AMAZON_EDA

December 2, 2025

[3]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings, os
warnings.filterwarnings('ignore')

%matplotlib inline

sns.set_theme(style='whitegrid')
sns.set_palette('husl')
```

[5]:

```
os.makedirs('charts', exist_ok=True)
```

1 LOAD DATA

[8]:

```
df = pd.read_csv("Amazon Sale Report.csv", low_memory=False)
```

[17]:

```
df.head(5)
```

[17]:

```
   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]
```

1.0.1 Data Cleaning

[20]:

```
df['Date'] = pd.to_datetime(df['Date'], errors='coerce')
```

[22]:

```
df.fillna("Unknown", inplace=True)
```

[24]:

```
df['Amount'] = pd.to_numeric(df['Amount'], errors='coerce')
df['Amount'].fillna(0, inplace=True)
```

[26]:

```
df.info()
df.describe()
```

```
<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  datetime64[ns]
 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            128976 non-null  object
 12  Amount              128976 non-null  float64
 13  ship-city           128976 non-null  object
 14  ship-state          128976 non-null  object
 15  ship-postal-code    128976 non-null  object
 16  ship-country        128976 non-null  object
 17  B2B                 128976 non-null  bool
 18  fulfilled-by        128976 non-null  object
 19  New                 128976 non-null  object
 20  PendingS            128976 non-null  object
dtypes: bool(1), datetime64[ns](1), float64(1), int64(2), object(16)
memory usage: 19.8+ MB
```

[26]:

```
               index                           Date            Qty  \
count  128976.000000                         128976  128976.000000
mean    64486.130427  2022-05-12 11:49:26.951991040       0.904401
min         0.000000            2022-03-31 00:00:00       0.000000
25%     32242.750000            2022-04-20 00:00:00       1.000000
50%     64486.500000            2022-05-10 00:00:00       1.000000
75%     96730.250000            2022-06-04 00:00:00       1.000000
max    128974.000000            2022-06-29 00:00:00      15.000000
std     37232.897832                            NaN       0.313368

              Amount
count  128976.000000
mean      609.339491
min         0.000000
25%       413.000000
50%       583.000000
75%       771.000000
max      5584.000000
std       313.342529
```

1.1 SALES OVERVIEW

[29]:

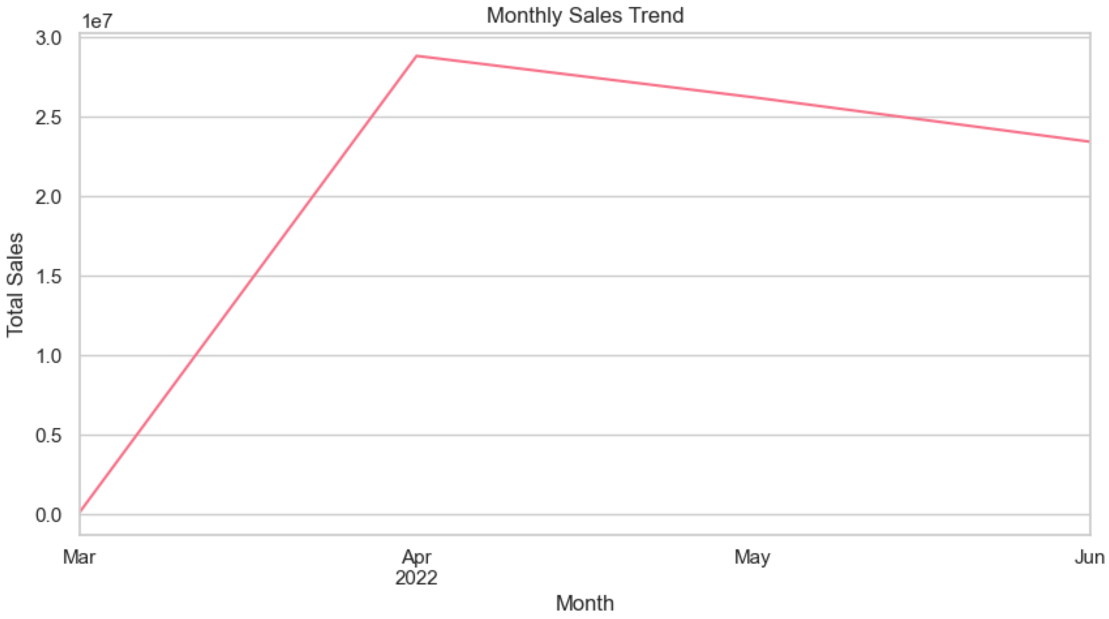
```
df['Amount'].sum()
```

[29]:

```
78590170.24999997
```

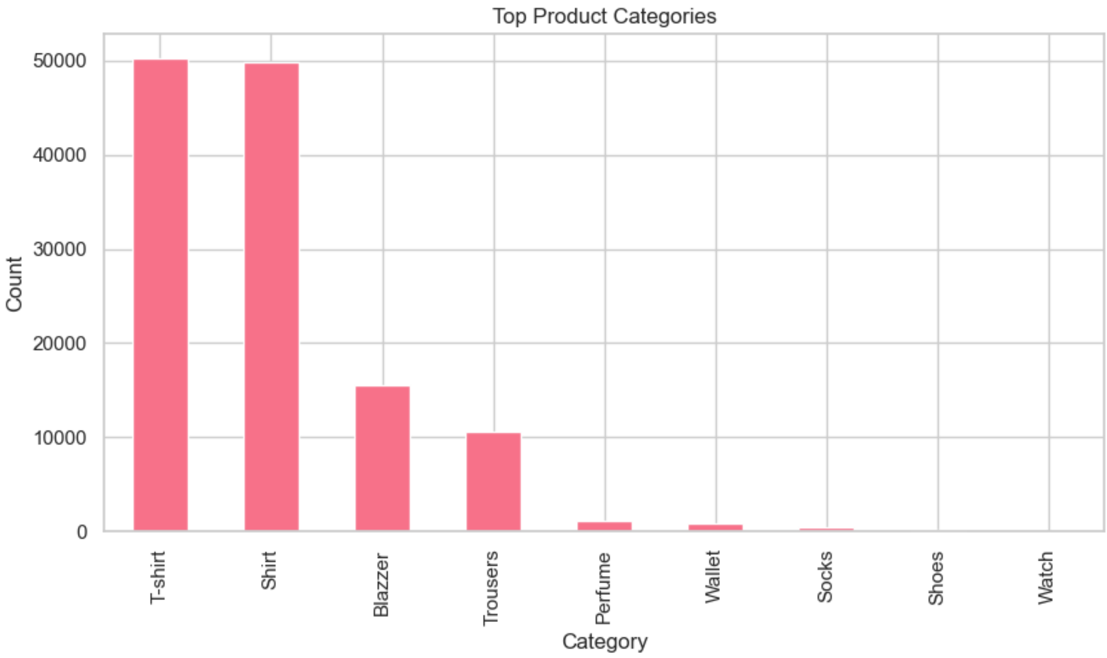
[31]:

```
df['Month'] = df['Date'].dt.to_period('M')

monthly_sales = df.groupby('Month')['Amount'].sum()

monthly_sales.plot(kind='line', figsize=(10,5))
plt.title("Monthly Sales Trend")
plt.ylabel("Total Sales")
plt.show()
```

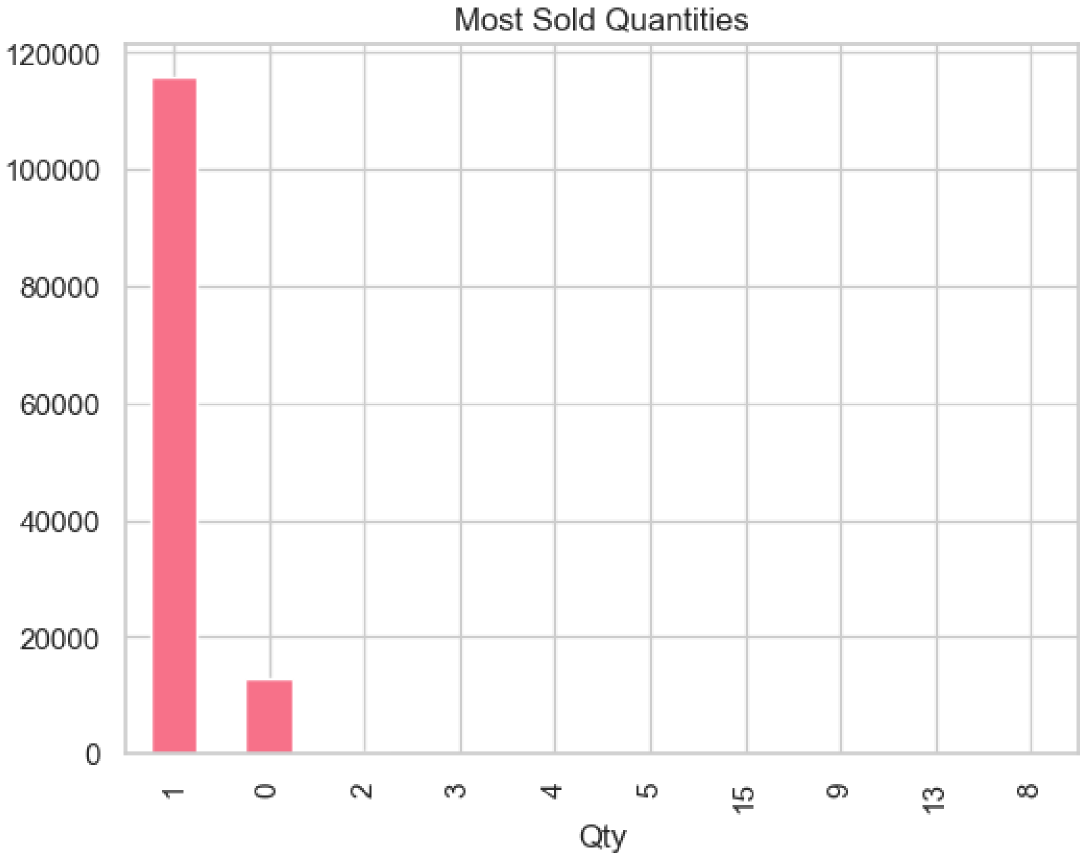


1.2 PRODUCT ANALYSIS

[34]:
```
df['Category'].value_counts().head(10).plot(kind='bar', figsize=(10,5))
plt.title("Top Product Categories")
plt.ylabel("Count")
plt.show()
```

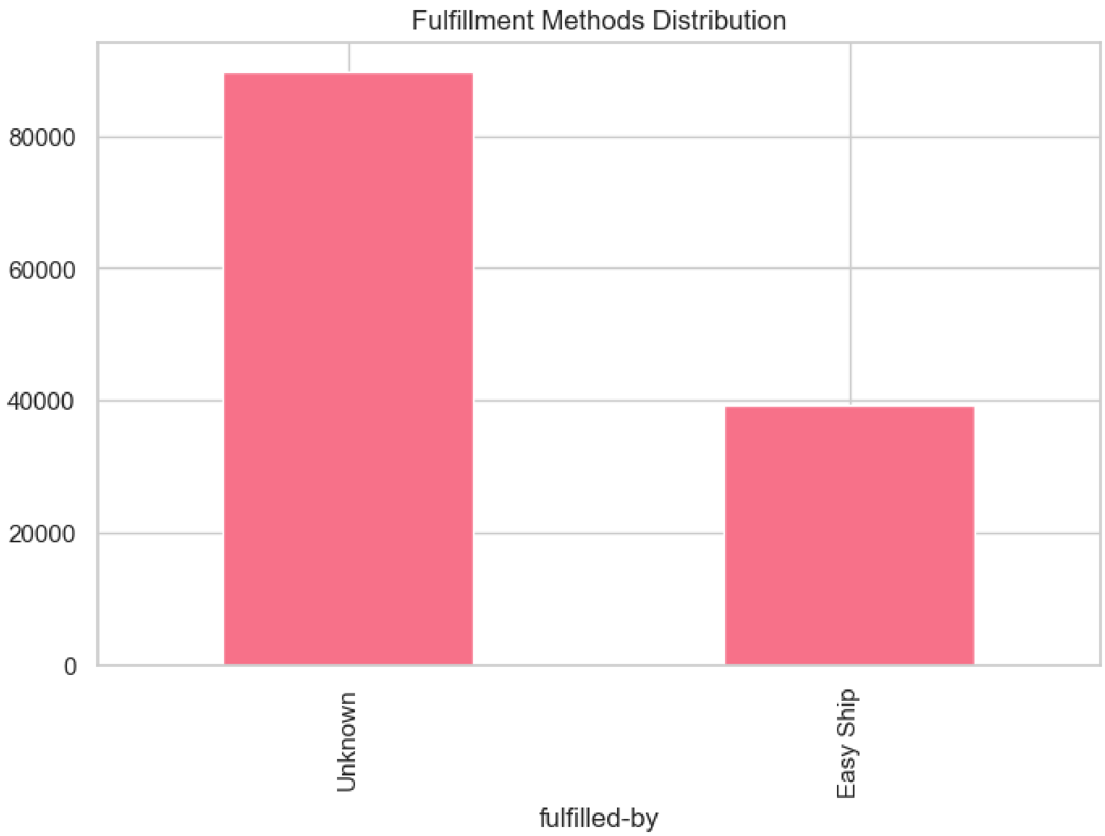


[36]:
```
df['Qty'].value_counts().head(10).plot(kind='bar')
plt.title("Most Sold Quantities")
plt.show()
```



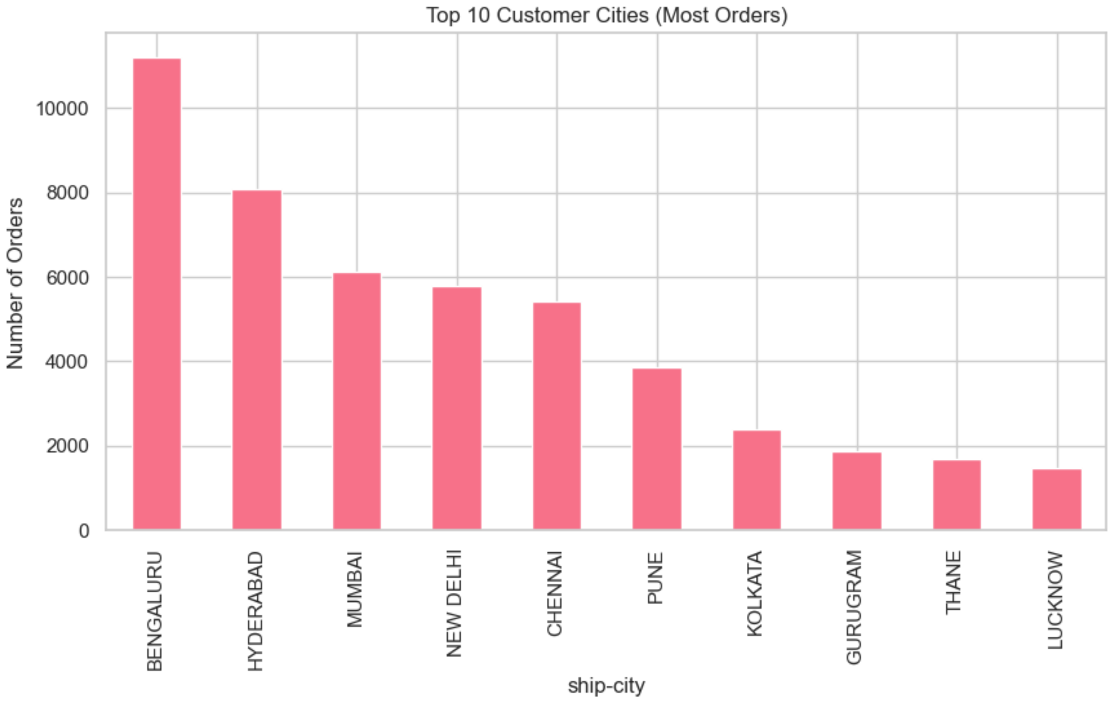
1.3 FULFILLMENT ANALYSIS

[42]:
```
df['fulfilled-by'].value_counts().plot(kind='bar', figsize=(8,5))
plt.title("Fulfillment Methods Distribution")
plt.show()
```



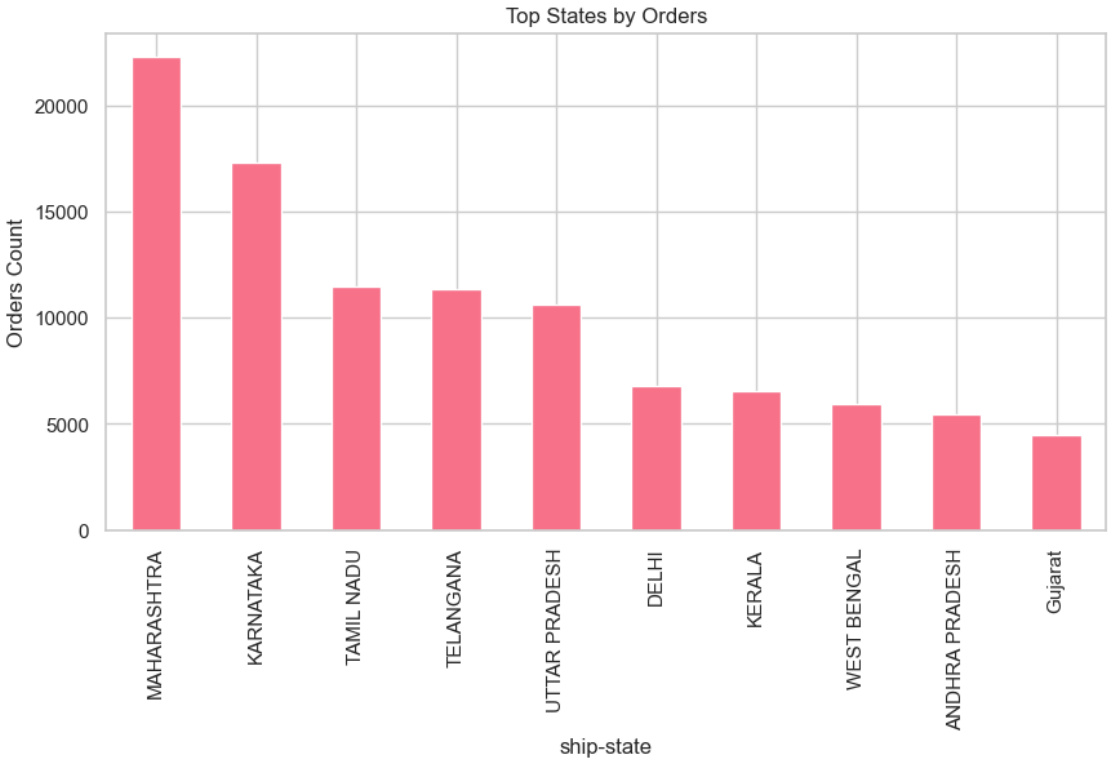
1.4 CUSTOMER SEGMENTATION

```
[48]: city_orders = df.groupby('ship-city')['Order ID'].count().
       ↪sort_values(ascending=False)

      city_orders.head(10).plot(kind='bar', figsize=(10,5))
      plt.title("Top 10 Customer Cities (Most Orders)")
      plt.ylabel("Number of Orders")
      plt.show()
```

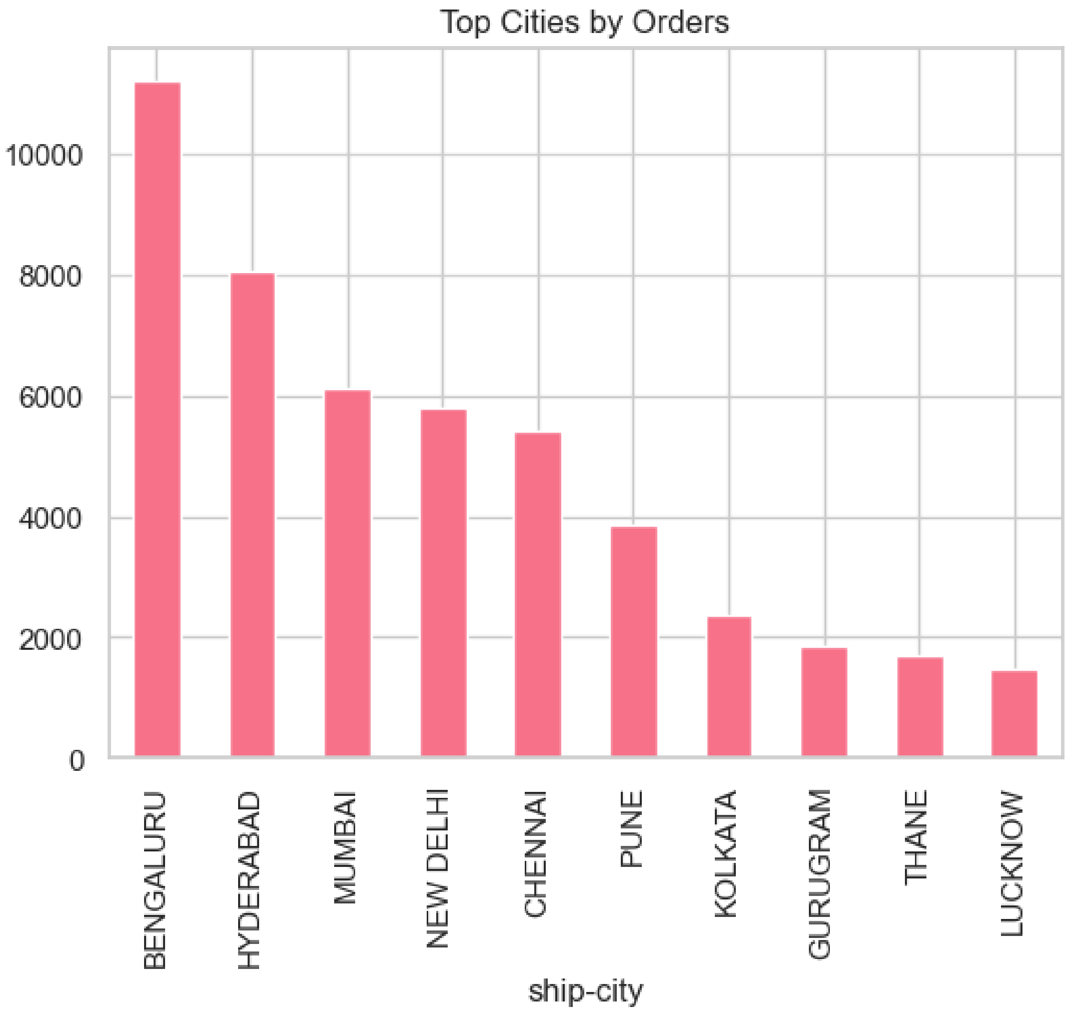


1.5 GEOGRAPHICAL ANALYSIS

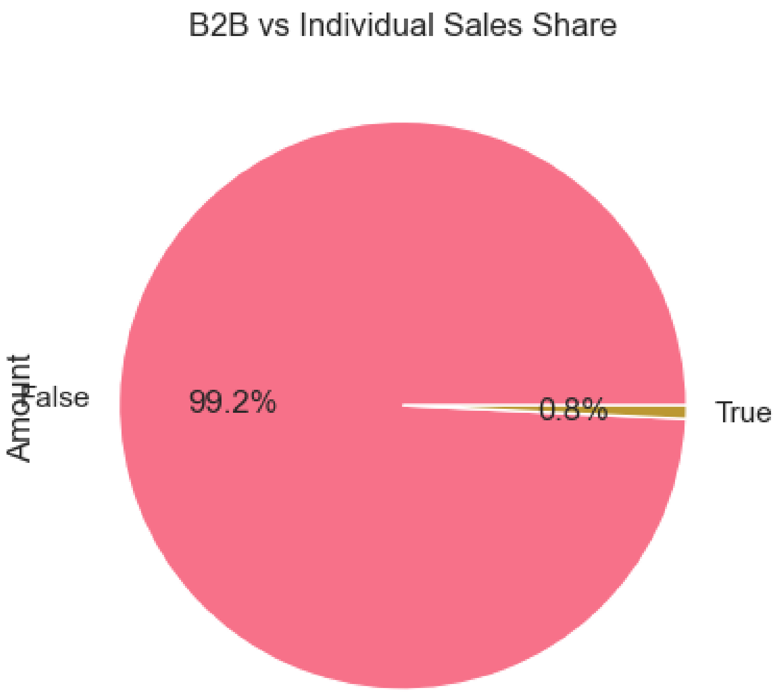
```
[51]: df['ship-state'].value_counts().head(10).plot(kind='bar', figsize=(10,5))
      plt.title("Top States by Orders")
      plt.ylabel("Orders Count")
      plt.show()
```



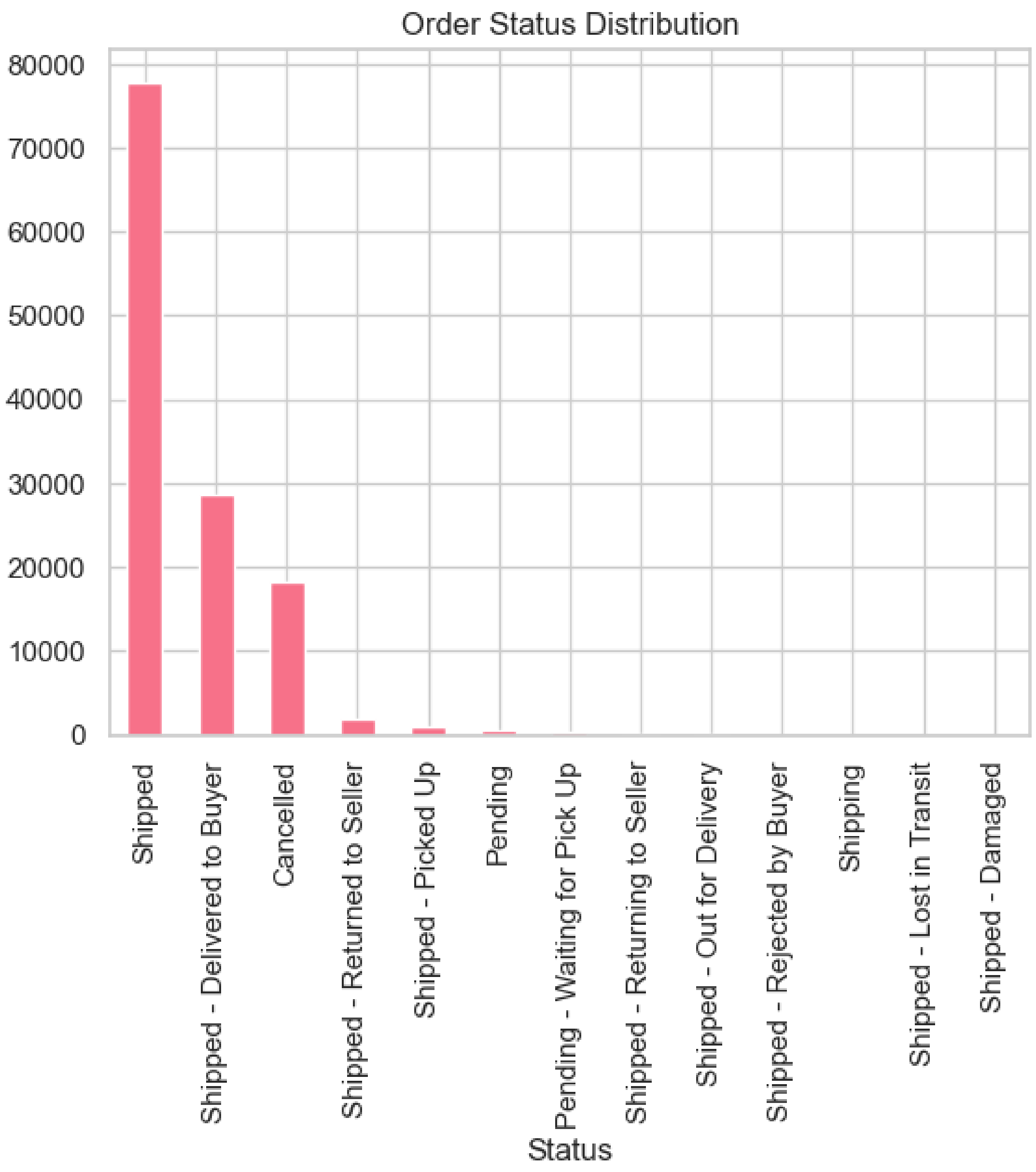
[53]:
```
df['ship-city'].value_counts().head(10).plot(kind='bar')
plt.title("Top Cities by Orders")
plt.show()
```



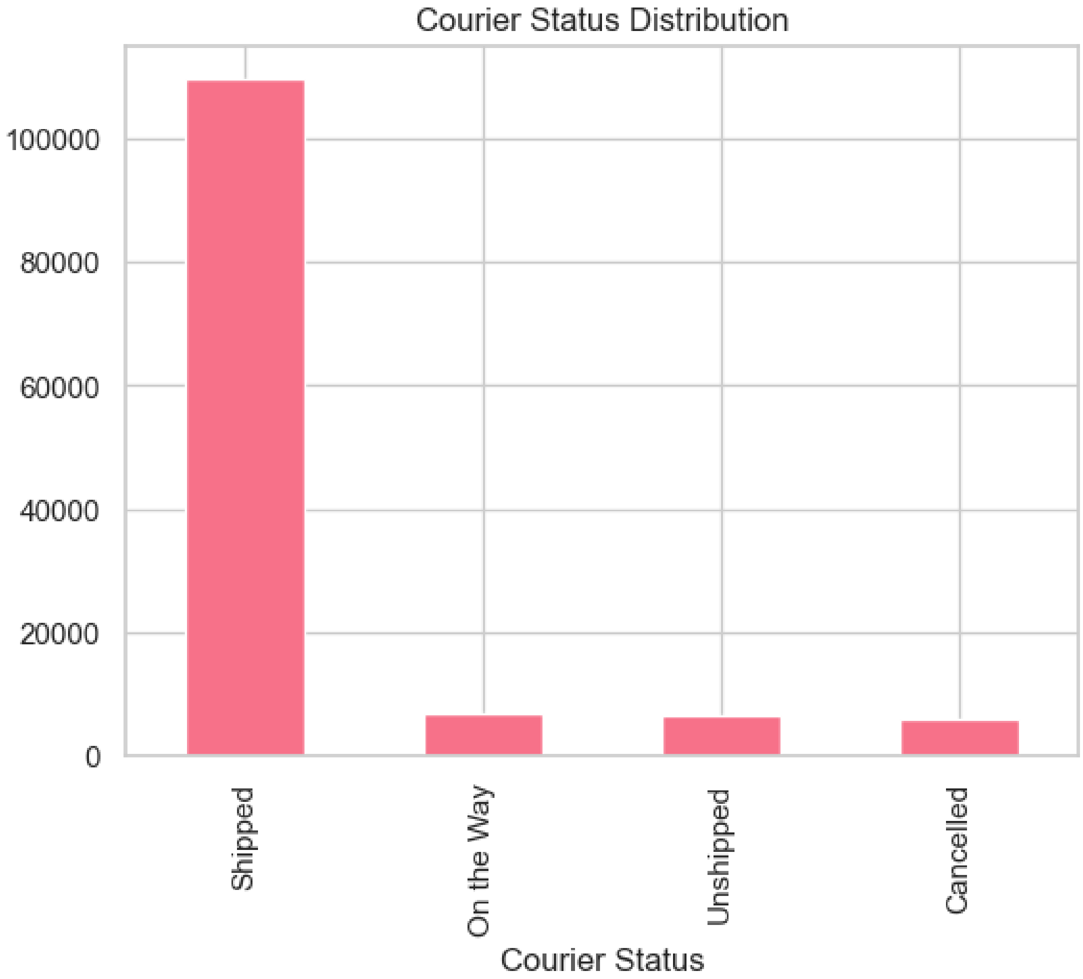
[65]:
```
# B2B vs Individual Sales Share Pie Chart
b2b_sales = df.groupby('B2B')['Amount'].sum()
b2b_sales.plot(kind='pie', autopct='%1.1f%%')
plt.title('B2B vs Individual Sales Share')
plt.savefig('b2b_sales_pie.png')
plt.show()
```



```
[73]: df['Status'].value_counts().plot(kind='bar')
      plt.title('Order Status Distribution')
      plt.savefig('order_status_bar.png')
      plt.show()
```

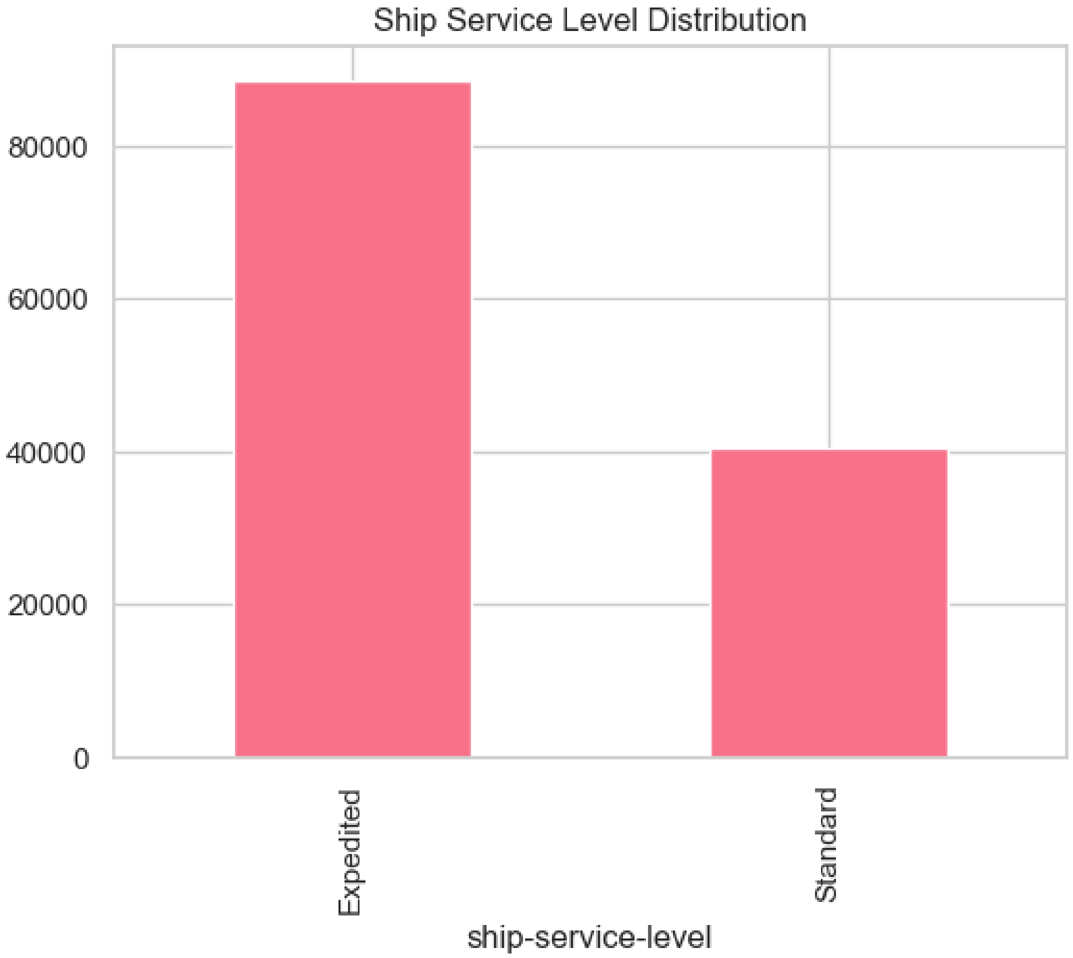


[75]:
```
df['Courier Status'].value_counts().plot(kind='bar')
plt.title('Courier Status Distribution')
plt.savefig('courier_status_bar.png')
plt.show()
```



[77]:

```
df['ship-service-level'].value_counts().plot(kind='bar')
plt.title('Ship Service Level Distribution')
plt.savefig('ship_service_bar.png')
plt.show()
```


Ship Service Level Distribution
80000
60000
40000
20000
0
Expedited
Standard
ship-service-level

```
[ ]:
```