

```
In [5]: import pandas as pd
```

```
In [7]: import zipfile
```

```
In [9]: zip_path = r"C:\Users\visha\Downloads\Amazon Sale Report.csv.zip"
```

```
In [11]: extract_path = r"C:\Users\visha\Downloads"
```

```
In [13]: with zipfile.ZipFile(zip_path, 'r') as zip_ref:  
    zip_ref.extractall(extract_path)  
  
    print("File Unzipped Successfully!")
```

```
File Unzipped Successfully!
```

```
In [15]: df = r"C:\Users\visha\Downloads\Amazon Sale Report.csv"  
df
```

```
Out[15]: 'C:\\Users\\\\visha\\\\Downloads\\\\Amazon Sale Report.csv'
```

```
In [17]: csv_path = r"C:\Users\visha\Downloads\Amazon Sale Report.csv"  
df = pd.read_csv(csv_path, low_memory=False)
```

```
In [19]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 128975 entries, 0 to 128974
Data columns (total 24 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   index            128975 non-null   int64  
 1   Order ID         128975 non-null   object  
 2   Date             128975 non-null   object  
 3   Status            128975 non-null   object  
 4   Fulfilment       128975 non-null   object  
 5   Sales Channel    128975 non-null   object  
 6   ship-service-level 128975 non-null   object  
 7   Style             128975 non-null   object  
 8   SKU               128975 non-null   object  
 9   Category          128975 non-null   object  
 10  Size              128975 non-null   object  
 11  ASIN              128975 non-null   object  
 12  Courier Status   122103 non-null   object  
 13  Qty               128975 non-null   int64  
 14  currency          121180 non-null   object  
 15  Amount             121180 non-null   float64 
 16  ship-city          128942 non-null   object  
 17  ship-state         128942 non-null   object  
 18  ship-postal-code   128942 non-null   float64 
 19  ship-country        128942 non-null   object  
 20  promotion-ids      79822 non-null   object  
 21  B2B                128975 non-null   bool    
 22  fulfilled-by       39277 non-null   object  
 23  Unnamed: 22         79925 non-null   object  
dtypes: bool(1), float64(2), int64(2), object(19)
memory usage: 22.8+ MB
```

In [21]: df.head()

Out[21]:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Style	SKU
0	0	8078784-5731545	405-30-22	Cancelled	Merchant	Amazon.in	Standard	SET389	SET389-KR-NP-S
1	1	9198151-1101146	171-30-22	Shipped Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3781	JNE3781-KR-XXXL
2	2	0687676-7273146	404-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3371	JNE3371-KR-XL
3	3	9615377-8133951	403-30-22	Cancelled	Merchant	Amazon.in	Standard	J0341	J0341-DR-L
4	4	1069790-7240320	407-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3671	JNE3671-TU-XXXL

5 rows × 24 columns



```
In [23]: print("Dataset Shape:", df.shape)
print("Columns in Dataset:", df.columns)
```

```
Dataset Shape: (128975, 24)
Columns in Dataset: Index(['index', 'Order ID', 'Date', 'Status', 'Fulfilment', 'Sales Channel',
   'ship-service-level', 'Style', 'SKU', 'Category', 'Size', 'ASIN',
   'Courier Status', 'Qty', 'currency', 'Amount', 'ship-city',
   'ship-state', 'ship-postal-code', 'ship-country', 'promotion-ids',
   'B2B', 'fulfilled-by', 'Unnamed: 22'],
  dtype='object')
```

```
In [25]: print(df.describe())
```

	index	Qty	Amount	ship-postal-code
count	128975.000000	128975.000000	121180.000000	128942.000000
mean	64487.000000	0.904431	648.561465	463966.236509
std	37232.019822	0.313354	281.211687	191476.764941
min	0.000000	0.000000	0.000000	110001.000000
25%	32243.500000	1.000000	449.000000	382421.000000
50%	64487.000000	1.000000	605.000000	500033.000000
75%	96730.500000	1.000000	788.000000	600024.000000
max	128974.000000	15.000000	5584.000000	989898.000000

```
In [27]: print(df.isnull().sum())
```

```
index          0
Order ID       0
Date           0
Status          0
Fulfilment     0
Sales Channel   0
ship-service-level 0
Style           0
SKU             0
Category         0
Size            0
ASIN            0
Courier Status  6872
Qty              0
currency        7795
Amount          7795
ship-city        33
ship-state       33
ship-postal-code 33
ship-country     33
promotion-ids    49153
B2B              0
fulfilled-by     89698
Unnamed: 22      49050
dtype: int64
```

```
In [29]: df = df.drop(columns=['promotion-ids', 'fulfilled-by', 'Unnamed: 22'], errors='ignore')
# Unwanted Columns हटाओ
df
```

Out[29]:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Style
0	0	405-8078784-5731545	04-30-22	Cancelled	Merchant	Amazon.in	Standard	SET389 SE KR
1	1	171-9198151-1101146	04-30-22	Shipped Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3781 JNE KR
2	2	404-0687676-7273146	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3371 JNE
3	3	403-9615377-8133951	04-30-22	Cancelled	Merchant	Amazon.in	Standard	J0341 J
4	4	407-1069790-7240320	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3671 JNE TU
...								
128970	128970	406-6001380-7673107	05-31-22	Shipped	Amazon	Amazon.in	Expedited	JNE3697 JNE
128971	128971	402-9551604-7544318	05-31-22	Shipped	Amazon	Amazon.in	Expedited	SET401 SE KI
128972	128972	407-9547469-3152358	05-31-22	Shipped	Amazon	Amazon.in	Expedited	J0157 J D
128973	128973	402-6184140-0545956	05-31-22	Shipped	Amazon	Amazon.in	Expedited	J0012 J ST
128974	128974	408-7436540-8728312	05-31-22	Shipped	Amazon	Amazon.in	Expedited	J0003 J

128975 rows × 21 columns



In [31]:

```
# Missing Values को Fill करो
df['Courier Status'] = df['Courier Status'].fillna("Unknown") # Text डेटा में "Unknown" को Fill करो
df['currency'] = df['currency'].fillna("INR") # Default currency INR मानकर Fill करो
df['Amount'] = df['Amount'].fillna(df['Amount'].median()) # Median से Fill करो
df['ship-city'] = df['ship-city'].fillna("Unknown")
df['ship-state'] = df['ship-state'].fillna("Unknown")
```

```
df['ship-country'] = df['ship-country'].fillna("Unknown")
```

Out[31]:

		index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Style
0	0	8078784-5731545	405-	04-30-22	Cancelled	Merchant	Amazon.in	Standard	SET389
1	1	9198151-1101146	171-	04-30-22	Shipped Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3781
2	2	0687676-7273146	404-	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3371
3	3	9615377-8133951	403-	04-30-22	Cancelled	Merchant	Amazon.in	Standard	J0341
4	4	1069790-7240320	407-	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3671
...									
128970	128970	6001380-7673107	406-	05-31-22	Shipped	Amazon	Amazon.in	Expedited	JNE3697
128971	128971	9551604-7544318	402-	05-31-22	Shipped	Amazon	Amazon.in	Expedited	SET401
128972	128972	9547469-3152358	407-	05-31-22	Shipped	Amazon	Amazon.in	Expedited	J0157
128973	128973	6184140-0545956	402-	05-31-22	Shipped	Amazon	Amazon.in	Expedited	J0012
128974	128974	7436540-8728312	408-	05-31-22	Shipped	Amazon	Amazon.in	Expedited	J0003

128975 rows × 21 columns



In [33]:

```
df['Date'] = pd.to_datetime(df['Date'], errors='coerce') # Date Format ਦੀ Convert ਕਰਾਵਾਂ
df['Amount'] = pd.to_numeric(df['Amount'], errors='coerce') # Amount ਦੀ Numeric ਕਰਾਵਾਂ
df['Qty'] = pd.to_numeric(df['Qty'], errors='coerce') # Quantity ਦੀ Numeric ਕਰਾਵਾਂ
```

```
C:\Users\visha\AppData\Local\Temp\ipykernel_24436\550112044.py:1: UserWarning: Could
not infer format, so each element will be parsed individually, falling back to `date
util`. To ensure parsing is consistent and as-expected, please specify a format.
  df['Date'] = pd.to_datetime(df['Date'], errors='coerce') # Date Format में Convert
  करें
```

Out[33]:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Style
0	0	405-8078784-5731545	2022-04-30	Cancelled	Merchant	Amazon.in	Standard	SET389 SIKF
1	1	171-9198151-1101146	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3781 JNIKF
2	2	404-0687676-7273146	2022-04-30	Shipped	Amazon	Amazon.in	Expedited	JNE3371 JNII
3	3	403-9615377-8133951	2022-04-30	Cancelled	Merchant	Amazon.in	Standard	J0341
4	4	407-1069790-7240320	2022-04-30	Shipped	Amazon	Amazon.in	Expedited	JNE3671 JNITL
...	...	...	...	...	...	...	...	...
128970	128970	406-6001380-7673107	2022-05-31	Shipped	Amazon	Amazon.in	Expedited	JNE3697 JNII
128971	128971	402-9551604-7544318	2022-05-31	Shipped	Amazon	Amazon.in	Expedited	SET401 SIK
128972	128972	407-9547469-3152358	2022-05-31	Shipped	Amazon	Amazon.in	Expedited	J0157 C
128973	128973	402-6184140-0545956	2022-05-31	Shipped	Amazon	Amazon.in	Expedited	J0012 S
128974	128974	408-7436540-8728312	2022-05-31	Shipped	Amazon	Amazon.in	Expedited	J0003

128975 rows × 21 columns



In [35]: # जैसे "DD-MM-YYYY" या "YYYY-MM-DD"), तो format आँप्शन देकर Date को Convert करो:  
`df['Date'] = pd.to_datetime(df['Date'], format="%d-%m-%Y", errors='coerce')`

Out[35]:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Style
0	0	8078784-5731545	2022-04-30	Cancelled	Merchant	Amazon.in	Standard	SET389 S KF
1	1	9198151-1101146	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3781 JN KF
2	2	0687676-7273146	2022-04-30	Shipped	Amazon	Amazon.in	Expedited	JNE3371 JN
3	3	9615377-8133951	2022-04-30	Cancelled	Merchant	Amazon.in	Standard	J0341
4	4	1069790-7240320	2022-04-30	Shipped	Amazon	Amazon.in	Expedited	JNE3671 JN TL
...	...	...	...	...	...	...	...	...
128970	128970	6001380-7673107	2022-05-31	Shipped	Amazon	Amazon.in	Expedited	JNE3697 JN
128971	128971	9551604-7544318	2022-05-31	Shipped	Amazon	Amazon.in	Expedited	SET401 S
128972	128972	9547469-3152358	2022-05-31	Shipped	Amazon	Amazon.in	Expedited	J0157 C
128973	128973	6184140-0545956	2022-05-31	Shipped	Amazon	Amazon.in	Expedited	J0012 S
128974	128974	7436540-8728312	2022-05-31	Shipped	Amazon	Amazon.in	Expedited	J0003

128975 rows × 21 columns



In [37]: `print(df['Date'].head(10)) # पहले 10 डेट्स देखो`

```
0    2022-04-30
1    2022-04-30
2    2022-04-30
3    2022-04-30
4    2022-04-30
5    2022-04-30
6    2022-04-30
7    2022-04-30
8    2022-04-30
9    2022-04-30
Name: Date, dtype: datetime64[ns]
```

In [39]: `print(df.columns) # यह सभी कॉलम्स को प्रिंट करेगा`

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

In [41]: `df['Total Sales'] = df['Amount'] * df['Qty'] # 'Price' की जगह 'Amount' यूज़ करे
print(df[['Amount', 'Qty', 'Total Sales']].head()) # Verify करते ही 'Total Sales' डिक्टेन्ड हो जाएगा`

	Amount	Qty	Total Sales
0	647.62	0	0.0
1	406.00	1	406.0
2	329.00	1	329.0
3	753.33	0	0.0
4	574.00	1	574.0

In [43]: `print(df['Total Sales'].describe()) # Min, Max, Mean, Median Sales चेक करें`

```
count    128975.000000
mean      590.311619
std       387.903150
min       0.000000
25%     399.000000
50%     568.000000
75%     771.000000
max    44672.000000
Name: Total Sales, dtype: float64
```

In [45]: `top_products = df.groupby('Category')['Total Sales'].sum().sort_values(ascending=False)
print(top_products) # best selling product`

Category	Total Sales
Set	37973154.0
kurta	20720724.0
Western Dress	10710352.0
Top	5245351.0
Ethnic Dress	762949.0

In [47]: `df['Year-Month'] = df['Date'].dt.to_period('M') # Year-Month Format में Convert करते ही
monthly_sales = df.groupby('Year-Month')['Total Sales'].sum()

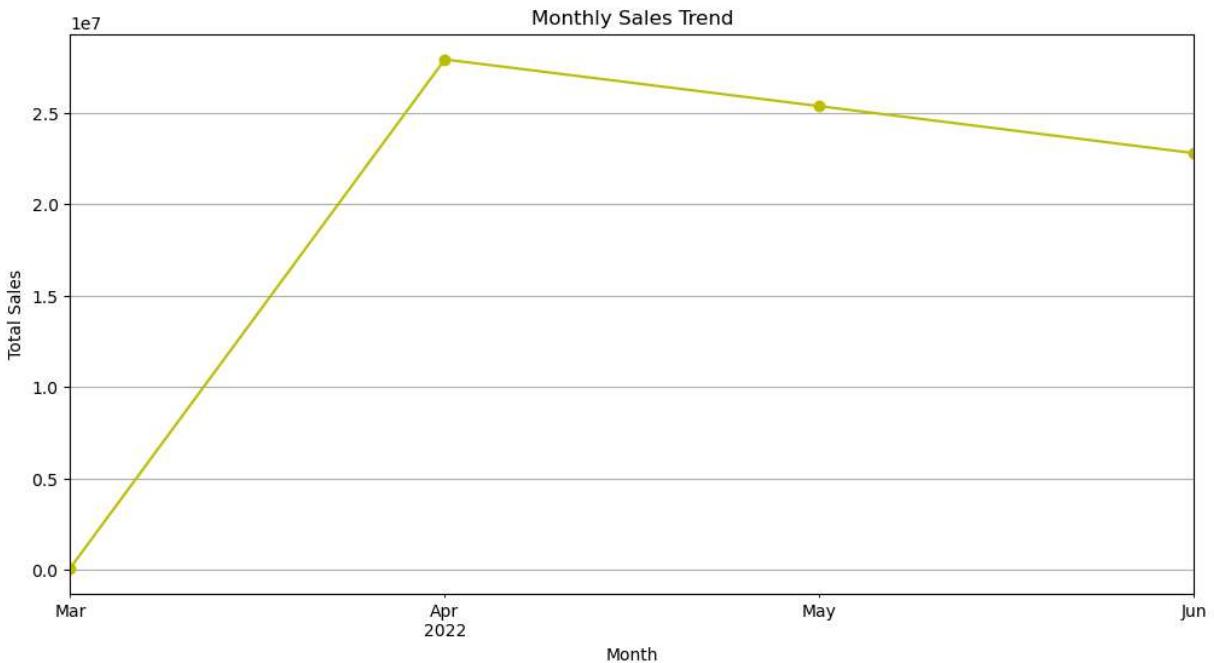
print(monthly_sales.head()) # Sales Trends चेक करें`

```
Year-Month
2022-03      98261.0
2022-04     27904720.0
2022-05    25351488.0
2022-06    22780972.0
Freq: M, Name: Total Sales, dtype: float64
```

In [49]: # Matplotlib: ग्राफ बनाने के लिए Python की सबसे पॉपुलर लाइब्रेरी।  
# Seaborn: Matplotlib को और सुंदर और आसान बनाता है।

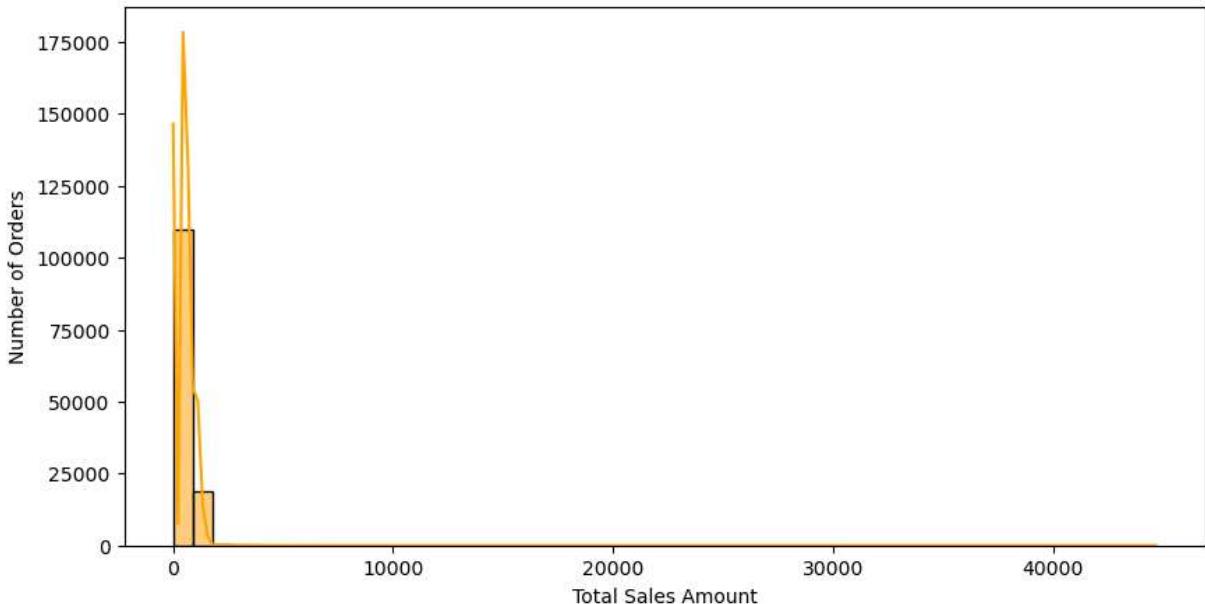
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [50]: plt.figure(figsize=(12,6))
monthly\_sales.plot(kind='line', marker='o', color='y') # show monthly wise, kind-line
# color-b for blue or y
plt.xlabel("Month") # X-Axis Label
plt.ylabel("Total Sales") # Y-Axis Label
plt.title("Monthly Sales Trend") # # ग्राफ का Title
plt.xticks(rotation=45) # rotation=45 → X-Axis पर महीनों के नाम 45° एंगल पर Rotate होते हैं
plt.grid() #Grid Lines जोड़ने से ग्राफ पढ़ना आसान हो जाता है। in a row form
plt.show()



In [51]: plt.figure(figsize=(10,5)) # chart ki L and b
sns.histplot(df['Total Sales'], bins=50, kde=True, color='orange') # sns.histplot--
plt.xlabel("Total Sales Amount") # X-Axis का नाम
plt.ylabel("Number of Orders") # Y-Axis का नाम
plt.title("Distribution of Total Sales") # Graph का Title
plt.show()

Distribution of Total Sales



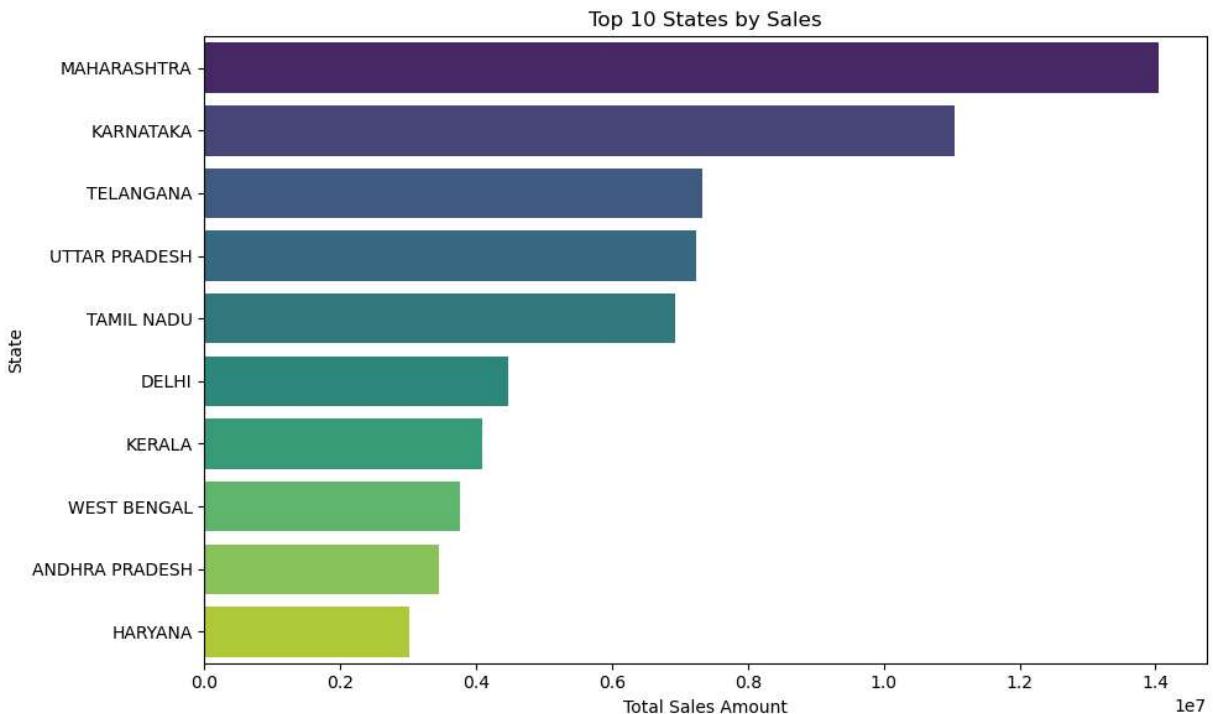
```
In [60]: top_states = df.groupby('ship-state')['Amount'].sum().sort_values(ascending=False).

plt.figure(figsize=(10,6))
sns.barplot(x=top_states.values, y=top_states.index, palette='viridis')
plt.title('Top 10 States by Sales')
plt.xlabel('Total Sales Amount')
plt.ylabel('State')
plt.tight_layout()
plt.show()
```

C:\Users\visha\AppData\Local\Temp\ipykernel\_24436\2215763642.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1  
4.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x=top_states.values, y=top_states.index, palette='viridis')
```



In [ ]: