

Amazon Sales Report

#import datasets

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
```

#loading the datasets

```
data = pd.read_csv("/content/drive/MyDrive/Copy of Amazon Sale Report.csv")
data.head()
```

Output:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	...	currency	Amount	ship-city	ship-state	ship-postal-code	ship-country	B2B	fulfilled-by	New	PendingS
0	0	8078784-5731545	04-30-22	Cancelled	Merchant	Amazon.in	Standard	T-shirt	S	On the Way	...	INR	647.62	MUMBAI	MAHARASHTRA	400081.0	IN	False	Easy Ship	NaN	NaN
1	1	9198151-1101146	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	3XL	Shipped	...	INR	406.00	BENGALURU	KARNATAKA	560085.0	IN	False	Easy Ship	NaN	NaN
2	2	0687676-7273146	04-30-22	Shipped	Amazon	Amazon.in	Expedited	Shirt	XL	Shipped	...	INR	329.00	NAVI MUMBAI	MAHARASHTRA	410210.0	IN	True	NaN	NaN	NaN
3	3	9615377-8133951	04-30-22	Cancelled	Merchant	Amazon.in	Standard	Blazzer	L	On the Way	...	INR	753.33	PUDUCHERRY	PUDUCHERRY	605008.0	IN	False	Easy Ship	NaN	NaN
4	4	1069790-7240320	04-30-22	Shipped	Amazon	Amazon.in	Expedited	Trousers	3XL	Shipped	...	INR	574.00	CHENNAI	TAMIL NADU	600073.0	IN	False	NaN	NaN	NaN

5 rows x 21 columns

#Sales Overview

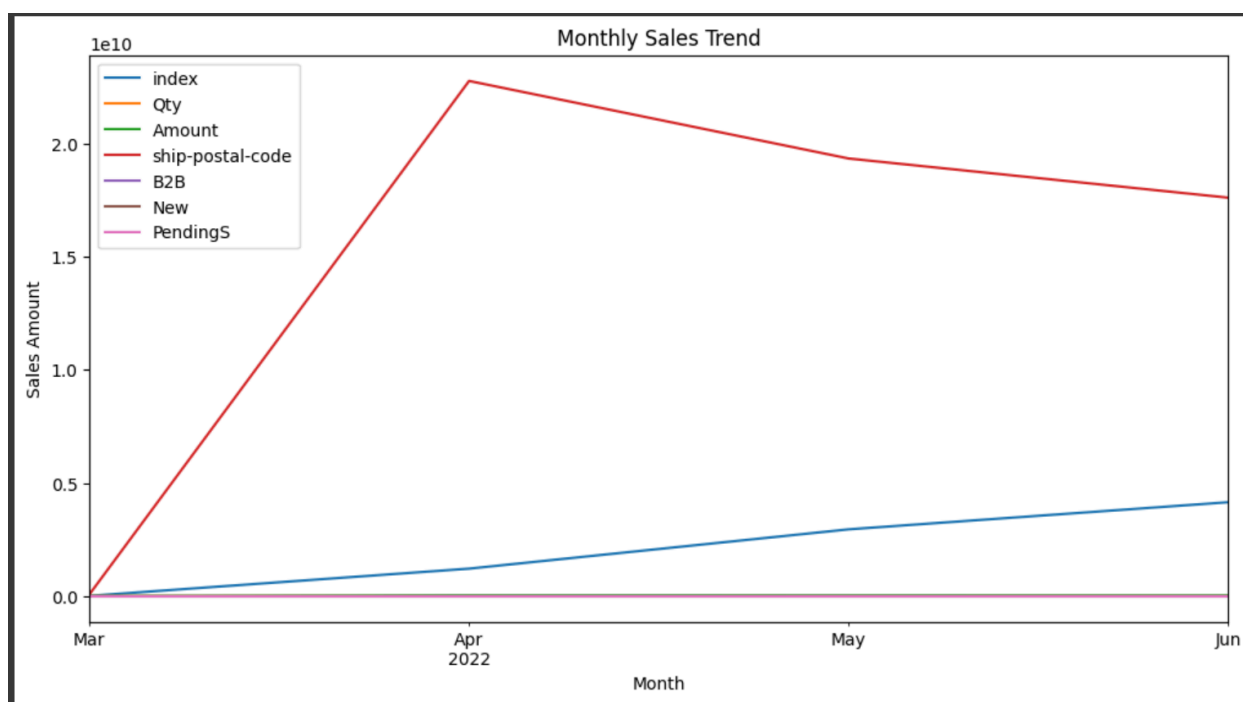
```
data['Date']=pd.to_datetime(data['Date'])
total_sales = data['Amount'].sum()
total_orders=data['Order ID'].nunique()
average_order_value=total_sales/total_orders
print(f'Total Sales: ${total_sales:.2f}')
print(f'Total Orders: {total_orders}')
print(f'Average Order Value: ${average_order_value:.2f}')
```

Output:

```
Total Sales: $78590170.25
Total Orders: 120229
Average Order Value: $653.67
```

```
sales_trend=data.groupby(data['Date'].dt.to_period('M')).sum(['Amount'])
sales_trend.plot(title='Monthly Sales Trend', xlabel='Month', ylabel='Sales Amount',
figsize=(12, 6))
plt.xticks(rotation=45)
plt.show()
```

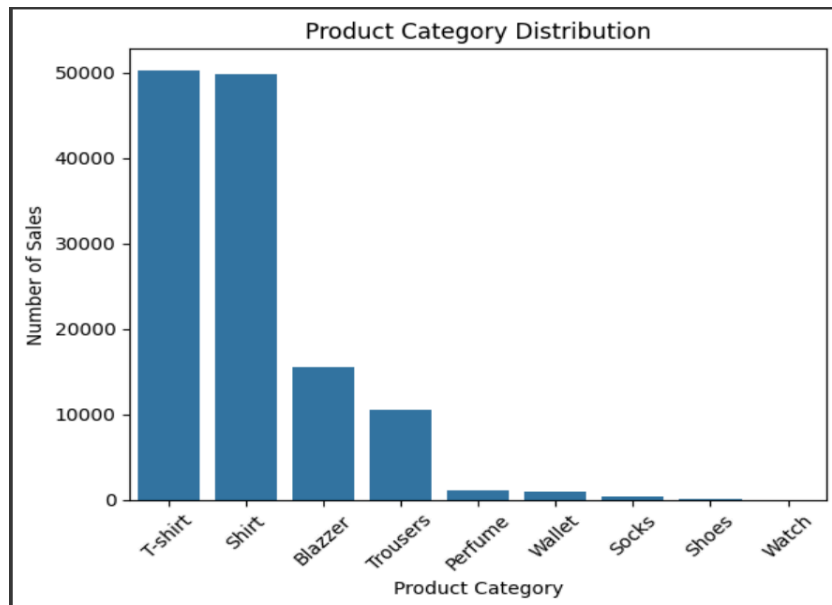
Output:



#product analysis

```
category_distribution = data['Category'].value_counts()
sns.barplot(x=category_distribution.index, y=category_distribution.values)
plt.title('Product Category Distribution')
plt.xlabel('Product Category')
plt.ylabel('Number of Sales')
plt.xticks(rotation=45)
plt.show()
```

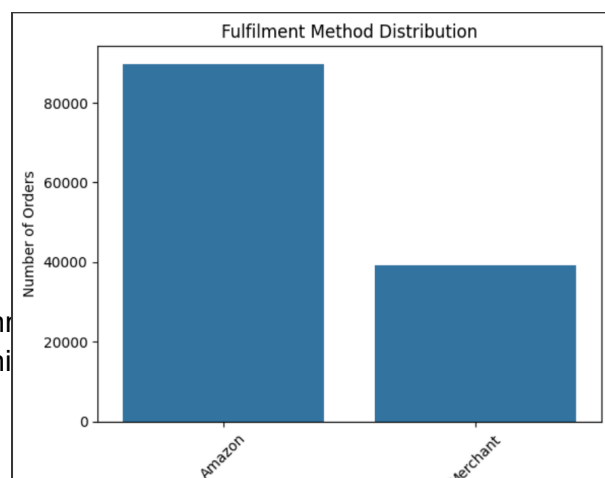
Output:



#fulfillment analysis

```
fulfilment_distribution=data['Fulfilment'].value_counts()
sns.barplot(x=fulfilment_distribution.index, y=fulfilment_distribution.values)
plt.title('Fulfilment Method Distribution')
plt.xlabel('Fulfilment Method')
plt.ylabel('Number of Orders')
plt.xticks(rotation=45)
plt.show()
```

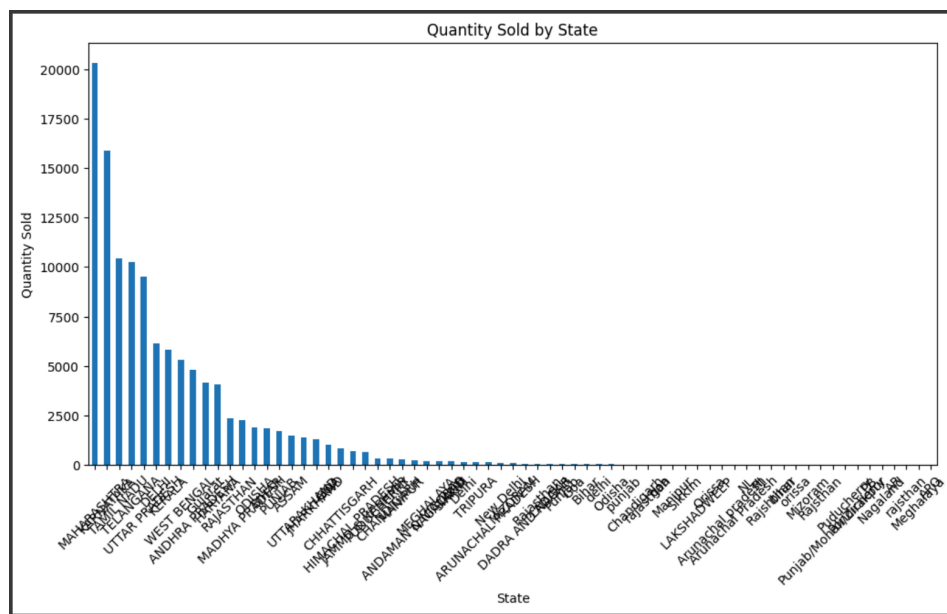
Output:



#geographical analysis

```
if 'ship-state' in data.columns:
    geographical_sales =
data.groupby('ship-state')['Qty'].sum().sort_values(ascending=False)
    geographical_sales.plot(kind='bar', title='Quantity Sold by State',
xlabel='State', ylabel='Quantity Sold', figsize=(12, 6))
    plt.xticks(rotation=45)
    plt.show()
else:
    print("No 'ship-state' column found in the dataset for geographical analysis.")
```

Output:



#Business Insights

```
print("Insights:")
print("- Focus on top-selling product categories for marketing.")
print("- Optimize fulfillment methods based on performance.")
print("- Analyze sales trends to identify peak periods for promotions.")
```

Output:

Insights:

- Focus on top-selling product categories for marketing.
- Optimize fulfillment methods based on performance.
- Analyze sales trends to identify peak periods for promotions.