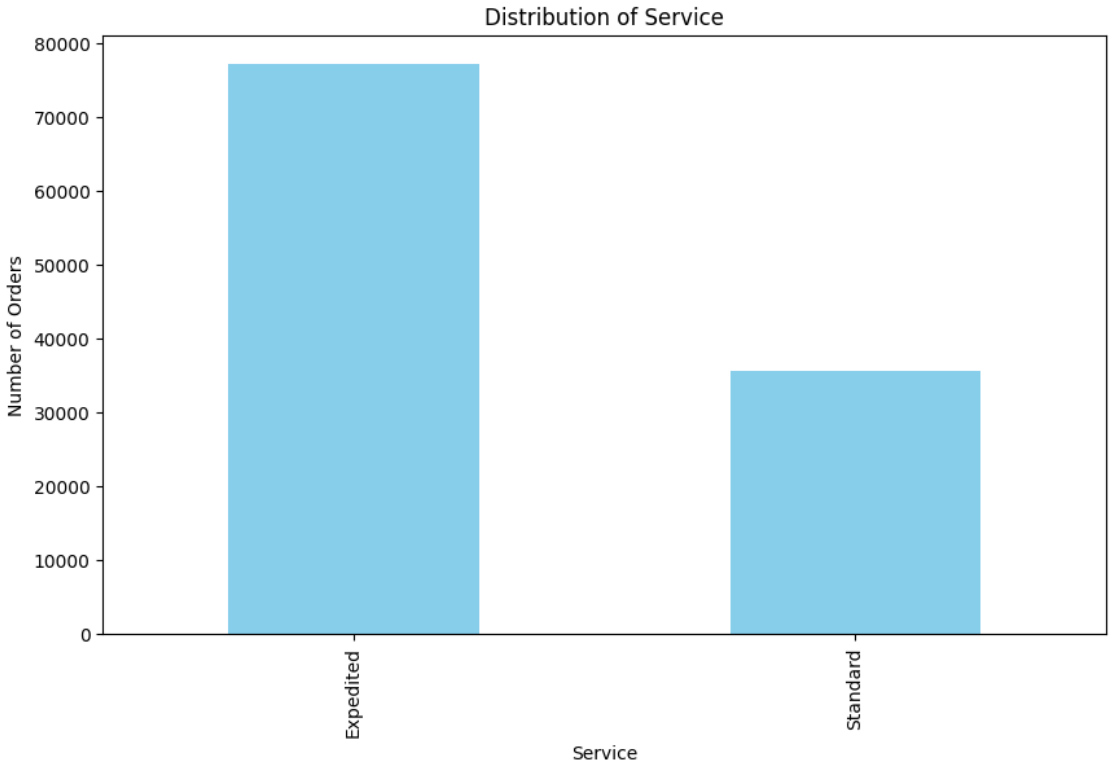
- Analyze service distribution

```
[ ]: service_distribution = amazons1_copy['Service'].value_counts()
```

```
[ ]: service_distribution
```

```
[ ]: Service
     Expedited    77251
     Standard     35583
     Name: count, dtype: int64
```

```
[ ]: ## plot service distribution
     plt.figure(figsize = (10,6))
     service_distribution.plot(kind = 'bar', color = 'skyblue')
     plt.title('Distribution of Service')
     plt.xlabel('Service')
     plt.ylabel('Number of Orders')
     plt.show()
```



1.27 Insights and Observation

- Expedited service is the preferred choice, with a significantly higher number of units fulfilled compared to Standard service.
- Standard service accounts for a smaller portion of the total sales, indicating that expedited delivery is more commonly used or preferred.
- Analyze Courier Status Distribution

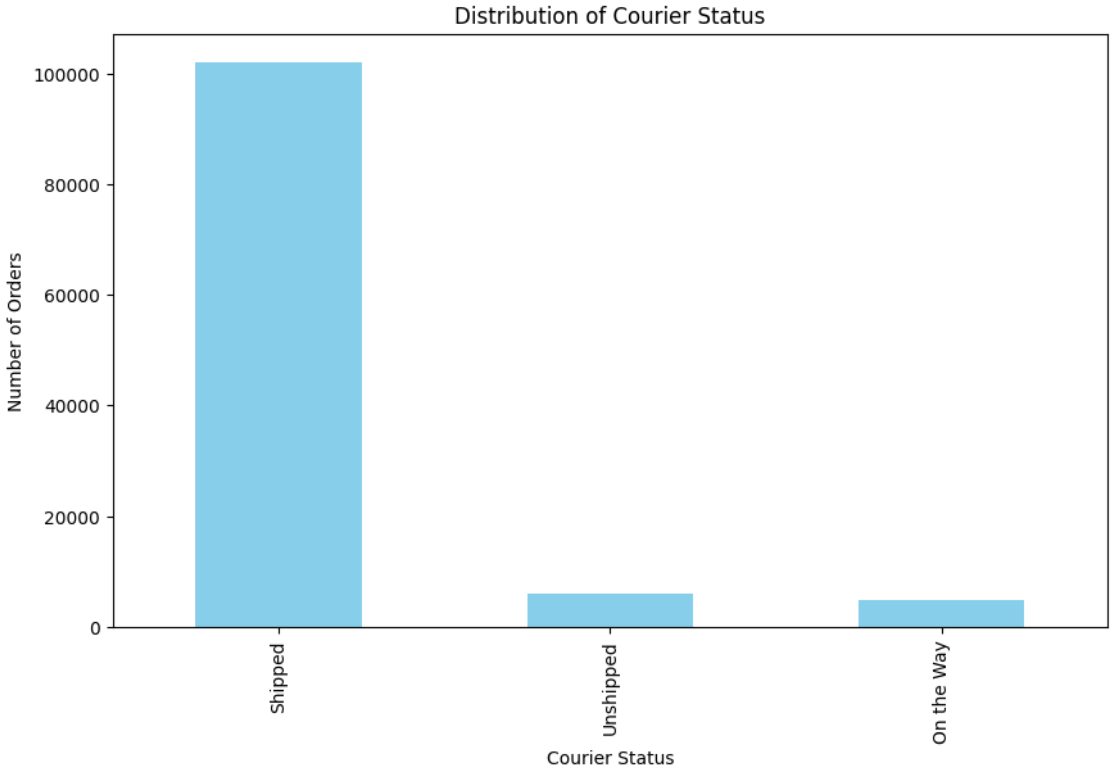
[ ]:
```
courier_status_distribution = amazons1_copy['Courier Status'].value_counts()
```

[ ]:
```
courier_status_distribution
```

[ ]:
```
Courier Status
Shipped       102049
Unshipped       6058
On the Way      4727
Name: count, dtype: int64
```

[ ]:
```
## plotting the courier status distribution
plt.figure(figsize = (10,6))
courier_status_distribution.plot(kind = 'bar', color = 'skyblue')
plt.title('Distribution of Courier Status')
```

```
plt.xlabel('Courier Status')
plt.ylabel('Number of Orders')
plt.show()
```



1.28 Insights and Observations

- Shipped is the predominant status, with a significantly higher number of units compared to Unshipped and On the Way.
- Unshipped and On the Way have much lower quantities, indicating that most of the orders have already been shipped.

- Order Status Distribution

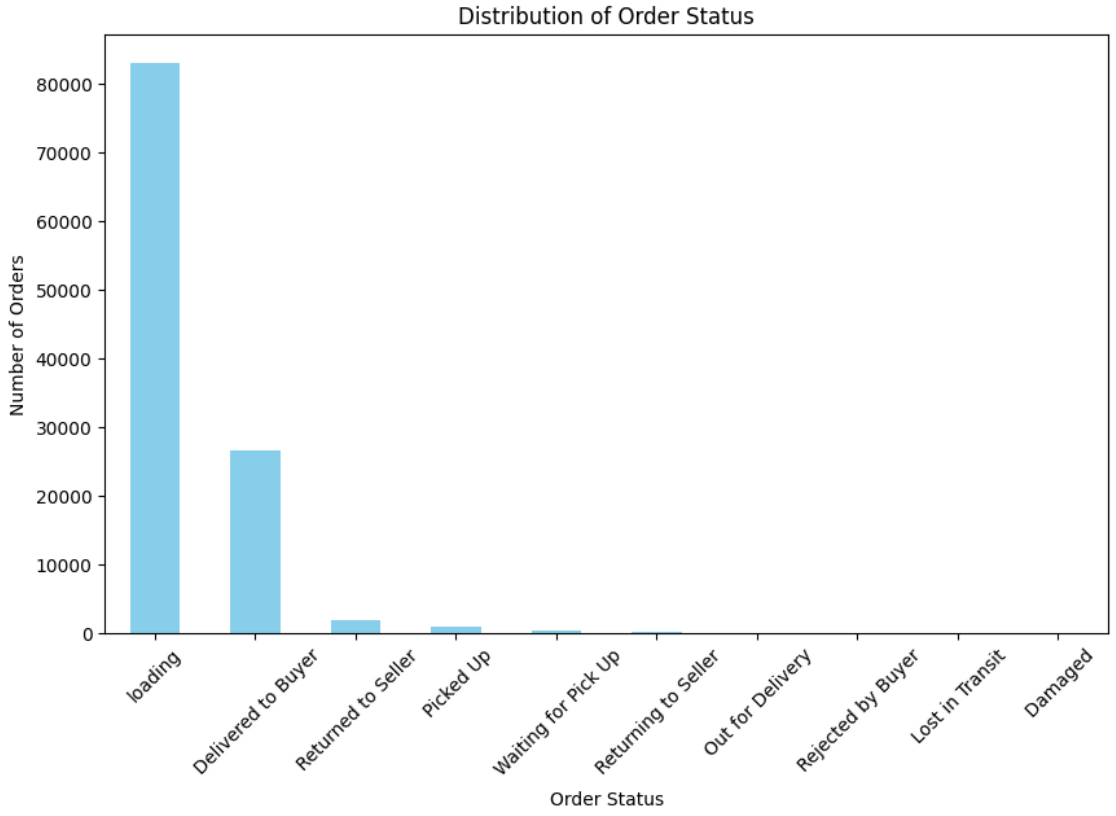
[ ]:
```
order_status_distribution = amazons1_copy['order status'].value_counts()
```

[ ]:
```
order_status_distribution
```

[ ]:
```
order status
loading                 83110
 Delivered to Buyer     26517
 Returned to Seller      1849
 Picked Up                918
 Waiting for Pick Up      262
```

```
Returning to Seller      130
Out for Delivery          32
Rejected by Buyer         11
Lost in Transit            4
Damaged                    1
Name: count, dtype: int64
```

[ ]:
```
## plotting the pre order status distribution
plt.figure(figsize = (10,6))
order_status_distribution.plot(kind = 'bar', color = 'skyblue')
plt.title('Distribution of Order Status')
plt.xlabel('Order Status')
plt.ylabel('Number of Orders')
plt.xticks(rotation = 45)
plt.show()
```



1.29 Insights and Observation

- Loading has the highest quantity, indicating that a large number of orders are still in the process of being prepared or shipped.
- Delivered to Buyer shows a significant number of completed transactions.

- Returned to Seller and Picked Up have relatively small quantities compared to the other statuses.
- Waiting for Pick Up, Returning to Seller, and other statuses such as Out for Delivery, Rejected by Buyer, Lost in Transit, and Damaged have minimal quantities, indicating less frequent occurrences.

1.29.1 Analyze Delivery Effectiveness

- On Time Delivery Rate

[ ]:
```
amazons1_copy.head(4)
```

[ ]:
```
                  Order ID       Date Fulfilment Sales Channel    Service  \
48971  404-1445673-1345134 2022-03-31   Merchant        Amazon   Standard
48997  402-0339394-3540335 2022-03-31     Amazon        Amazon  Expedited
48998  408-6597776-7485121 2022-03-31     Amazon        Amazon  Expedited
48999  404-1415044-3213110 2022-03-31     Amazon        Amazon  Expedited

       Category Size Courier Status  Qty currency  Amount       City  \
48971     Shirt    L        Shipped    1      INR   495.0    KOLKATA
48997   T-shirt   XL        Shipped    1      INR   688.0  GHAZIABAD
48998  Trousers   XS        Shipped    1      INR   354.0      DELHI
48999   T-shirt  3XL        Shipped    1      INR   698.0      SIKAR

               State Postal Code Country    B2B fulfilled-by  \
48971    WEST BENGAL    700124.0   INDIA  False    Easy Ship
48997  UTTAR PRADESH    201005.0   INDIA  False           NA
48998      NEW DELHI    110016.0   INDIA  False           NA
48999      RAJASTHAN    332001.0   INDIA  False           NA

              order status  Total Sales  Total Amount
48971   Delivered to Buyer        495.0         495.0
48997              loading        688.0         688.0
48998              loading        354.0         354.0
48999              loading        698.0         698.0
```

[ ]:
```
amazons1_copy['order status'].unique()
```

[ ]:
```
array([' Delivered to Buyer', 'loading', ' Returned to Seller',
       ' Rejected by Buyer', ' Picked Up', ' Returning to Seller',
       ' Out for Delivery', ' Lost in Transit', ' Damaged',
       ' Waiting for Pick Up'], dtype=object)
```

[ ]:
```
amazons1_copy['order status'] = amazons1_copy['order status'].str.strip()
```

[ ]:
```

```

```
amazons1_copy['On Time Delivery'] = amazons1_copy['order status'].
 ↪map({'Delivered to Buyer': 1, 'loading':0, 'Returned to Seller':0, 'Rejected␣
 ↪by Buyer': 0, 'Picked Up':0, 'Returning to Seller': 0,'Out for Delivery': 0,␣
 ↪'Lost in Transit': 0, 'Damaged':0,'Waiting for Pick Up':0})
```

[ ]:
```
amazons1_copy.head(2)
```

[ ]:
```
                  Order ID       Date Fulfilment Sales Channel    Service  \
48971  404-1445673-1345134 2022-03-31   Merchant        Amazon   Standard
48997  402-0339394-3540335 2022-03-31     Amazon        Amazon  Expedited

      Category Size Courier Status  Qty currency  …       City  \
48971    Shirt    L        Shipped    1      INR  …    KOLKATA
48997  T-shirt   XL        Shipped    1      INR  …  GHAZIABAD

               State Postal Code Country    B2B  fulfilled-by  \
48971    WEST BENGAL    700124.0   INDIA  False     Easy Ship
48997  UTTAR PRADESH    201005.0   INDIA  False            NA

             order status Total Sales  Total Amount  On Time Delivery
48971  Delivered to Buyer       495.0         495.0                 1
48997             loading       688.0         688.0                 0

[2 rows x 21 columns]
```

[ ]:
```
amazons1_copy['On Time Delivery'].unique()
```

[ ]:
```
array([1, 0])
```

[ ]:
```
## Now we find the on time delivery rate
on_time_delivery = amazons1_copy.groupby('Fulfilment')['On Time Delivery'].
 ↪mean()*100
```

[ ]:
```
on_time_delivery
```

[ ]:
```
Fulfilment
Amazon       0.000000
Merchant    76.483992
Name: On Time Delivery, dtype: float64
```

[ ]:
```
## plotting on time delivery rate by fulfilment method
plt.figure(figsize = (10,6))
on_time_delivery.plot(kind = 'bar', color = 'coral')
plt.title('On-Time Delivery Rate by Fulfillment Method')
plt.xlabel('Fulfillment Method')
plt.ylabel('On-Time Delivery Rate (%)')
plt.xticks(rotation=45)
```

```
plt.show()
```

1.30 Insights and Observations

- Amazon has an on-time delivery rate of 0%, which might indicate no data available or a reporting issue for this channel.
- Merchant has a substantial on-time delivery rate of 76.48%, suggesting that this channel performs well in terms of timely deliveries.

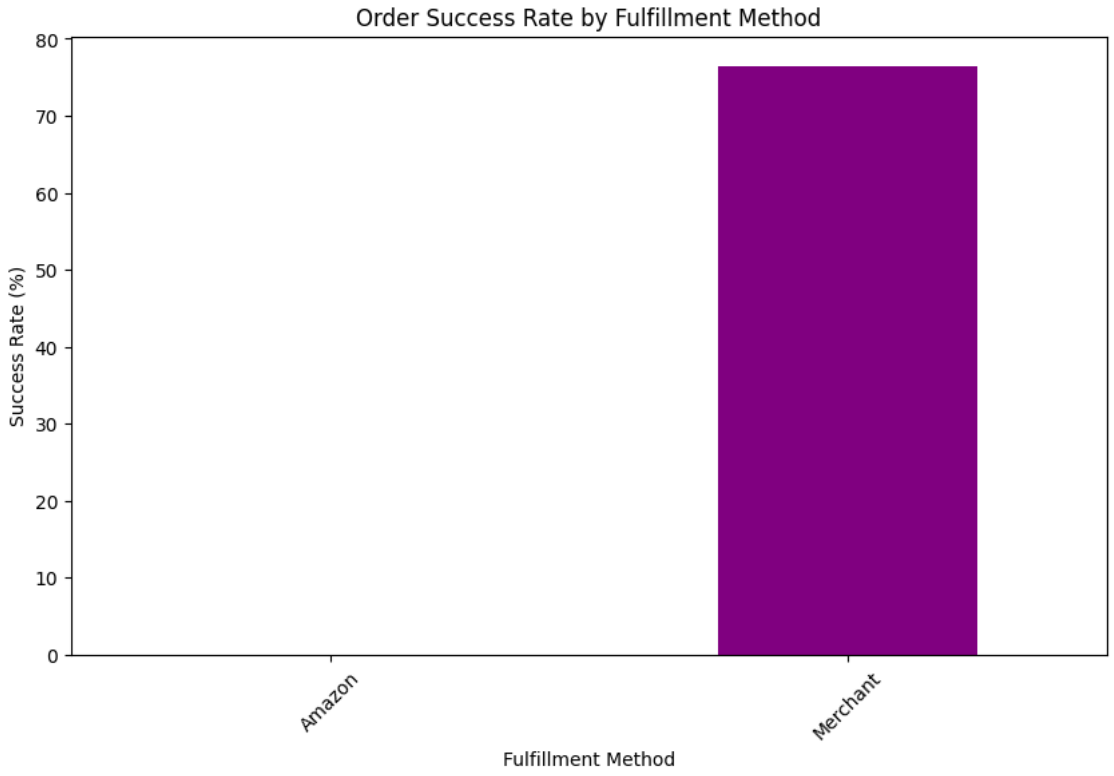
- Order Sucess Rate

[ ]:
```
success_rate = amazons1_copy[amazons1_copy['order status'] == 'Delivered to␣
↪Buyer'].groupby('Fulfilment')['Order ID'].count()/amazons1_copy.
↪groupby('Fulfilment')['Order ID'].count()*100
```

[ ]:
```
success_rate
```

[ ]:
```
Fulfilment
Amazon            NaN
Merchant    76.483992
Name: Order ID, dtype: float64
```

[ ]:
```
# Plotting order success rate by fulfillment method
plt.figure(figsize=(10, 6))
success_rate.plot(kind='bar', color='purple')
plt.title('Order Success Rate by Fulfillment Method')
plt.xlabel('Fulfillment Method')
plt.ylabel('Success Rate (%)')
plt.xticks(rotation=45)
plt.show()
```



1.31 Insights and Observation

- Amazon: No data available for the number of orders fulfilled through Amazon, indicating either missing data or no orders recorded for this channel.
- Merchant: A total of 76.48 orders were fulfilled, though the fractional value suggests it might be an average or aggregate figure.

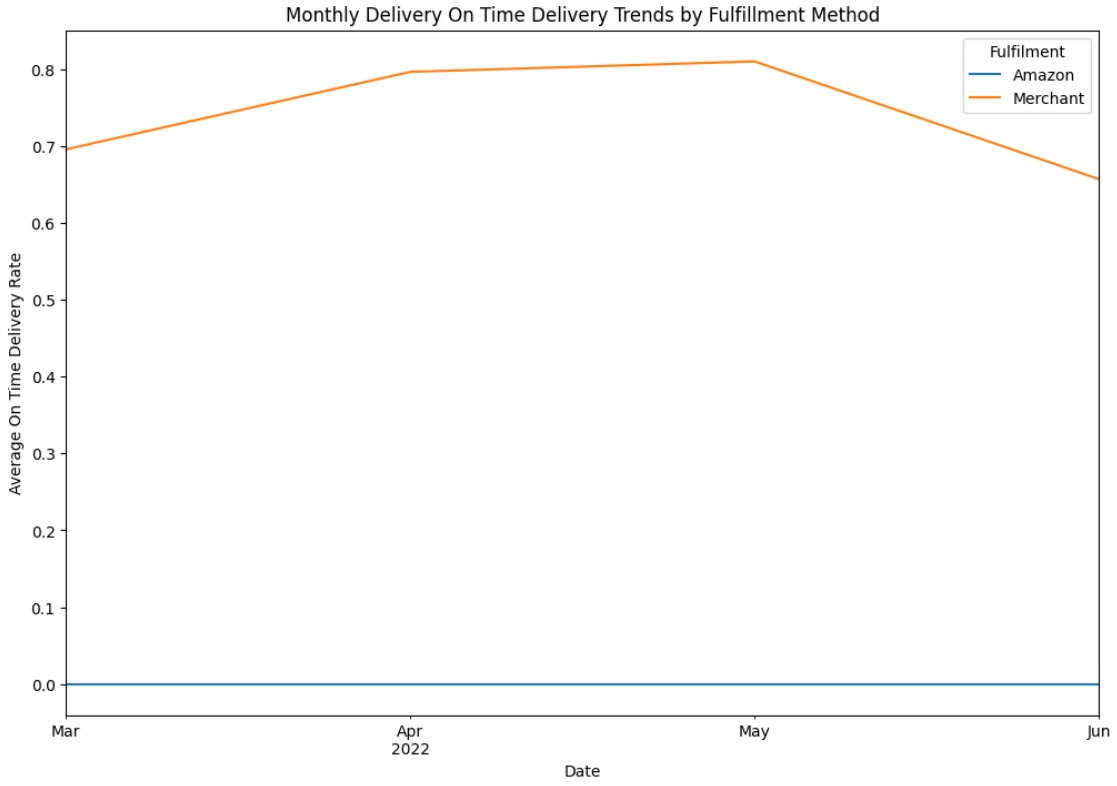
1.31.1 Analyze Trends over Time

[ ]:
```
monthly_trends = amazons1_copy.groupby([pd.Grouper(key = 'Date', freq = 'M'),␣
↪'Fulfilment'])['On Time Delivery'].mean().unstack()
```

[ ]:
```
monthly_trends
```

[ ]:
```
Fulfilment  Amazon  Merchant
Date
2022-03-31     0.0  0.695652
2022-04-30     0.0  0.796481
2022-05-31     0.0  0.810061
2022-06-30     0.0  0.656955
```

[ ]:
```
# Plotting the trends
monthly_trends.plot(kind='line', figsize=(12, 8))
plt.title('Monthly Delivery On Time Delivery Trends by Fulfillment Method')
plt.xlabel('Date')
plt.ylabel('Average On Time Delivery Rate')
plt.show()
```



1.32 Insights and Observation

- Amazon shows a fulfillment rate of 0.0 across all dates, indicating either no data available, no orders fulfilled, or an issue with reporting for Amazon.
- Merchant shows varying fulfillment rates across the dates: March 2022: 69.6% April 2022: 79.6% May 2022: 81.0% June 2022: 65.7%
- Merchant's fulfillment rates are relatively high, with a peak in May and a decrease in June.

1.32.1 4. Geographical Analysis: Explore the geographical distribution of sales, focusing on states and cities.

- Aggregate Sales by State and City

[ ]:
```
amazons1_copy['State'].unique()
```

[ ]:
```
array(['WEST BENGAL', 'UTTAR PRADESH', 'NEW DELHI', 'RAJASTHAN', 'KERALA',
       'TAMIL NADU', 'KARNATAKA', 'HARYANA', 'TELANGANA', 'MAHARASHTRA',
       'BIHAR', 'GUJARAT', 'TRIPURA', 'ANDHRA PRADESH', 'CHANDIGARH',
       'UTTARAKHAND', 'JHARKHAND', 'JAMMU & KASHMIR', 'ODISHA',
       'CHHATTISGARH', 'MADHYA PRADESH', 'GOA', 'ASSAM', 'PUNJAB',
       'HIMACHAL PRADESH', 'MEGHALAYA', 'MANIPUR', 'ANDAMAN & NICOBAR',
       'ARUNACHAL PRADESH', 'DADRA AND NAGAR', 'SIKKIM', 'PUDUCHERRY',
       'NAGALAND', 'MIZORAM', 'LADAKH', 'NA', 'LAKSHADWEEP', 'ORISSA',
       'PONDICHERRY'], dtype=object)
```

[ ]:
```
amazons1_copy['State'] = amazons1_copy['State'].replace('NA', 'NAGALAND')
```

[ ]:
```
## Aggregate Sales by state
sales_by_state = amazons1_copy.groupby('State')['Total Amount'].sum().
 ↪reset_index()
```

[ ]:
```
sales_by_state
```

[ ]:

	State	Total Amount
0	ANDAMAN & NICOBAR	138623.0
1	ANDHRA PRADESH	2803870.0
2	ARUNACHAL PRADESH	85415.0
3	ASSAM	944576.0
4	BIHAR	1299735.0
5	CHANDIGARH	190573.0
6	CHHATTISGARH	522418.0
7	DADRA AND NAGAR	34976.0
8	GOA	592715.0
9	GUJARAT	2432284.0
10	HARYANA	2653209.0
11	HIMACHAL PRADESH	445436.0
12	JAMMU & KASHMIR	414821.0
13	JHARKHAND	832516.0
14	KARNATAKA	9566741.0
15	KERALA	3380506.0
16	LADAKH	36087.0
17	LAKSHADWEEP	2441.0
18	MADHYA PRADESH	1431410.0
19	MAHARASHTRA	12070829.0
20	MANIPUR	192491.0
21	MEGHALAYA	103896.0

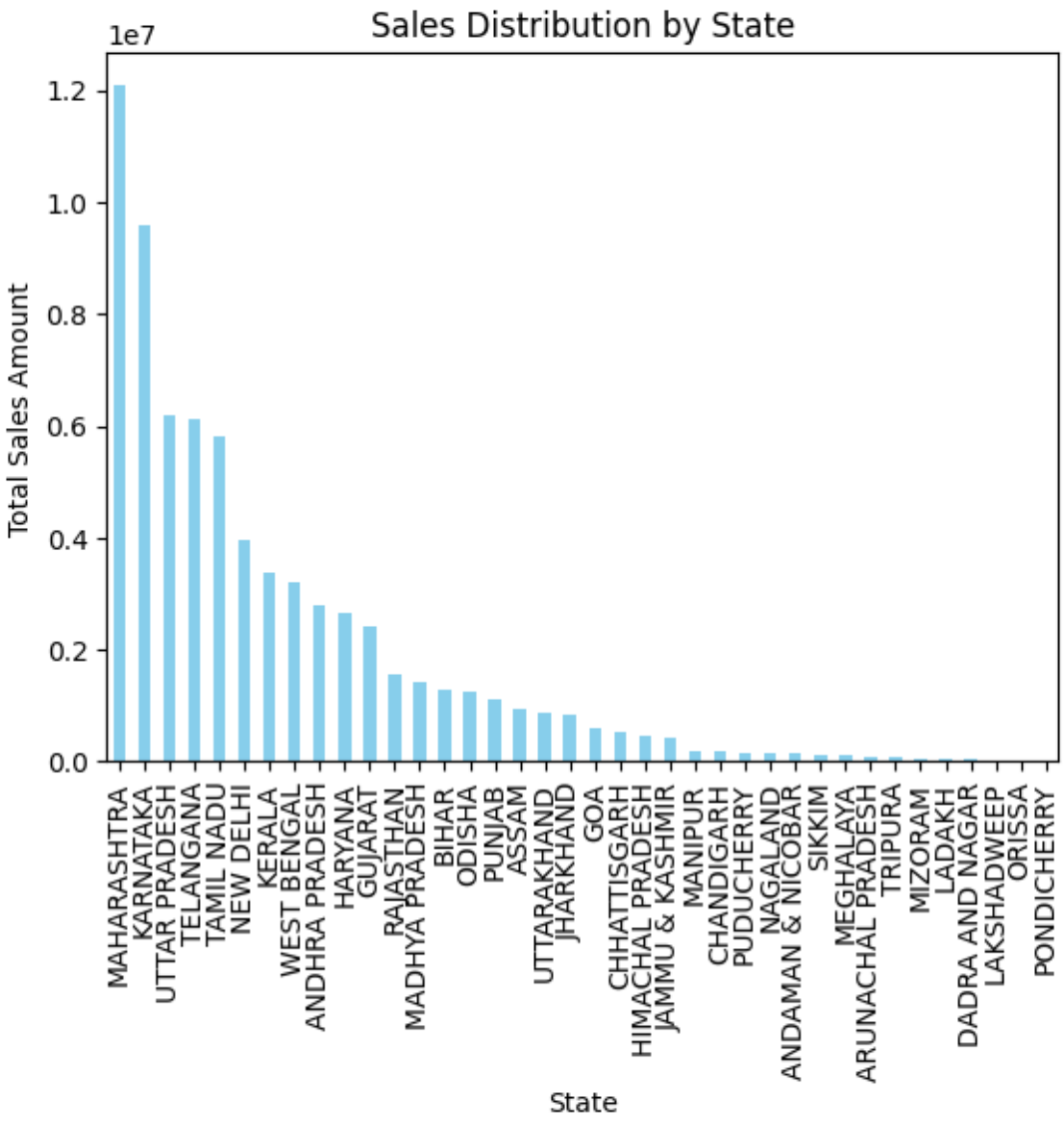
22	MIZORAM	38343.0
23	NAGALAND	148880.0
24	NEW DELHI	3976609.0
25	ODISHA	1261644.0
26	ORISSA	1737.0
27	PONDICHERRY	529.0
28	PUDUCHERRY	159516.0
29	PUNJAB	1127442.0
30	RAJASTHAN	1555231.0
31	SIKKIM	129598.0
32	TAMIL NADU	5806896.0
33	TELANGANA	6134184.0
34	TRIPURA	82779.0
35	UTTAR PRADESH	6189888.0
36	UTTARAKHAND	886892.0
37	WEST BENGAL	3192925.0

[ ]:
```
## Aggregate Sales by city
sales_by_city = amazons1_copy.groupby(['City'])['Total Amount'].sum().
↪reset_index()
```

- Visualize Sales Distribution by State

[ ]:
```
# Plotting sales distribution by state
plt.figure(figsize=(12,8))
sales_by_state.sort_values(by='Total Amount', ascending=False).plot(kind='bar',␣
↪x='State', y='Total Amount', color='skyblue', legend=False)
plt.title('Sales Distribution by State')
plt.xlabel('State')
plt.ylabel('Total Sales Amount')
plt.show()
```

```
<Figure size 1200x800 with 0 Axes>
```



1.33 Insights and Observations

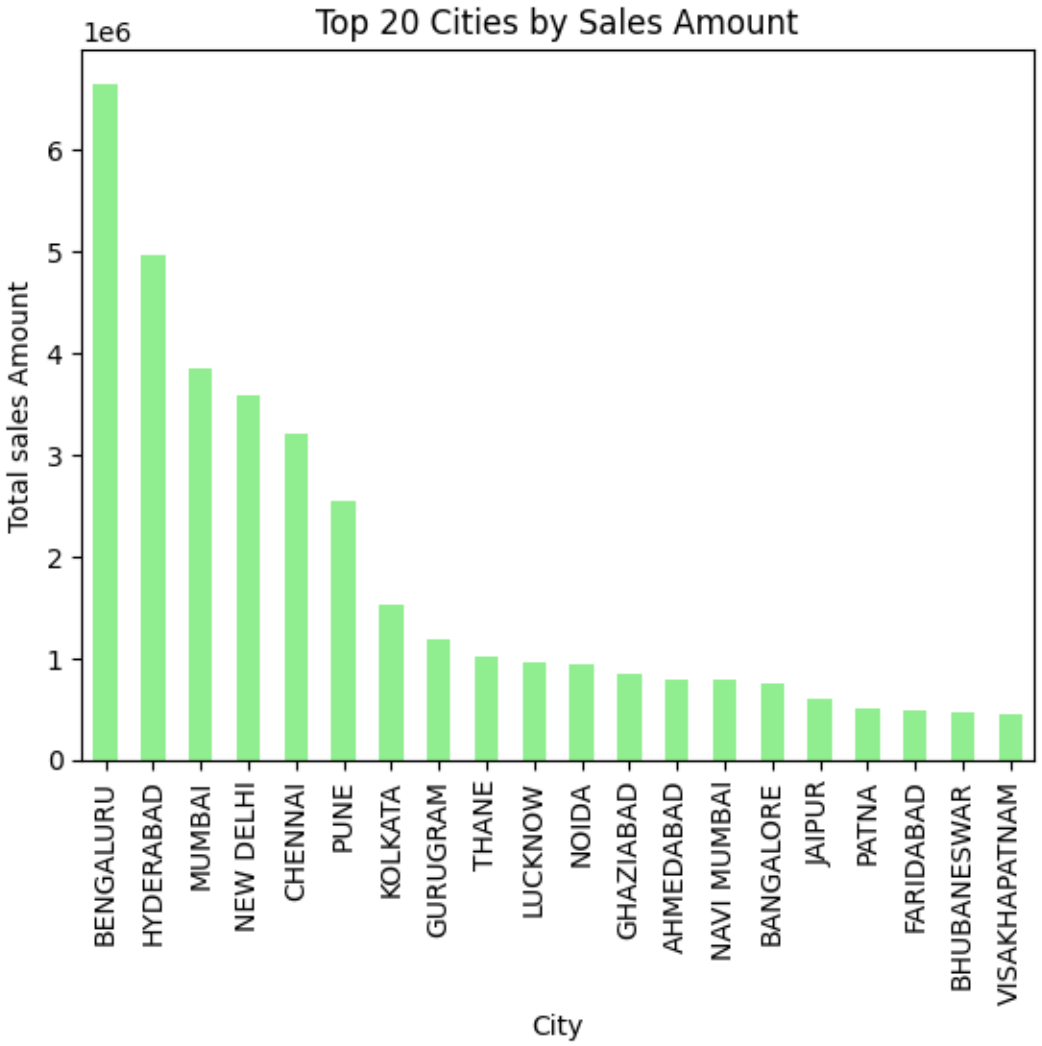
- Maharashtra has the highest total sales with over 12 million.
- Karnataka, Telangana, Uttar Pradesh, and Tamil Nadu also have high sales figures, all above 5 million.
- Lakshadweep, Pondicherry, and Orissa have the lowest sales, with figures under 3,000.
- The Northern and Southern regions, particularly Maharashtra, Karnataka, and Tamil Nadu, dominate the sales distribution.

- Visualize Sales Distribution by City

[ ]:
```
# Plotting sales distribution by city
plt.figure(figsize=(14, 10))
```

```
sales_by_city.sort_values(by='Total Amount', ascending=False).head(20).
 ↪plot(kind='bar', x='City', y='Total Amount', color='lightgreen',␣
 ↪legend=False)
plt.title('Top 20 Cities by Sales Amount')
plt.xlabel('City')
plt.ylabel('Total sales Amount')
plt.show()
```

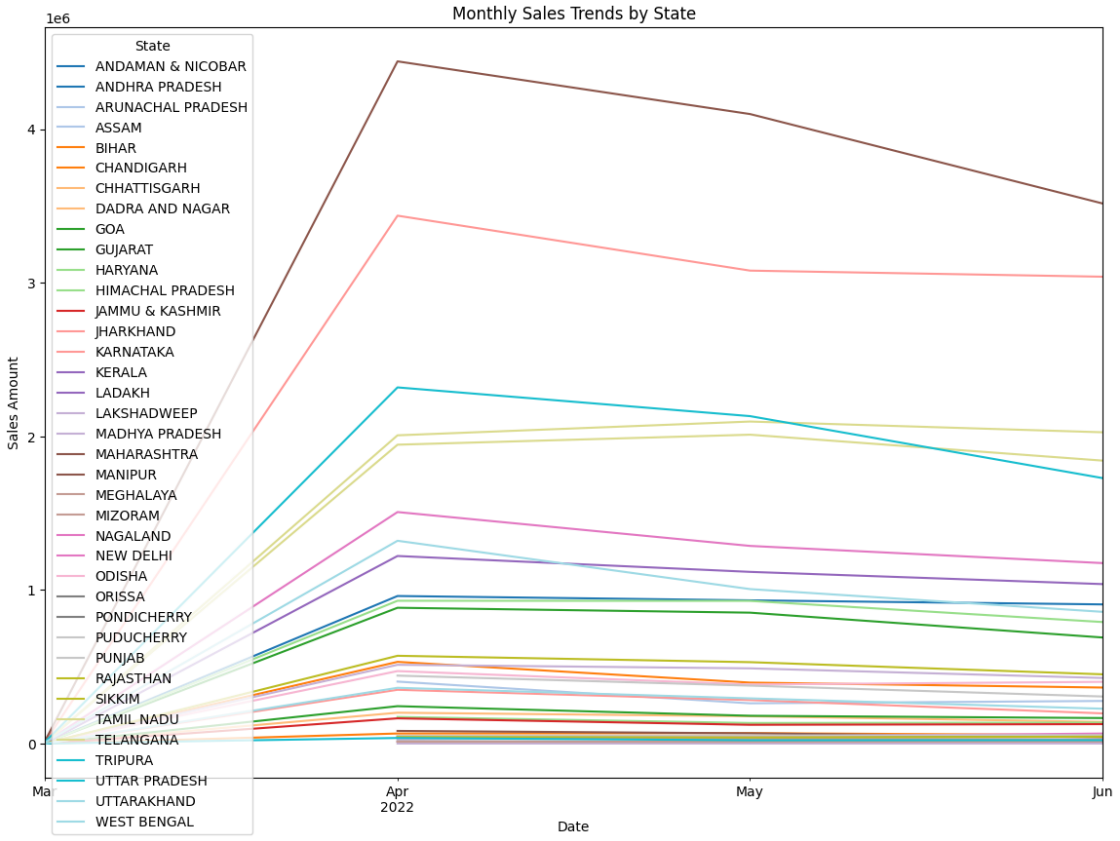
```
<Figure size 1400x1000 with 0 Axes>
```



- Comparing Sales Growth between cities or States

[ ]:
```
monthly_sales_by_state = amazons1_copy.groupby([pd.Grouper(key='Date',␣
 ↪freq='M'), 'State'])['Total Amount'].sum().unstack()
# Plotting the trends
```

```
monthly_sales_by_state.plot(kind='line', figsize=(14, 10), colormap='tab20') #␣
 ↪Changed 'lines' to 'line'
plt.title('Monthly Sales Trends by State')
plt.xlabel('Date')
plt.ylabel('Sales Amount')
plt.show()
```



1.34 Insights and Observations:

- April month is the highest sales growth.

1.35 Conclusion:

- The analysis of the Amazon sales transactions dataset provided valuable insights that can be used to support strategic decision-making, enhance operational efficiency, and drive business growth. Through data cleaning, exploratory data analysis, and data visualization, the report offers a comprehensive understanding of the sales performance and identifies key opportunities for improvement.