

```
In [4]: #import Libraries
```

```
In [24]: import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns
```

```
In [25]: df = pd.read_csv('Amazon Sale Report.csv')
```

```
In [26]: df
```

Out[26]:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category
0	0	405- 8078784- 5731545	04- 30- 22	Cancelled	Merchant	Amazon.in	Standard	T-shirt
1	1	171- 9198151- 1101146	04- 30- 22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt
2	2	404- 0687676- 7273146	04- 30- 22	Shipped	Amazon	Amazon.in	Expedited	Shirt
3	3	403- 9615377- 8133951	04- 30- 22	Cancelled	Merchant	Amazon.in	Standard	Blazzer
4	4	407- 1069790- 7240320	04- 30- 22	Shipped	Amazon	Amazon.in	Expedited	Trousers
...
128971	128970	406- 6001380- 7673107	05- 31- 22	Shipped	Amazon	Amazon.in	Expedited	Shirt
128972	128971	402- 9551604- 7544318	05- 31- 22	Shipped	Amazon	Amazon.in	Expedited	T-shirt
128973	128972	407- 9547469- 3152358	05- 31- 22	Shipped	Amazon	Amazon.in	Expedited	Blazzer
128974	128973	402- 6184140- 0545956	05- 31- 22	Shipped	Amazon	Amazon.in	Expedited	T-shirt
128975	128974	408- 7436540- 8728312	05- 31- 22	Shipped	Amazon	Amazon.in	Expedited	T-shirt

128976 rows × 21 columns

In [27]: `## Remove and drop null values`In [28]: `pd.isnull(df)`

Out[28]:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Co S
0	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	
...
128971	False	False	False	False	False	False	False	False	False	
128972	False	False	False	False	False	False	False	False	False	
128973	False	False	False	False	False	False	False	False	False	
128974	False	False	False	False	False	False	False	False	False	
128975	False	False	False	False	False	False	False	False	False	

128976 rows × 21 columns

In [29]: `df.drop('New', axis = 1, inplace = True)`In [30]: `df.drop('PendingS', axis = 1, inplace = True)`In [31]: `df`

Out[31]:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category
0	0	405-8078784-5731545	04-30-22	Cancelled	Merchant	Amazon.in	Standard	T-shirt
1	1	171-9198151-1101146	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt
2	2	404-0687676-7273146	04-30-22	Shipped	Amazon	Amazon.in	Expedited	Shirt
3	3	403-9615377-8133951	04-30-22	Cancelled	Merchant	Amazon.in	Standard	Blazzer
4	4	407-1069790-7240320	04-30-22	Shipped	Amazon	Amazon.in	Expedited	Trousers
...
128971	128970	406-6001380-7673107	05-31-22	Shipped	Amazon	Amazon.in	Expedited	Shirt
128972	128971	402-9551604-7544318	05-31-22	Shipped	Amazon	Amazon.in	Expedited	T-shirt
128973	128972	407-9547469-3152358	05-31-22	Shipped	Amazon	Amazon.in	Expedited	Blazzer
128974	128973	402-6184140-0545956	05-31-22	Shipped	Amazon	Amazon.in	Expedited	T-shirt
128975	128974	408-7436540-8728312	05-31-22	Shipped	Amazon	Amazon.in	Expedited	T-shirt

128976 rows × 19 columns

In [32]: `pd.isnull(df).sum()`

```
Out[32]: index          0
        Order ID      0
        Date          0
        Status        0
        Fulfilment     0
        Sales Channel  0
        ship-service-level 0
        Category       0
        Size           0
        Courier Status  0
        Qty            0
        currency       7800
        Amount         7800
        ship-city      35
        ship-state     35
        ship-postal-code 35
        ship-country   35
        B2B            0
        fulfilled-by   89713
        dtype: int64
```

```
In [33]: df.shape
```

```
Out[33]: (128976, 19)
```

```
In [15]: ## drop null values
```

```
In [34]: df.dropna(inplace = True)
```

```
In [35]: df.shape
```

```
Out[35]: (37514, 19)
```

```
In [36]: df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Index: 37514 entries, 0 to 128892
Data columns (total 19 columns):
#   Column                Non-Null Count  Dtype
---  -
0   index                 37514 non-null  int64
1   Order ID              37514 non-null  object
2   Date                  37514 non-null  object
3   Status                37514 non-null  object
4   Fulfilment            37514 non-null  object
5   Sales Channel         37514 non-null  object
6   ship-service-level    37514 non-null  object
7   Category              37514 non-null  object
8   Size                  37514 non-null  object
9   Courier Status        37514 non-null  object
10  Qty                   37514 non-null  int64
11  currency              37514 non-null  object
12  Amount                37514 non-null  float64
13  ship-city             37514 non-null  object
14  ship-state            37514 non-null  object
15  ship-postal-code      37514 non-null  float64
16  ship-country          37514 non-null  object
17  B2B                   37514 non-null  bool
18  fulfilled-by          37514 non-null  object
dtypes: bool(1), float64(2), int64(2), object(14)
memory usage: 5.5+ MB

```

In [152...

df.head(20)

Out[152...

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size
0	0	405-8078784-5731545	2022-04-30	Cancelled	Merchant	Amazon.in	Standard	T-shirt	S
1	1	171-9198151-1101146	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	3XL
3	3	403-9615377-8133951	2022-04-30	Cancelled	Merchant	Amazon.in	Standard	Blazzer	L
7	7	406-7807733-3785945	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	S
12	12	405-5513694-8146768	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	XS
14	14	408-1298370-1920302	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	T-shirt	L
15	15	403-4965581-9520319	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	6XL
18	18	402-4030358-5835511	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	XXL
23	23	404-6019946-2909948	2022-04-30	Cancelled	Merchant	Amazon.in	Standard	T-shirt	M
25	25	405-8191138-5176316	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	T-shirt	XS
26	26	403-9230474-9657916	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	3XL
32	32	404-9632124-1107550	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	T-shirt	M

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size
33	33	402-1465437-0579556	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	M
35	35	402-2764952-1492318	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	XL
42	42	406-7398201-3869914	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	M
49	49	171-9208368-0157156	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	M
60	60	171-2592464-6846743	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	T-shirt	XL
70	70	405-9966506-3155561	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	XL
72	72	407-2189901-7515567	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	T-shirt	S
74	74	406-1326801-8886709	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	XL

20 rows × 21 columns

Changing Data Type

```
In [39]: df['ship-postal-code'] = df['ship-postal-code'].astype('int')
```

```
In [40]: df['ship-postal-code'].dtype
```

```
Out[40]: dtype('int32')
```

```
In [154... df['Date'] = pd.to_datetime(df['Date'], format = '%y-%m-%d')
```


In [155... `df['Date'].dtype`

Out[155... `dtype('<M8[ns]')`

In [156... `df.describe()`

Out[156...

	index	Date	Qty	Amount	ship-postal-code	
count	37514.000000	37514	37514.000000	37514.000000	37514.000000	37514.000000
mean	60953.809858	2022-05-11 07:56:47.303939840	0.867383	646.553960	463291.552754	463291.552754
min	0.000000	2022-03-31 00:00:00	0.000000	0.000000	110001.000000	110001.000000
25%	27235.250000	2022-04-20 00:00:00	1.000000	458.000000	370465.000000	370465.000000
50%	63470.500000	2022-05-09 00:00:00	1.000000	629.000000	500019.000000	500019.000000
75%	91790.750000	2022-06-01 00:00:00	1.000000	771.000000	600042.000000	600042.000000
max	128891.000000	2022-06-29 00:00:00	5.000000	5495.000000	989898.000000	989898.000000
std	36844.853039	NaN	0.354160	279.952414	194550.425637	194550.425637




In [44]: `# describe our data with objects datatypes`

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

Out[44]:

	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size
count	37514	37514	37514	37514	37514	37514	37514	37514
unique	34664	91	11	1	1	1	8	11
top	171-5057375-2831560	04-25-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	T-shirt	M
freq	12	697	28741	37514	37514	37514	14062	6806



In []:

In []:

In [157... `## describe for specific columns`

```
df[['Qty', 'Amount']].describe()
```

Out[157...

	Qty	Amount
count	37514.000000	37514.000000
mean	0.867383	646.553960
std	0.354160	279.952414
min	0.000000	0.000000
25%	1.000000	458.000000
50%	1.000000	629.000000
75%	1.000000	771.000000
max	5.000000	5495.000000

In [158...

```
df.head()
```

Out[158...

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size
0	0	405-8078784-5731545	2022-04-30	Cancelled	Merchant	Amazon.in	Standard	T-shirt	S
1	1	171-9198151-1101146	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	3XL
3	3	403-9615377-8133951	2022-04-30	Cancelled	Merchant	Amazon.in	Standard	Blazzer	L
7	7	406-7807733-3785945	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	S
12	12	405-5513694-8146768	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	XS

5 rows × 21 columns



In [159...

```
sns.color_palette()
```

Out[159...



In [160...

```
sns.color_palette('pastel')
```

Out[160...



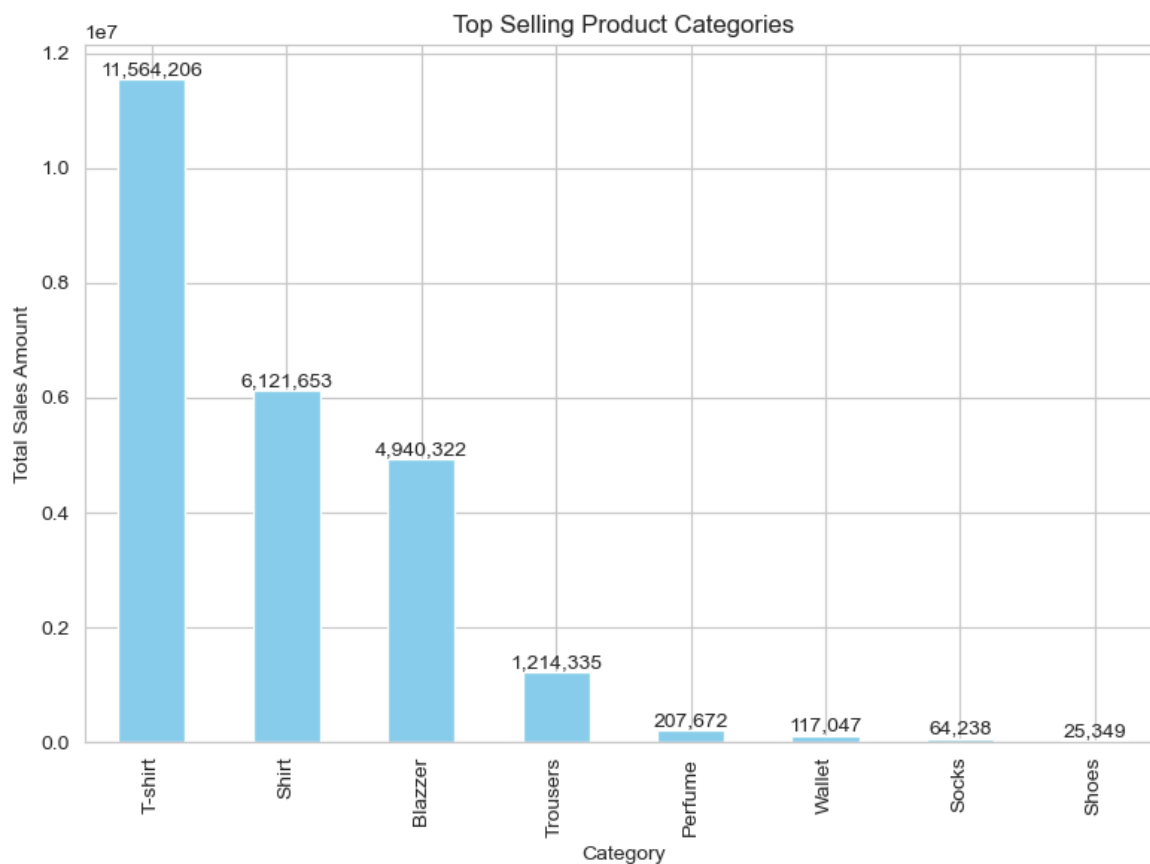
VISUALIZATION USING SEABORN & MATPLOTLIB

In [108...

```
import matplotlib.pyplot as plt

top_categories = df.groupby('Category')['Amount'].sum().sort_values(ascending=False)
ax = top_categories.plot(kind='bar', color='skyblue', figsize=(8, 6))
plt.title('Top Selling Product Categories')
plt.xlabel('Category')
plt.ylabel('Total Sales Amount')

for bar in ax.patches:
    ax.text(bar.get_x() + bar.get_width() / 2, bar.get_height(), f'{int(bar.get_height() * 1e7)}')
plt.tight_layout()
plt.show()
```



insights:

The sales distribution across product categories reveals that T-shirts (₹11.56M) and Shirts (₹6.12M) significantly outperform other categories, contributing the highest revenue. Jeans and Trousers show moderate performance, while categories such as Perfumes, Wallets, and Shoes show comparatively lower sales.

In []:

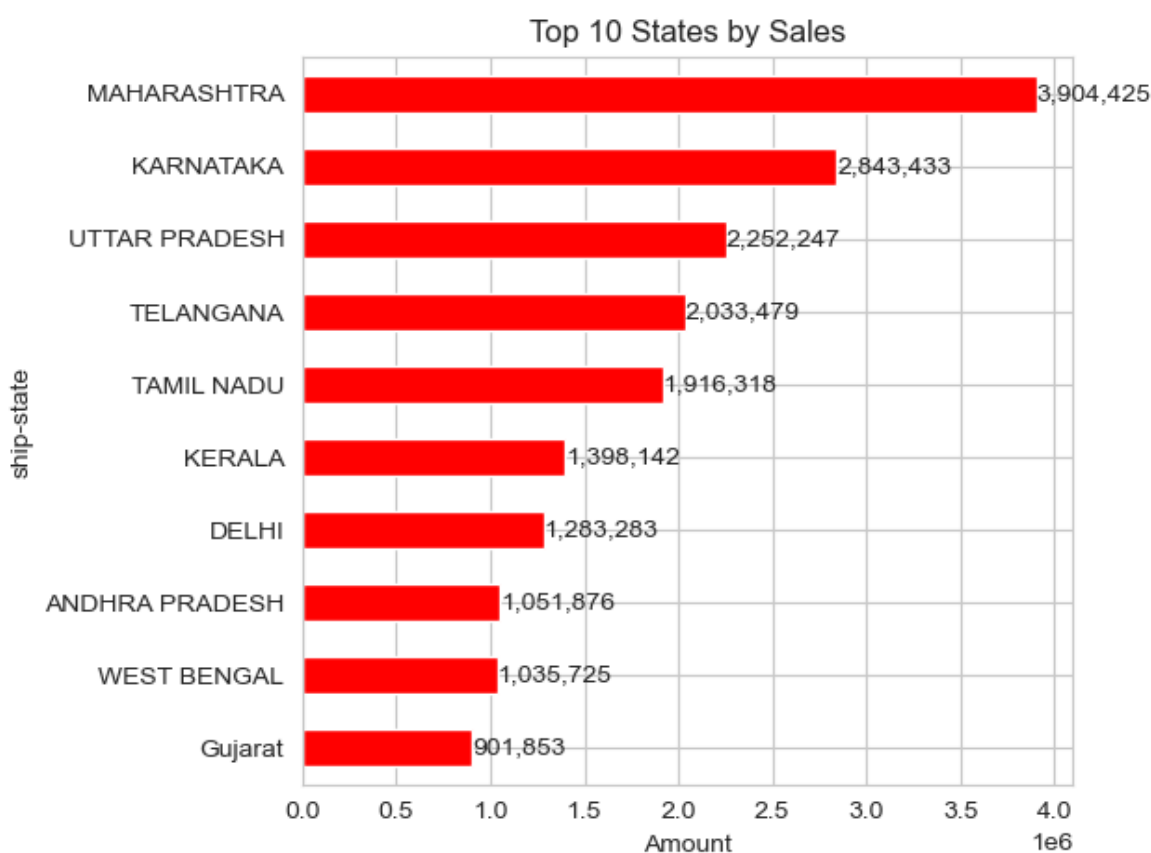
2.Top 10 States By Sales

```
In [130... top10_sales_by_states = df.groupby('ship-state')['Amount'].sum().sort_values(asc
df_top10 = df[df['ship-state'].isin(top10_sales_by_states.index)]
ax = df_top10.groupby('ship-state')['Amount'].sum().sort_values().plot(kind='bar

plt.title('Top 10 States by Sales')
plt.xlabel('Amount')

for bar in ax.patches:
    ax.text(bar.get_width() + 1000, bar.get_y() + bar.get_height() / 2,
            f'{bar.get_width():.0f}', va='center')

plt.tight_layout()
plt.show()
```



insights

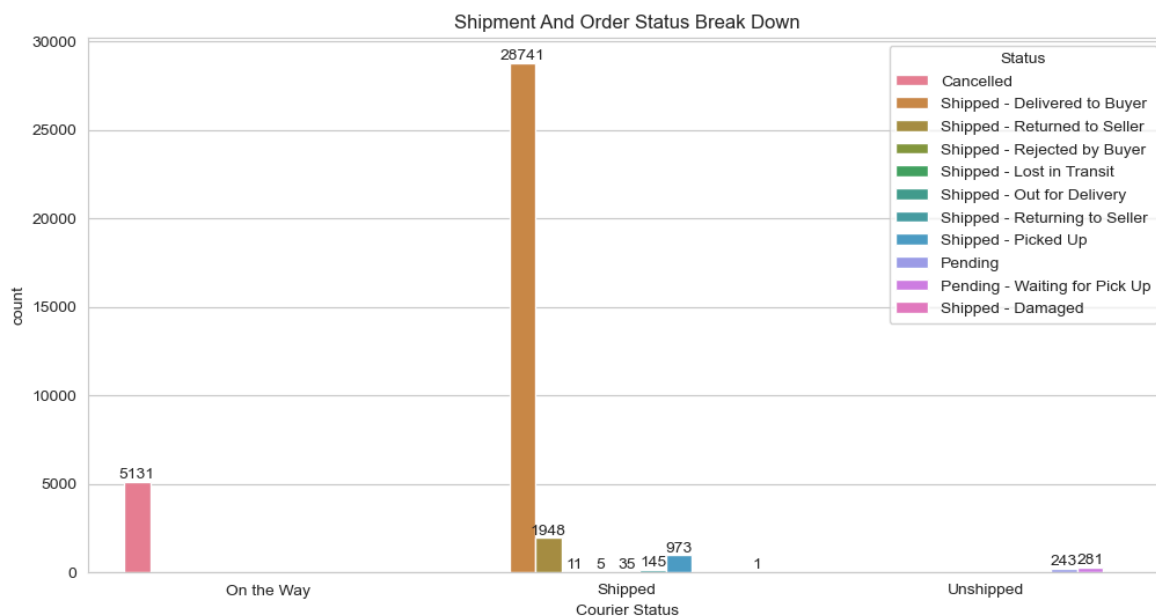
The sales data shows Maharashtra at the top with total sales of 39,442,500, followed by Karnataka with 28,434,330, and Uttar Pradesh at 22,522,470. States like Telangana (₹20,33,479) and Tamil Nadu (₹19,16,318) also contribute notably respectively. On the lower end, West Bengal and Gujarat still made it into the top 10. The wide range in sales values highlights both strong market regions and opportunities for expansion.

In []:

3.Order Status BreakDown

Cancelled vs Delivered Orders

```
In [110... ## courier status
plt.figure(figsize = (12,6))
ax = sns.countplot(data = df, x = 'Courier Status', hue = 'Status')
plt.title('Shipment And Order Status Break Down')
for bars in ax.containers:
    ax.bar_label(bars)
plt.show()
```



insights:

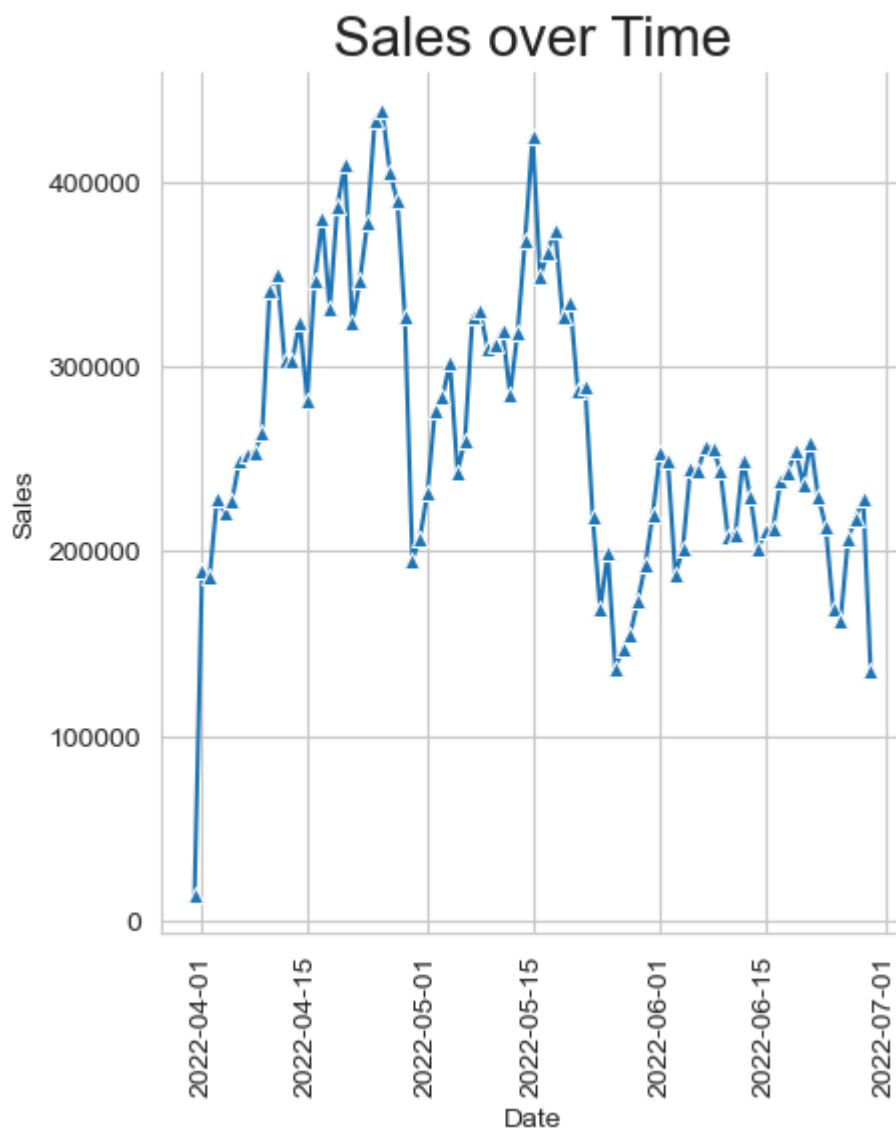
The majority of orders, 2,43,281, fall under the Unshipped category, indicating a major delay or pending fulfillment. Among shipped items, 28,741 were successfully delivered to the buyer, showing strong logistics performance in that area. However, there are 5,131 cancelled orders, which may suggest issues in inventory, pricing, or customer satisfaction. A total of 1,948 shipments were returned to the seller, and 973 are in the process of returning, pointing to possible quality or mismatch issues. The low numbers in damaged, rejected, and lost shipments are positive, but the unshipped volume needs urgent attention.

```
In [ ]:
```

4.Sales Over Time

```
In [150... plt.figure(figsize = (20,6))
sns.set_style('whitegrid')
sns.relplot(data = sales_df, x = 'Date', y = 'Sales', kind = 'line', marker = '^')
plt.title('Sales over Time', fontsize = 20)
plt.xticks(rotation = 'vertical')
plt.show()
```

<Figure size 2000x600 with 0 Axes>



insights:

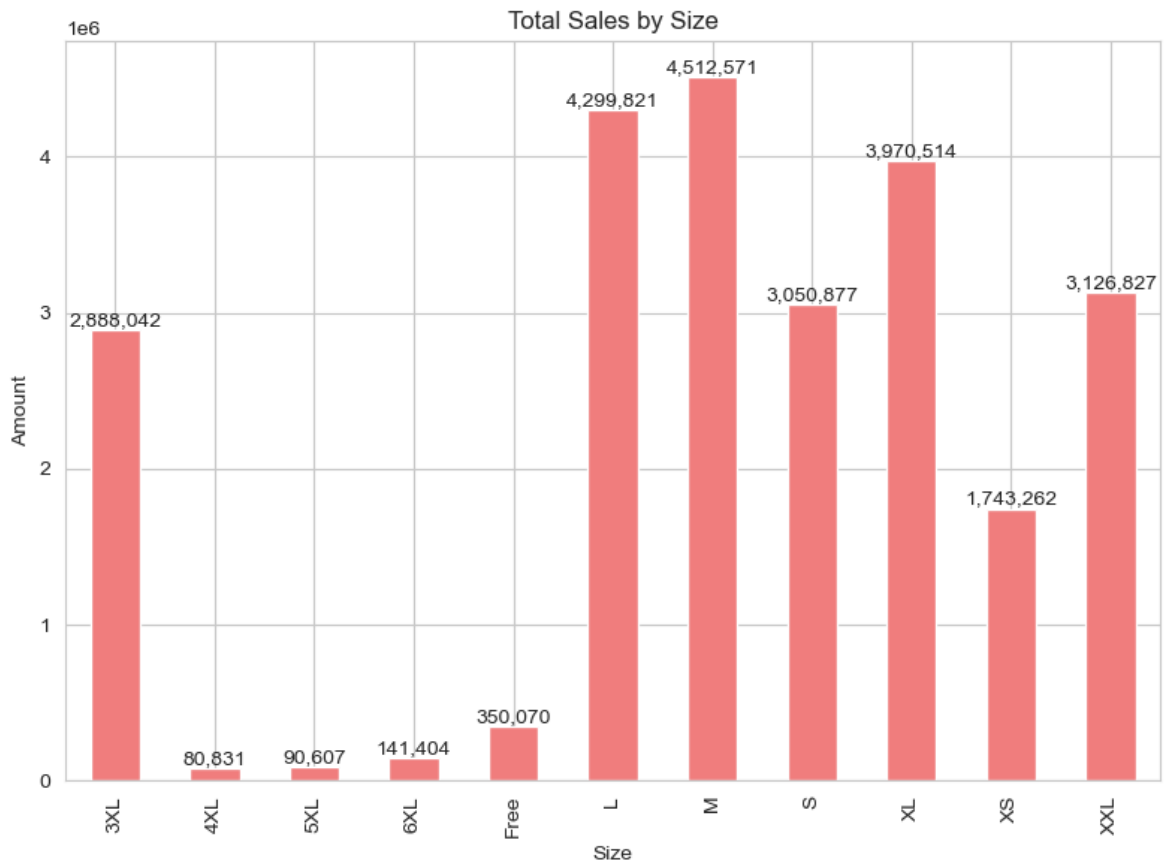
The line chart showing Sales Over Time highlights distinct patterns in Amazon's sales performance during the observed period. Initially, there's a sharp growth in sales from early April 2022, reaching multiple peaks, particularly around late April to mid-May, where sales consistently cross the 400,000 mark. However, post mid-May, there's a noticeable downward trend, with sales declining steadily and stabilizing at a lower average level (~200,000–300,000). This trend may suggest seasonal demand fluctuations, completed promotional campaigns, or supply chain variations. Monitoring such patterns can help in better forecasting and strategic planning for future sales cycles.

In []:

5.Quantity Sold by Size

In [113... `import matplotlib.pyplot as plt`

```
ax = df.groupby('Size')['Amount'].sum().plot(kind='bar', color='lightcoral', fig
plt.title('Total Sales by Size')
plt.ylabel('Amount')
plt.xlabel('Size')
for bar in ax.patches:
    ax.text(bar.get_x() + bar.get_width() / 2, bar.get_height(), f'{bar.get_height()}'
plt.tight_layout()
plt.show()
```



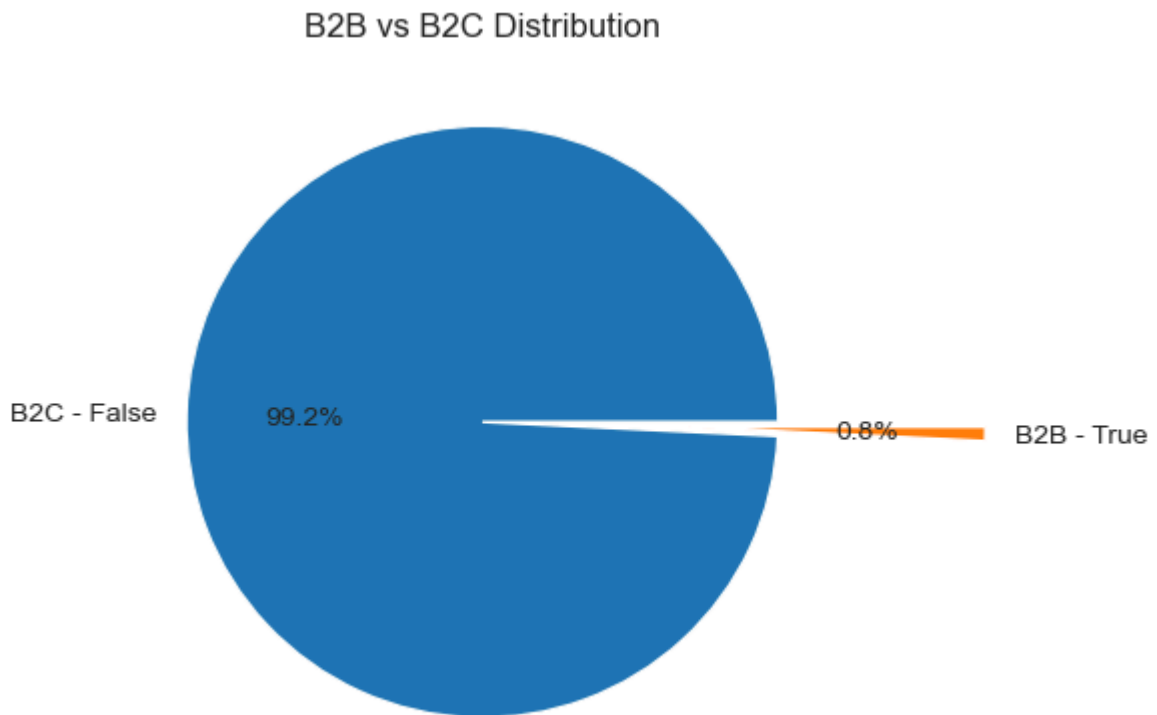
insights:

The data shows that medium (M) and large (L) sizes are the top-selling, with 4.51 million and 4.29 million units sold respectively. These are followed by XL (3.97M), S (3.05M), and XXL (3.12M), indicating strong demand in standard adult sizes. In contrast, plus sizes such as 3XL (2.88M), 4XL–6XL show significantly lower sales, with 4XL and 5XL contributing under 100K units. XS and Free size also have moderate demand.

In []:

6.B2B VS B2C Order Distribution

```
In [149... check_B2B = df['B2B'].value_counts()
labels = ['B2B - True' if val else 'B2C - False' for val in check_B2B.index]
plt.pie(check_B2B, labels=labels, autopct='%0.1f%%', explode=[0, 0.7])
plt.title('B2B vs B2C Distribution')
plt.show()
```



insights:

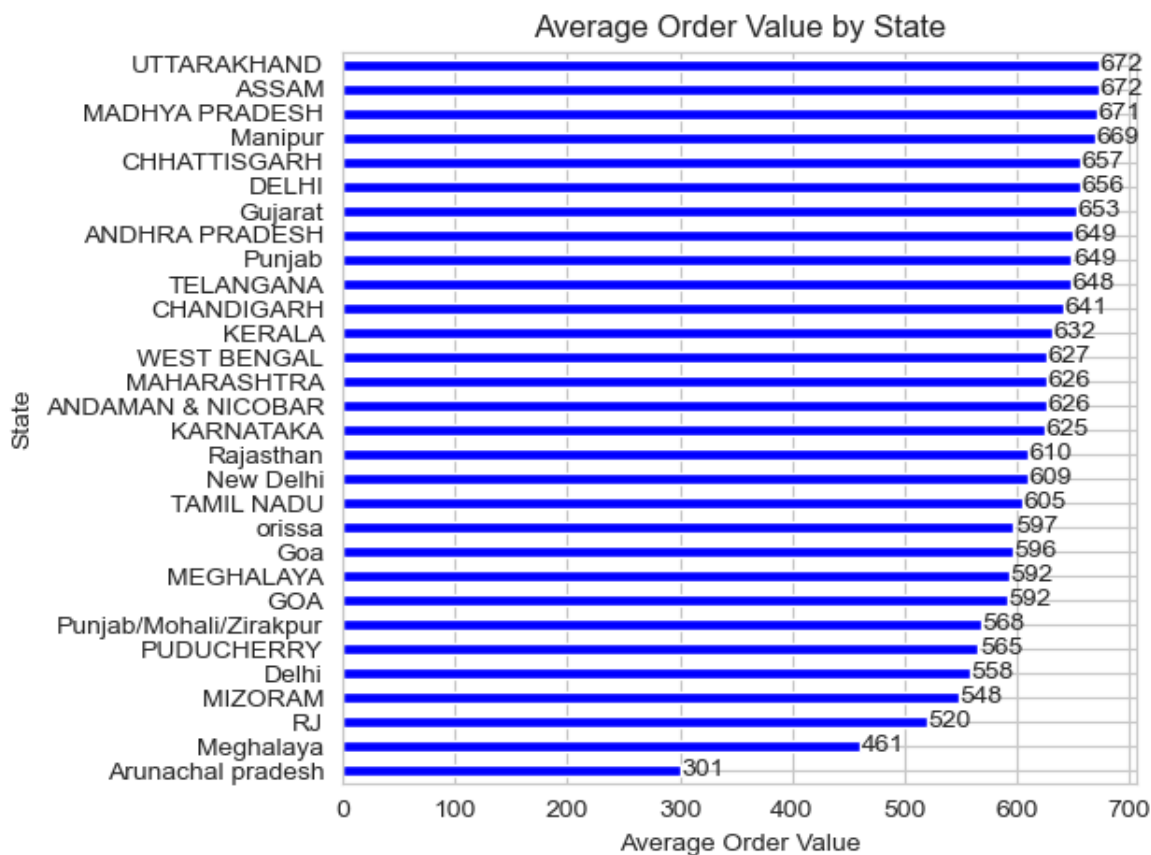
The pie chart shows a clear dominance of B2C (Business to Consumer) orders in the dataset, accounting for approximately 99.2% of the total orders. In contrast, B2B (Business to Business) orders make up only about 0.8%, highlighting that the majority of sales transactions are directed towards individual customers rather than businesses. This indicates a consumer-focused market strategy, with very limited business-oriented sales in the current dataset.

In []:

7.Average Order Value by State

```
In [116... import matplotlib.pyplot as plt
plt.figure(figsize=(18, 8))
ax = aov_by_state.head(30).plot( kind='barh', x='State', y='Average Order Value'
plt.title('Average Order Value by State')
plt.xlabel('Average Order Value')
plt.tight_layout()
for i in ax.patches:
    ax.text(i.get_width() + 1, i.get_y() + i.get_height() / 2, f'{i.get_width():
    plt.tight_layout()
plt.show()
```

<Figure size 1800x800 with 0 Axes>



insights:

The chart showing the average order value by state reveals significant regional variation in customer spending. Uttarakhand and Assam top the list with the highest average order value of ₹672, followed closely by Madhya Pradesh, Manipur, and Chhattisgarh, all exceeding ₹650. Major urban centers like Delhi and Gujarat also reflect strong online spending, with average values of ₹656 and ₹653, respectively. In contrast, northeastern and smaller states such as Mizoram, Meghalaya, and Arunachal Pradesh exhibit much lower averages, with Arunachal Pradesh recording the lowest at ₹301. This disparity indicates differing levels of purchasing power, digital access, and e-commerce penetration across various Indian states.

In []:

8.Sales Channel

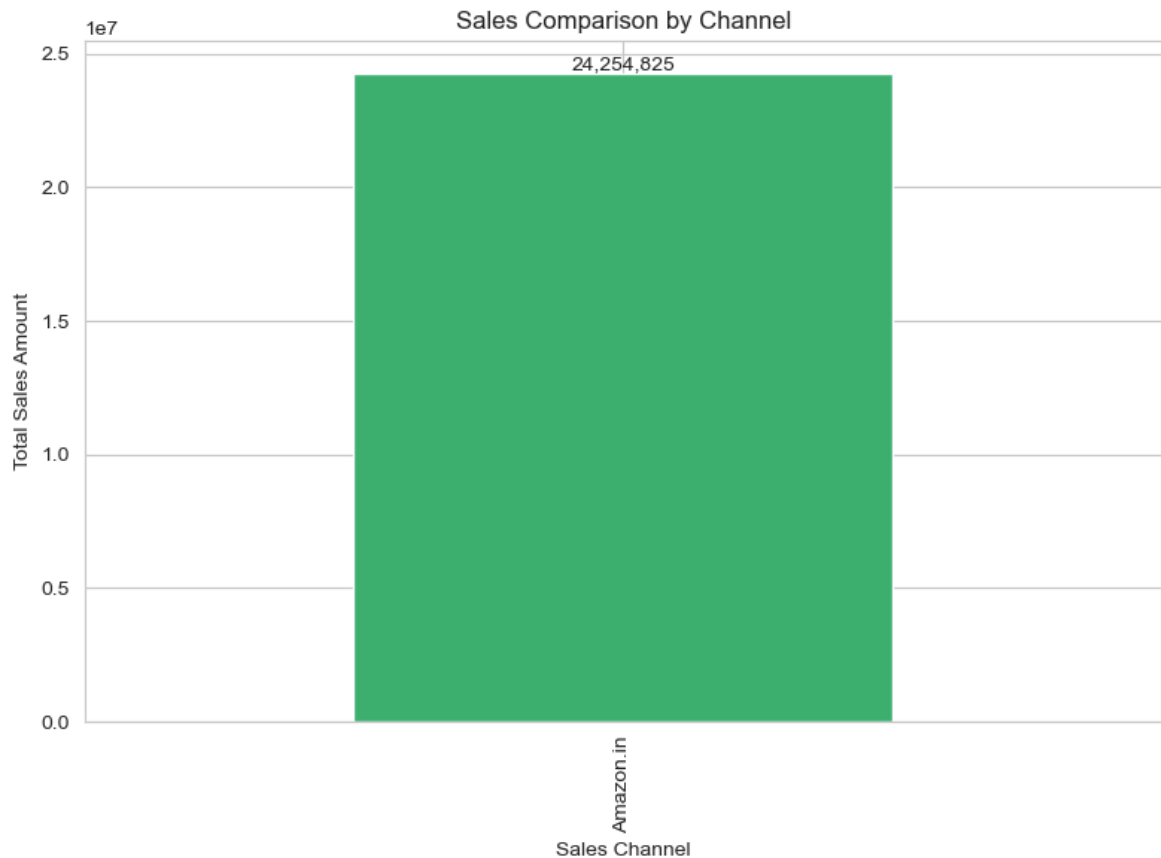
In [118...

```
import matplotlib.pyplot as plt

sales_by_channel = df.groupby('Sales Channel')['Amount'].sum().sort_values(ascen
ax = sales_by_channel.plot(kind='bar', color='mediumseagreen', figsize=(8, 6))
plt.title('Sales Comparison by Channel')
plt.xlabel('Sales Channel')
plt.ylabel('Total Sales Amount')

for bar in ax.patches:
    ax.text(bar.get_x() + bar.get_width() / 2, bar.get_height(), f'{int(bar.get_
```

```
plt.tight_layout()
plt.show()
```



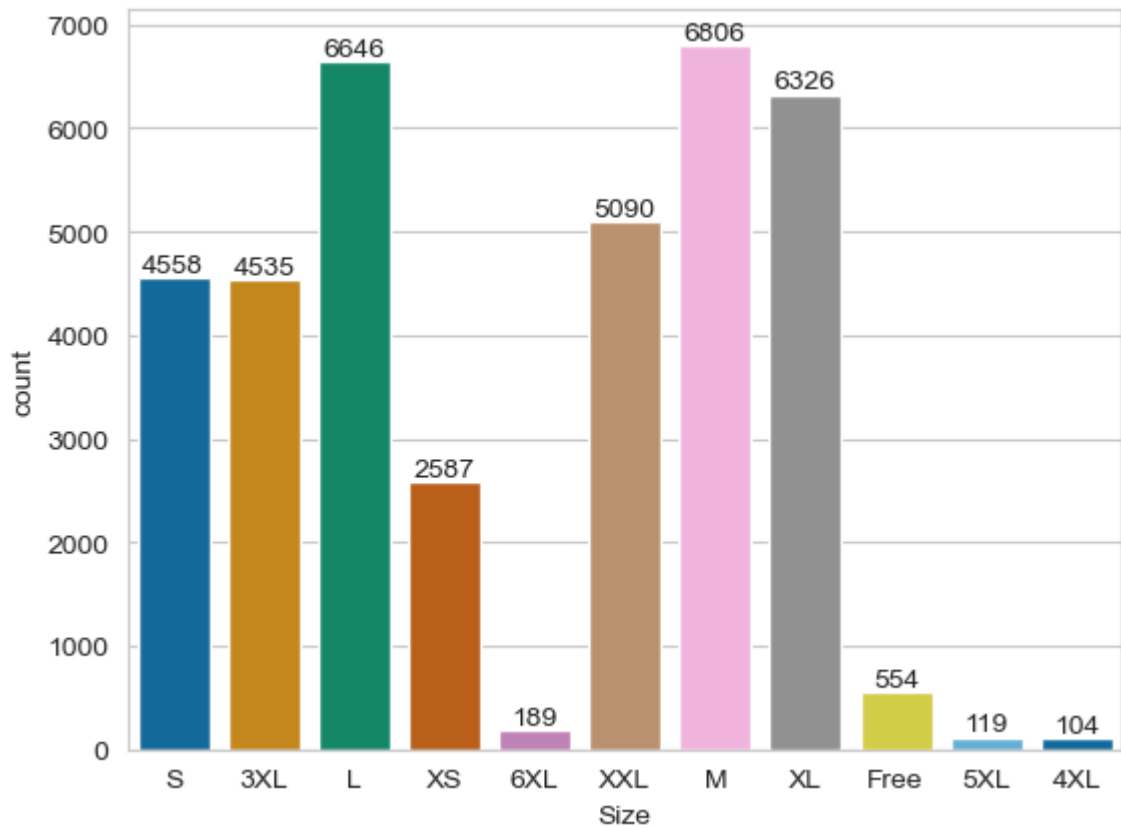
insights:

Based on the sales channel comparison with a total of 24,254,825 records, the insights indicate a clear dominance of one channel over the other. A major portion of the sales volume is attributed to Amazon (main platform), reflecting strong customer preference and higher order frequency through the direct Amazon site

In []:

9.Count of Product Size

```
In [119... ax = sns.countplot(x = 'Size', data=df, palette = 'colorblind', hue = 'Size')
## checking for data labels
for bars in ax.containers:
    ax.bar_label(bars)
```



insights:

Based on the size-wise order count distribution, we observe that the most frequently ordered sizes are M (6806 orders), L (6646 orders), and XL (6326 orders), indicating a high demand for medium to large-sized products. XXL (5090) and S (4558) also show substantial counts, suggesting consistent preferences across various larger and smaller fits. On the lower end, sizes like 4XL (104 orders), 5XL (119), and 6XL (189) have very minimal demand, which could be due to limited availability or niche customer base. The 'Free' size (554 orders), typically used for one-size-fits-all items, reflects moderate interest. These insights can help inventory teams prioritize stocking popular sizes while considering strategy for low-demand ones.

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []: