```
#Importing libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
```

```
#Load dataset
df = pd.read_csv('/content/Amazon Sale Report.csv')
```

```
<ipython-input-49-d238bbdf341c>:2: DtypeWarning: Columns (17) have mixed types. Specify dtype option on import or set low_memory=Fal
  df = pd.read_csv('/content/Amazon Sale Report.csv')
```

```
# Explore data
print(df.head())
```

```
   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

  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
2     Amazon     Amazon.in          Expedited     Shirt   XL        Shipped
3   Merchant     Amazon.in           Standard   Blazzer    L     On the Way
4     Amazon     Amazon.in          Expedited  Trousers  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
2  ...       INR  329.00  NAVI MUMBAI  MAHARASHTRA         410210.0
3  ...       INR  753.33   PUDUCHERRY   PUDUCHERRY         605008.0
4  ...       INR  574.00      CHENNAI   TAMIL NADU         600073.0

   ship-country    B2B fulfilled-by New  PendingS
0            IN  False    Easy Ship NaN       NaN
1            IN  False    Easy Ship NaN       NaN
2            IN   True          NaN NaN       NaN
3            IN  False    Easy Ship NaN       NaN
4            IN  False          NaN NaN       NaN

[5 rows x 21 columns]
```

```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35417 entries, 0 to 35416
Data columns (total 21 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   index               35417 non-null  int64
 1   Order ID            35417 non-null  object
 2   Date                35416 non-null  object
 3   Status              35416 non-null  object
 4   Fulfilment          35416 non-null  object
 5   Sales Channel       35416 non-null  object
 6   ship-service-level  35416 non-null  object
 7   Category            35416 non-null  object
 8   Size                35416 non-null  object
 9   Courier Status      35416 non-null  object
 10  Qty                 35416 non-null  float64
 11  currency            33261 non-null  object
 12  Amount              33261 non-null  float64
 13  ship-city           35403 non-null  object
 14  ship-state          35403 non-null  object
 15  ship-postal-code    35403 non-null  float64
 16  ship-country        35403 non-null  object
 17  B2B                 35416 non-null  object
 18  fulfilled-by        12427 non-null  object
 19  New                 0 non-null      float64
 20  PendingS            0 non-null      float64
dtypes: float64(5), int64(1), object(15)
memory usage: 5.7+ MB
None
```

```
print(df.describe())
```

```
               index           Qty        Amount  ship-postal-code  New  \
count  35417.000000  35416.000000  33261.000000      35403.000000  0.0   
mean   17705.654149      0.898069    624.160220     462525.759201  NaN   
std    10225.809051      0.334645    271.424316     194581.644203  NaN   
min        0.000000      0.000000      0.000000     110001.000000  NaN   
25%     8853.000000      1.000000    432.000000     380007.000000  NaN   
50%    17707.000000      1.000000    568.570000     500018.000000  NaN   
75%    26561.000000      1.000000    771.000000     600024.000000  NaN   
max    35415.000000     15.000000   5495.000000     989898.000000  NaN   

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

```
#Product Analysis
```

```
#Top Selling Product
top_products = df.groupby('Category')['Amount'].sum().sort_values(ascending=False).head(10)
print(top_products)
```

```
Category
T-shirt     11125870.18
Shirt        5825026.10
Blazzer      2106488.88
Trousers     1276833.84
Perfume       185205.70
Wallet        146447.17
Socks          49799.97
Shoes          44521.24
Name: Amount, dtype: float64
```

```
top_size = df.groupby('Size')['Amount'].sum()
print(top_size)
```

```
Size
3XL     2383567.92
4XL        5973.00
5XL       15966.96
6XL       15300.10
Free     376174.11
L       3432902.17
M       3722848.54
S       2934810.48
XL      3239460.95
XS      1905777.22
XXL     2727411.63
Name: Amount, dtype: float64
```

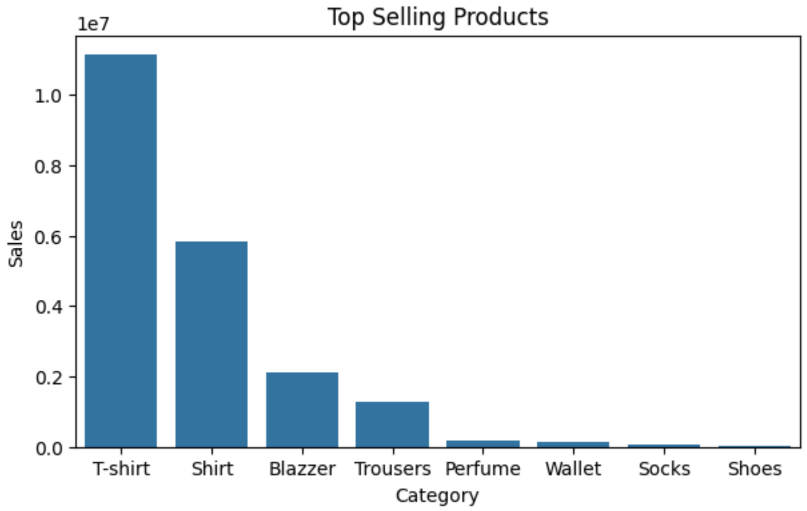
```
top_quantity = df.groupby('Qty')['Amount'].sum().sort_values(ascending=False).head(10)
print(top_quantity)
```

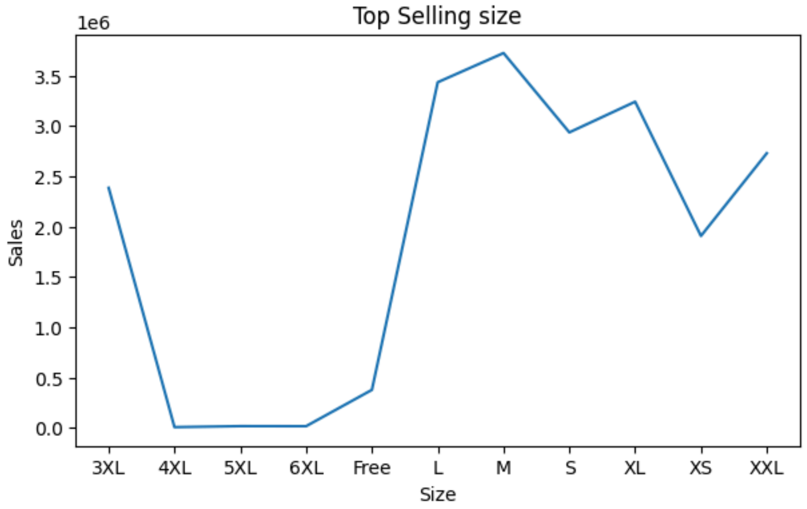
```
Qty
1.0     19663528.00
0.0       970329.08
2.0       101140.00
3.0        16317.00
5.0         5495.00
4.0         3384.00
9.0            0.00
13.0           0.00
15.0           0.00
Name: Amount, dtype: float64
```

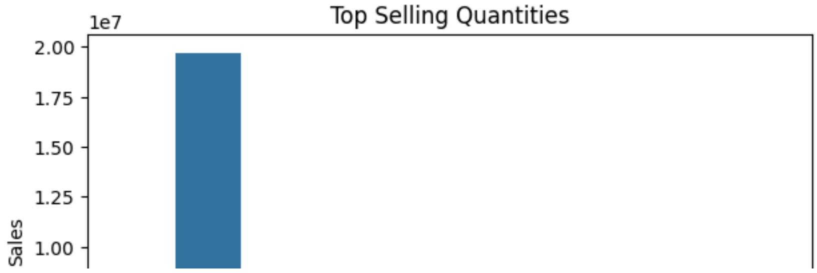
```
# Visualize the top selling products
plt.figure(figsize=(7,4))
sns.barplot(x=top_products.index, y=top_products.values)
plt.title('Top Selling Products')
plt.xlabel('Category')
plt.ylabel('Sales')
plt.show()

# Visualize the top selling sizes
plt.figure(figsize=(7,4))
sns.lineplot(x=top_size.index, y=top_size.values)
plt.title('Top Selling size')
plt.xlabel('Size')
plt.ylabel('Sales')
plt.show()

# Visualize the top selling Quantity
plt.figure(figsize=(7,4))
sns.barplot(x=top_quantity.index, y=top_quantity.values)
plt.title('Top Selling Quantities')
plt.ylabel('Sales')
plt.xlabel('Quantity')
plt.show()
```







```
#Fulfillment Analysis to average sales
ana = df.groupby('Fulfilment')['Amount'].mean()
print(ana)
```

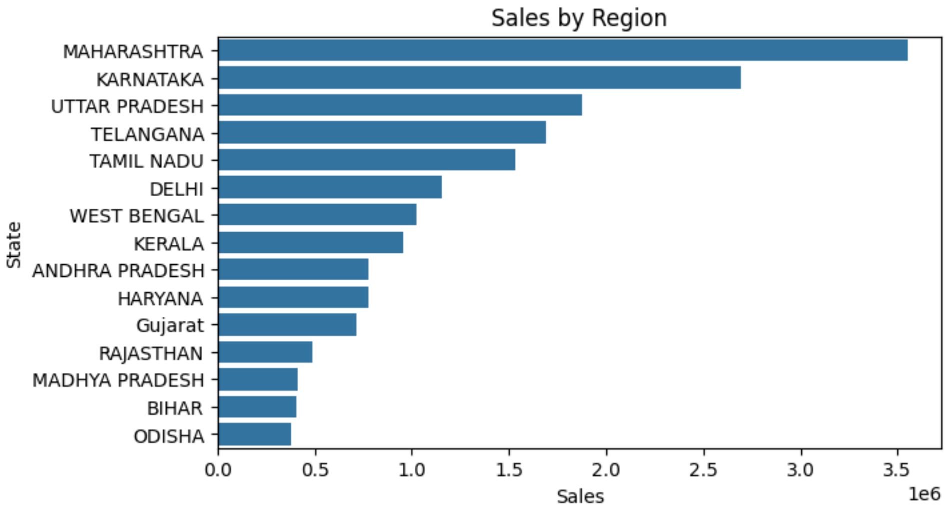
```
Fulfilment
Amazon      629.631680
```

```
Merchant    614.327145
Name: Amount, dtype: float64
```

```
# Sales by Region
region_sales = df.groupby('ship-state')['Amount'].sum().sort_values(ascending=False).head(15)
print(region_sales)
```

```
ship-state
MAHARASHTRA       3549593.89
KARNATAKA         2693442.86
UTTAR PRADESH     1875064.33
TELANGANA         1693616.20
TAMIL NADU        1531064.43
DELHI             1154753.06
WEST BENGAL       1026407.94
KERALA             955467.14
ANDHRA PRADESH     776984.06
HARYANA            774261.64
Gujarat            718589.72
RAJASTHAN          490356.23
MADHYA PRADESH     411159.77
BIHAR              405210.90
ODISHA             379909.17
Name: Amount, dtype: float64
```

```
# Visualize the sales by region
plt.figure(figsize=(7,4))
sns.barplot(y=region_sales.index, x=region_sales.values)
plt.title('Sales by Region')
plt.xlabel('Sales')
plt.ylabel('State')
plt.show()
```



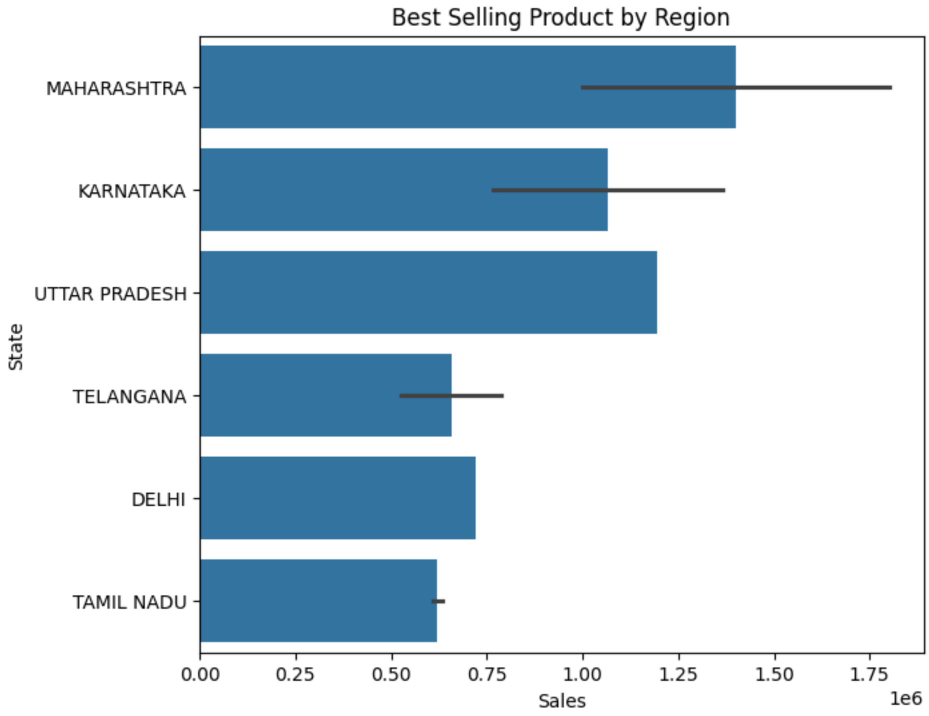
```
#Sales over view
avg_order_value = df['Amount'].mean()
print(f'Average Order Value: {avg_order_value:.2f} INR')
```

```
Average Order Value: 624.16 INR
```

```
best_selling_product_by_region = df.groupby(['ship-state', 'Category'])['Amount'].sum().sort_values(ascending=False).head(10)
print(best_selling_product_by_region)
```

```
ship-state     Category
MAHARASHTRA    T-shirt     1800753.34
KARNATAKA      T-shirt     1364233.32
UTTAR PRADESH  T-shirt     1195907.40
MAHARASHTRA    Shirt        999846.08
TELANGANA      T-shirt      788095.47
KARNATAKA      Shirt        767239.67
DELHI          T-shirt      721870.52
TAMIL NADU     Shirt        633120.77
               T-shirt      608900.40
TELANGANA      Shirt        526870.62
Name: Amount, dtype: float64
```

```
#Visualize the best selling product by region
plt.figure(figsize=(7,6))
sns.barplot(y=best_selling_product_by_region.index.get_level_values('ship-state'), x=best_selling_product_by_region.values)
plt.title('Best Selling Product by Region')
plt.xlabel('Sales')
plt.ylabel('State')
plt.show()
```



```
#Insights and Recommendations based on analysis

print(" Insights and Recommendations based on analysis:")
print("1.Recognize the needs, tastes, and purchasing patterns of your target market to customize your offering.")
print("2.Identify the best-performing sales channels.")
print("3.Consider market demand, financial objectives, and product value when determining pricing and discounting strategies.")
print("4.Engage clients through a variety of channels to ensure a smooth experience.")
print("5.To increase customer happiness, provide individualized care and top-notch post-purchase assistance.")
```

```
 Insights and Recommendations based on analysis:
1.Recognize the needs, tastes, and purchasing patterns of your target market to customize your offering.
2.Identify the best-performing sales channels.
3.Consider market demand, financial objectives, and product value when determining pricing and discounting strategies.
4.Engage clients through a variety of channels to ensure a smooth experience.
5.To increase customer happiness, provide individualized care and top-notch post-purchase assistance.
```