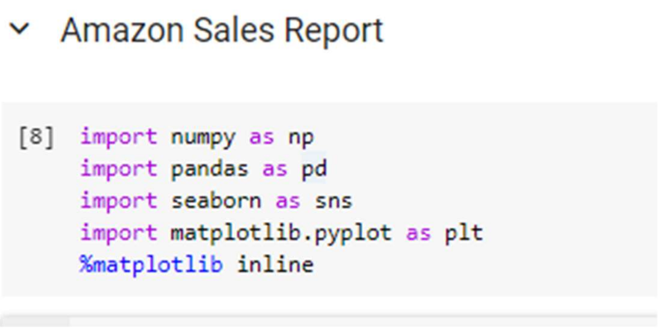
Problem Statement:

Analyze and Provide Insights on Amazon Sales Report

Problem Description:

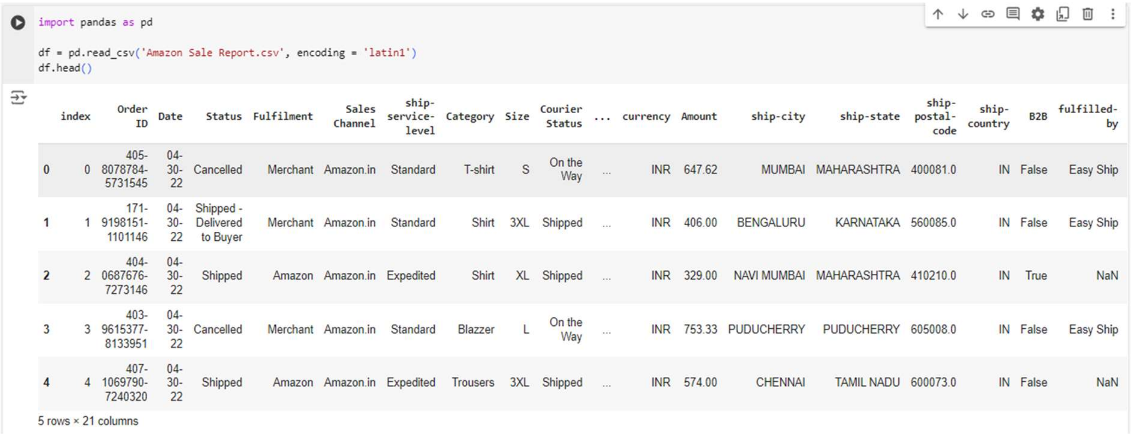
The provided dataset contains information about sales transactions on Amazon, including details such as order ID, date, status, fulfillment method, sales channel, product category, size, quantity, amount, shipping details, and more. The objective is to conduct a comprehensive analysis of the data and extract actionable insights to support business decision-making.

Problem Solution:



Amazon Sales Report

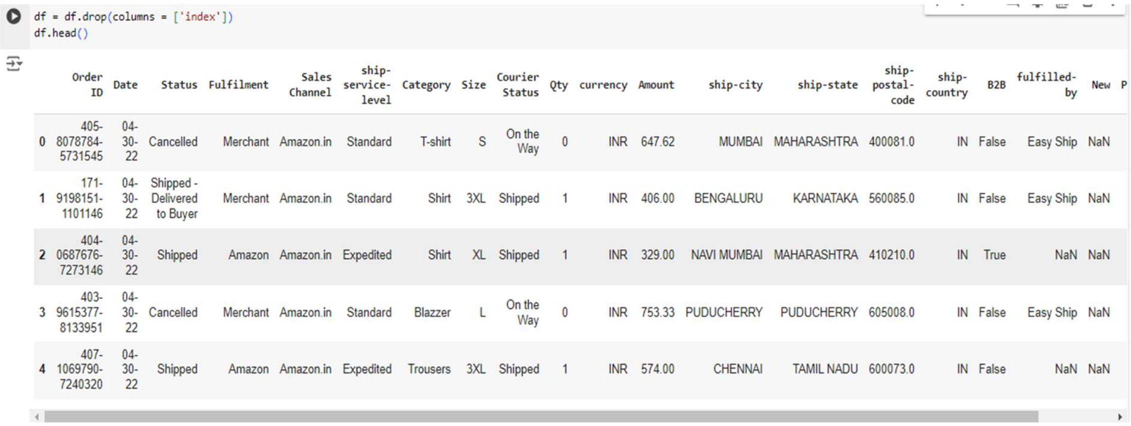
```
[8] import numpy as np
    import pandas as pd
    import seaborn as sns
    import matplotlib.pyplot as plt
    %matplotlib inline
```



```
import pandas as pd

df = pd.read_csv('Amazon Sale Report.csv', encoding = 'latin1')
df.head()
```

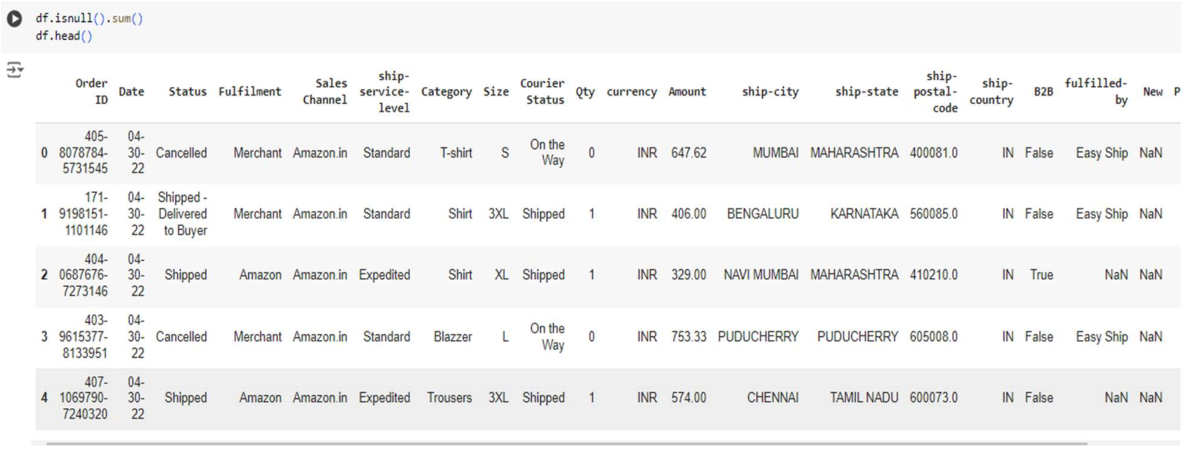
	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	...	currency	Amount	ship-city	ship-state	ship-postal-code	ship-country	B2B	fulfilled-by
0	0	405-8078784-5731545	04-30-22	Cancelled	Merchant	Amazon.in	Standard	T-shirt	S	On the Way	...	INR	647.62	MUMBAI	MAHARASHTRA	400081.0	IN	False	Easy Ship
1	1	171-9198151-1101146	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
2	2	404-0687676-7273146	04-30-22	Shipped	Amazon	Amazon.in	Expedited	Shirt	XL	Shipped	...	INR	329.00	NAVI MUMBAI	MAHARASHTRA	410210.0	IN	True	NaN
3	3	403-9615377-8133951	04-30-22	Cancelled	Merchant	Amazon.in	Standard	Blazzer	L	On the Way	...	INR	753.33	PUDUCHERRY	PUDUCHERRY	605008.0	IN	False	Easy Ship
4	4	407-1069790-7240320	04-30-22	Shipped	Amazon	Amazon.in	Expedited	Trousers	3XL	Shipped	...	INR	574.00	CHENNAI	TAMIL NADU	600073.0	IN	False	NaN

5 rows × 21 columns



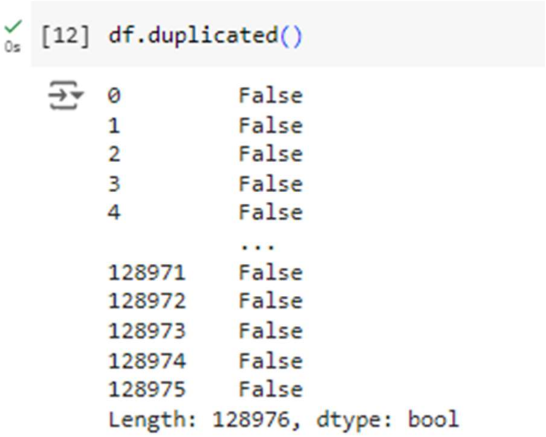
```
df = df.drop(columns = ['index'])
df.head()
```

	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	Qty	currency	Amount	ship-city	ship-state	ship-postal-code	ship-country	B2B	fulfilled-by	New	P
0	405-8078784-5731545	04-30-22	Cancelled	Merchant	Amazon.in	Standard	T-shirt	S	On the Way	0	INR	647.62	MUMBAI	MAHARASHTRA	400081.0	IN	False	Easy Ship	NaN	
1	171-9198151-1101146	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	3XL	Shipped	1	INR	406.00	BENGALURU	KARNATAKA	560085.0	IN	False	Easy Ship	NaN	
2	404-0687676-7273146	04-30-22	Shipped	Amazon	Amazon.in	Expedited	Shirt	XL	Shipped	1	INR	329.00	NAVI MUMBAI	MAHARASHTRA	410210.0	IN	True	NaN	NaN	
3	403-9615377-8133951	04-30-22	Cancelled	Merchant	Amazon.in	Standard	Blazzer	L	On the Way	0	INR	753.33	PUDUCHERRY	PUDUCHERRY	605008.0	IN	False	Easy Ship	NaN	
4	407-1069790-7240320	04-30-22	Shipped	Amazon	Amazon.in	Expedited	Trousers	3XL	Shipped	1	INR	574.00	CHENNAI	TAMIL NADU	600073.0	IN	False	NaN	NaN	



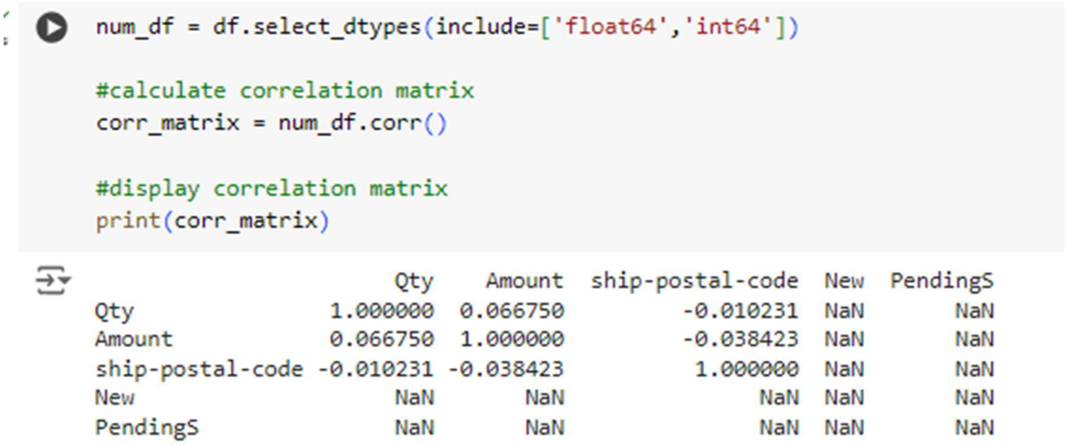
```
df.isnull().sum()
df.head()
```

	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Category	Size	Courier Status	Qty	currency	Amount	ship-city	ship-state	ship-postal-code	ship-country	B2B	fulfilled-by	New	P
0	405-8078784-5731545	04-30-22	Cancelled	Merchant	Amazon.in	Standard	T-shirt	S	On the Way	0	INR	647.62	MUMBAI	MAHARASHTRA	400081.0	IN	False	Easy Ship	NaN	
1	171-9198151-1101146	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	Shirt	3XL	Shipped	1	INR	406.00	BENGALURU	KARNATAKA	560085.0	IN	False	Easy Ship	NaN	
2	404-0687676-7273146	04-30-22	Shipped	Amazon	Amazon.in	Expedited	Shirt	XL	Shipped	1	INR	329.00	NAVI MUMBAI	MAHARASHTRA	410210.0	IN	True	NaN	NaN	
3	403-9615377-8133951	04-30-22	Cancelled	Merchant	Amazon.in	Standard	Blazzer	L	On the Way	0	INR	753.33	PUDUCHERRY	PUDUCHERRY	605008.0	IN	False	Easy Ship	NaN	
4	407-1069790-7240320	04-30-22	Shipped	Amazon	Amazon.in	Expedited	Trousers	3XL	Shipped	1	INR	574.00	CHENNAI	TAMIL NADU	600073.0	IN	False	NaN	NaN	



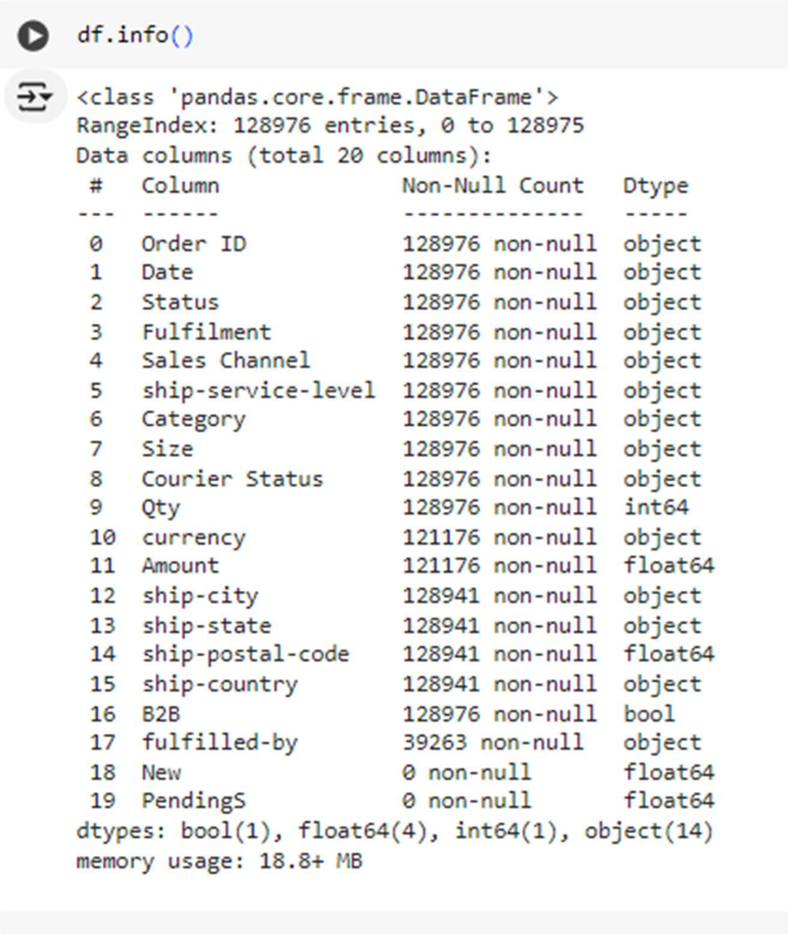
```
[12] df.duplicated()
```

```
0         False
1         False
2         False
3         False
4         False
          ...
128971    False
128972    False
128973    False
128974    False
128975    False
Length: 128976, dtype: bool
```



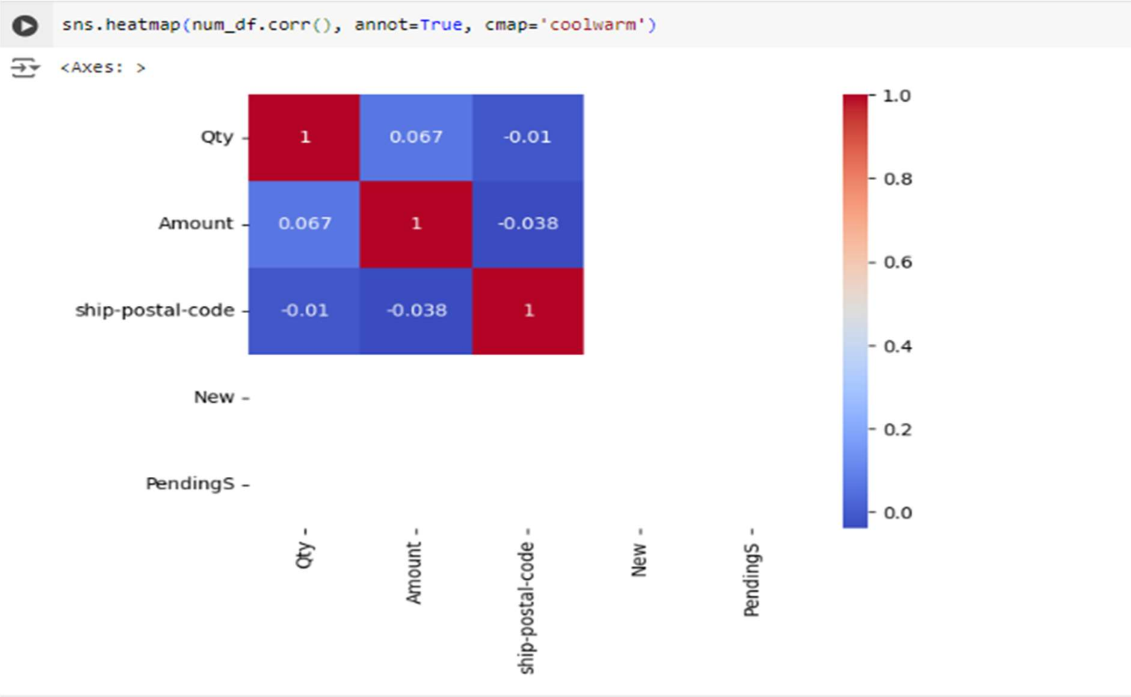
```
num_df = df.select_dtypes(include=['float64','int64'])

#calculate correlation matrix
corr_matrix = num_df.corr()

#display correlation matrix
print(corr_matrix)
```

```
                       Qty     Amount  ship-postal-code  New  PendingS
Qty               1.000000  0.066750         -0.010231  NaN       NaN
Amount            0.066750  1.000000         -0.038423  NaN       NaN
ship-postal-code -0.010231 -0.038423          1.000000  NaN       NaN
New                    NaN       NaN               NaN  NaN       NaN
PendingS               NaN       NaN               NaN  NaN       NaN
```



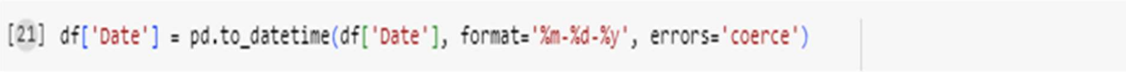
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 128976 entries, 0 to 128975
Data columns (total 20 columns):
 #   Column              Non-Null Count   Dtype  
---  ------              --------------   -----  
 0   Order ID            128976 non-null  object 
 1   Date                128976 non-null  object 
 2   Status              128976 non-null  object 
 3   Fulfilment          128976 non-null  object 
 4   Sales Channel       128976 non-null  object 
 5   ship-service-level  128976 non-null  object 
 6   Category            128976 non-null  object 
 7   Size                128976 non-null  object 
 8   Courier Status      128976 non-null  object 
 9   Qty                 128976 non-null  int64  
 10  currency            121176 non-null  object 
 11  Amount              121176 non-null  float64
 12  ship-city           128941 non-null  object 
 13  ship-state          128941 non-null  object 
 14  ship-postal-code    128941 non-null  float64
 15  ship-country        128941 non-null  object 
 16  B2B                 128976 non-null  bool   
 17  fulfilled-by        39263 non-null   object 
 18  New                 0 non-null       float64
 19  PendingS            0 non-null       float64
dtypes: bool(1), float64(4), int64(1), object(14)
memory usage: 18.8+ MB
```

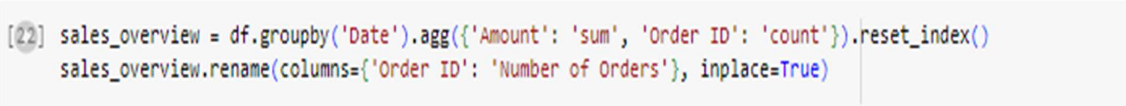


```
sns.heatmap(num_df.corr(), annot=True, cmap='coolwarm')
```

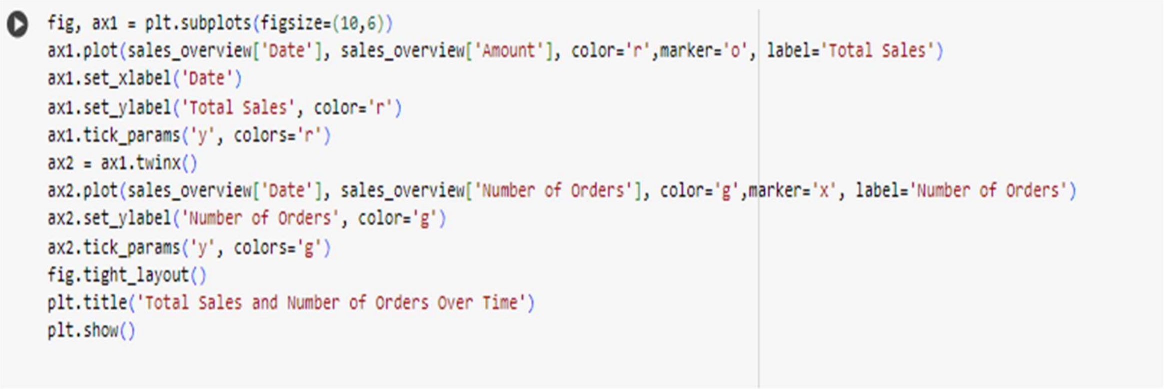
```
<Axes: >
```



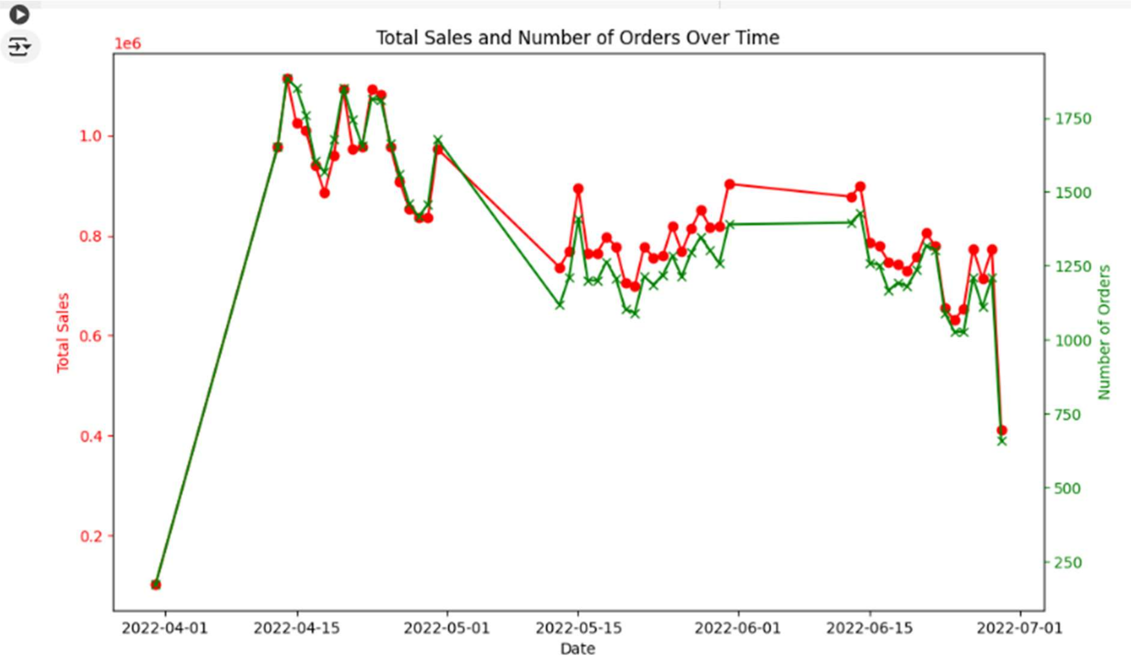
```
[21] df['Date'] = pd.to_datetime(df['Date'], format='%m-%d-%y', errors='coerce')
```

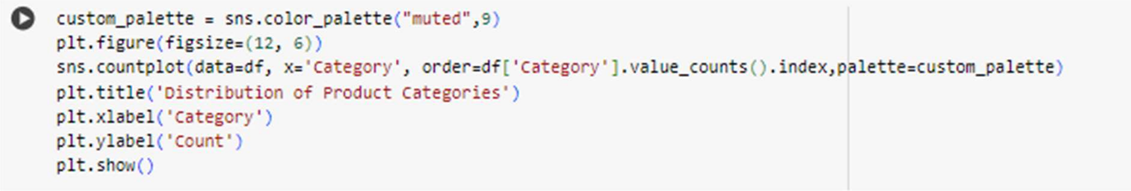


```
[22] sales_overview = df.groupby('Date').agg({'Amount': 'sum', 'Order ID': 'count'}).reset_index()
     sales_overview.rename(columns={'Order ID': 'Number of Orders'}, inplace=True)
```



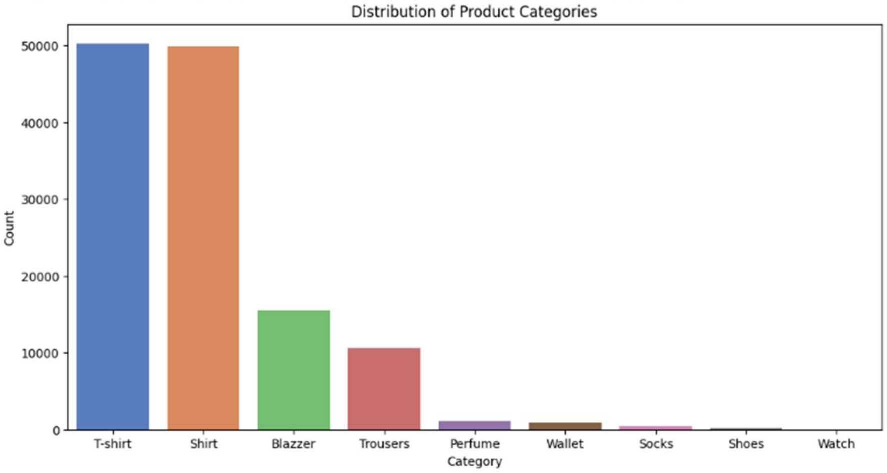
```
fig, ax1 = plt.subplots(figsize=(10,6))
ax1.plot(sales_overview['Date'], sales_overview['Amount'], color='r',marker='o', label='Total Sales')
ax1.set_xlabel('Date')
ax1.set_ylabel('Total Sales', color='r')
ax1.tick_params('y', colors='r')
ax2 = ax1.twinx()
ax2.plot(sales_overview['Date'], sales_overview['Number of Orders'], color='g',marker='x', label='Number of Orders')
ax2.set_ylabel('Number of Orders', color='g')
ax2.tick_params('y', colors='g')
fig.tight_layout()
plt.title('Total Sales and Number of Orders Over Time')
plt.show()
```

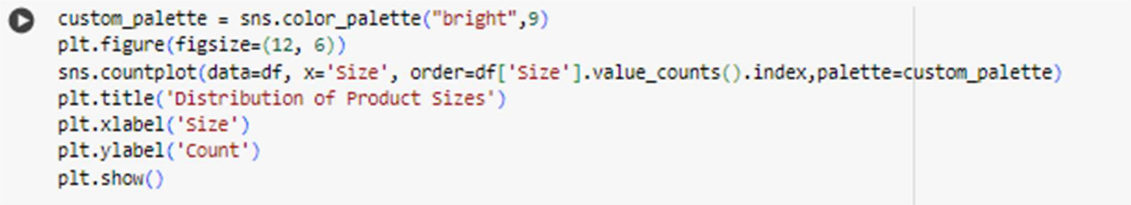




```
custom_palette = sns.color_palette("muted",9)
plt.figure(figsize=(12, 6))
sns.countplot(data=df, x='Category', order=df['Category'].value_counts().index,palette=custom_palette)
plt.title('Distribution of Product Categories')
plt.xlabel('Category')
plt.ylabel('Count')
plt.show()
```

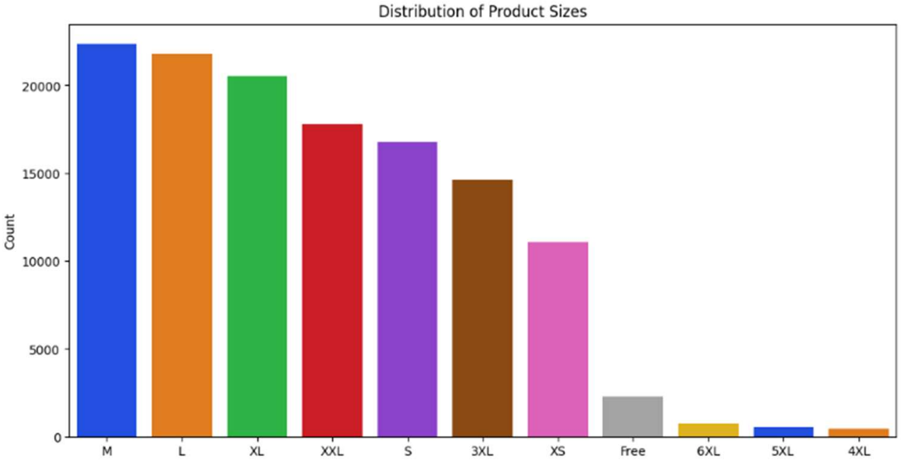
```
<ipython-input-25-2c619c65a554>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect

  sns.countplot(data=df, x='Category', order=df['Category'].value_counts().index,palette=custom_palette)
```

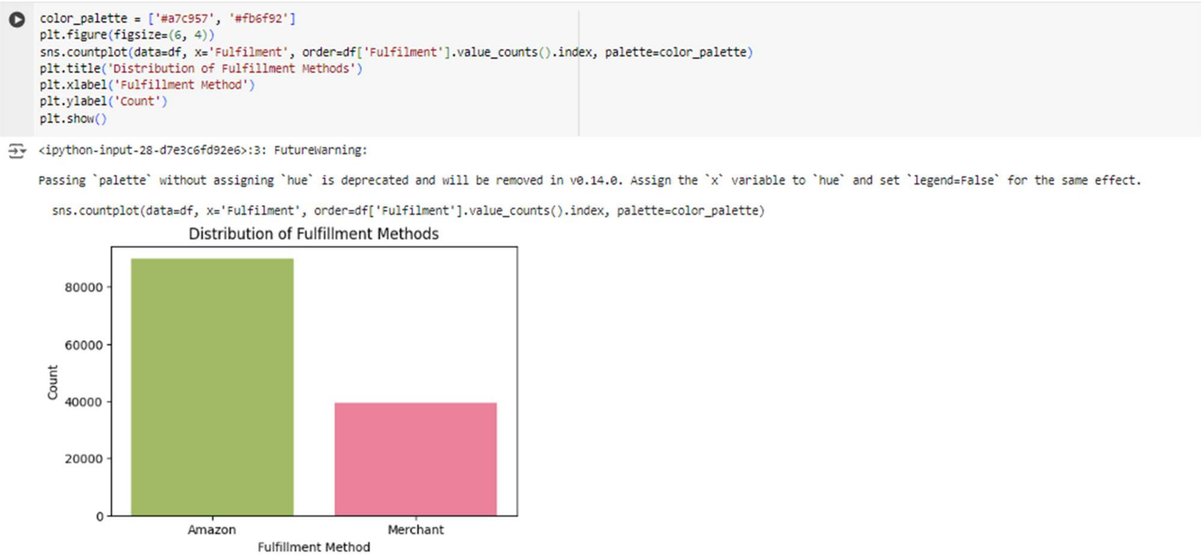




```
custom_palette = sns.color_palette("bright",9)
plt.figure(figsize=(12, 6))
sns.countplot(data=df, x='Size', order=df['Size'].value_counts().index,palette=custom_palette)
plt.title('Distribution of Product Sizes')
plt.xlabel('Size')
plt.ylabel('Count')
plt.show()
```

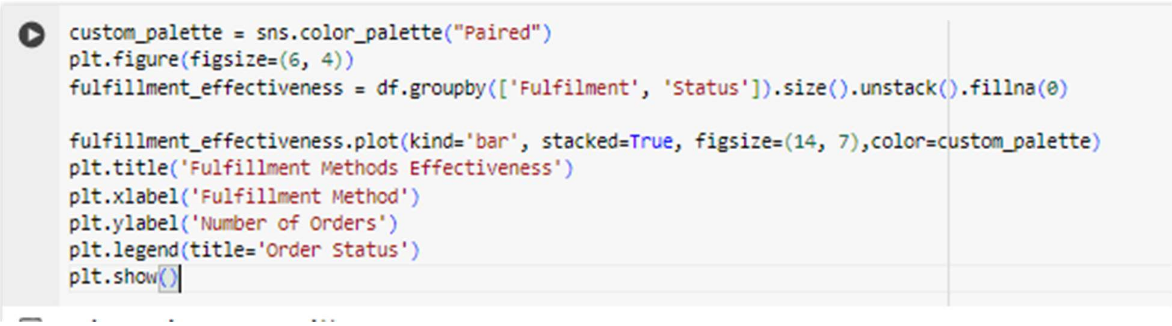
```
<ipython-input-27-c90f8ecdede7>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same eff

  sns.countplot(data=df, x='Size', order=df['Size'].value_counts().index,palette=custom_palette)
<ipython-input-27-c90f8ecdede7>:3: UserWarning:
The palette list has fewer values (9) than needed (11) and will cycle, which may produce an uninterpretable plot.
  sns.countplot(data=df, x='Size', order=df['Size'].value_counts().index,palette=custom_palette)
```



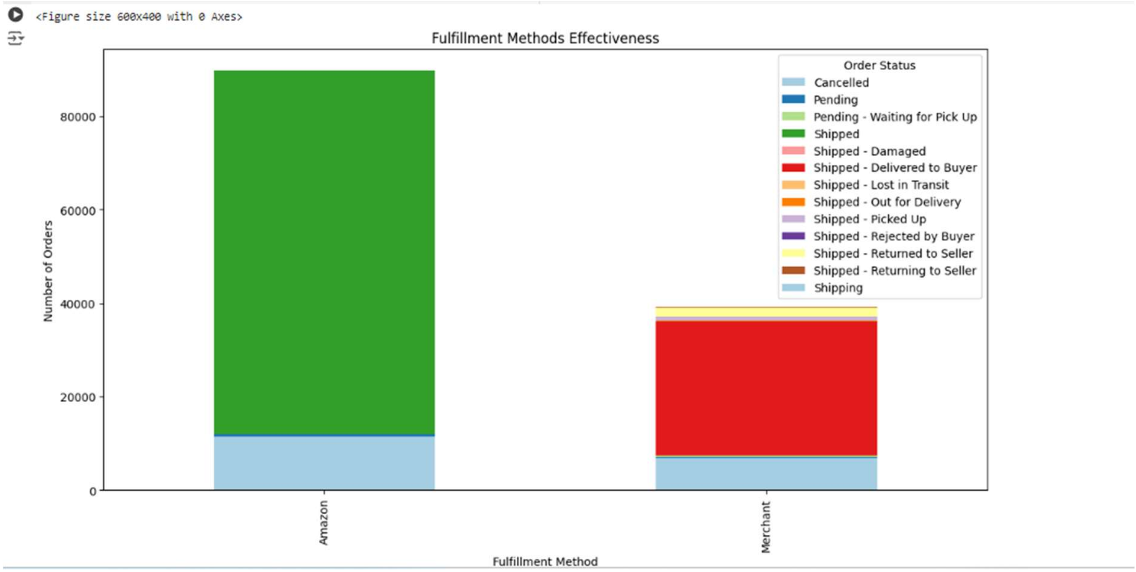


```
color_palette = ['#a7c957', '#fb6f92']
plt.figure(figsize=(6, 4))
sns.countplot(data=df, x='Fulfilment', order=df['Fulfilment'].value_counts().index, palette=color_palette)
plt.title('Distribution of Fulfillment Methods')
plt.xlabel('Fulfillment Method')
plt.ylabel('Count')
plt.show()
```

```
<ipython-input-28-d7e3c6fd92e6>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.countplot(data=df, x='Fulfilment', order=df['Fulfilment'].value_counts().index, palette=color_palette)
```

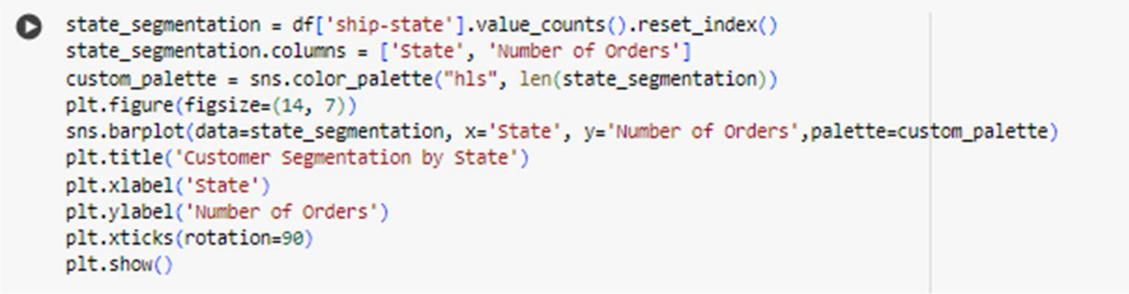


```
custom_palette = sns.color_palette("Paired")
plt.figure(figsize=(6, 4))
fulfillment_effectiveness = df.groupby(['Fulfilment', 'Status']).size().unstack().fillna(0)

fulfillment_effectiveness.plot(kind='bar', stacked=True, figsize=(14, 7),color=custom_palette)
plt.title('Fulfillment Methods Effectiveness')
plt.xlabel('Fulfillment Method')
plt.ylabel('Number of Orders')
plt.legend(title='Order Status')
plt.show()
```

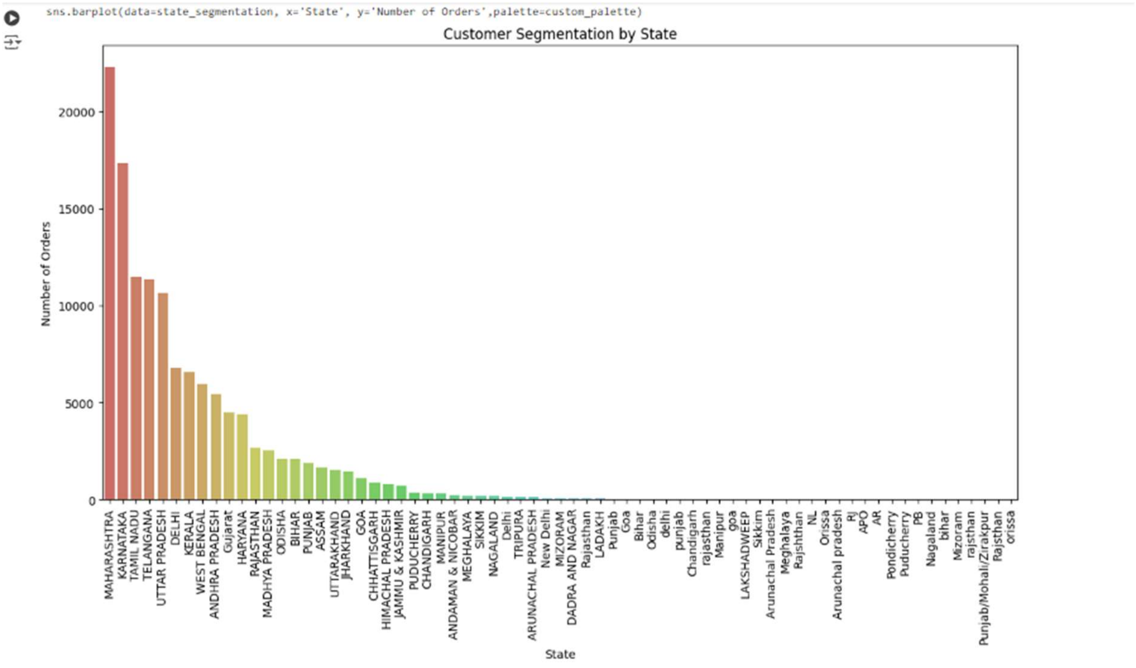


```
<Figure size 600x400 with 0 Axes>
```

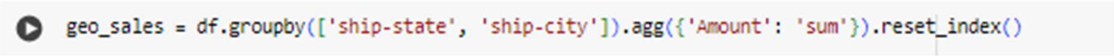
Fulfillment Methods Effectiveness
Order Status
Cancelled
Pending
Pending - Waiting for Pick Up
Shipped
Shipped - Damaged
Shipped - Delivered to Buyer
Shipped - Lost in Transit
Shipped - Out for Delivery
Shipped - Picked Up
Shipped - Rejected by Buyer
Shipped - Returned to Seller
Shipped - Returning to Seller
Shipping
80000
60000
40000
20000
0
Number of Orders
Amazon
Merchant
Fulfillment Method



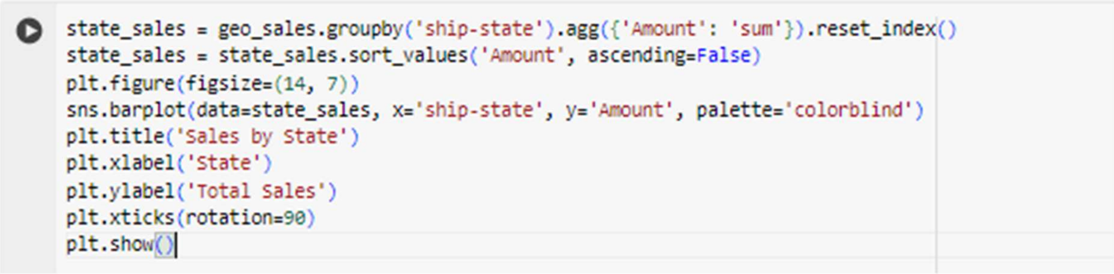
```
state_segmentation = df['ship-state'].value_counts().reset_index()
state_segmentation.columns = ['State', 'Number of Orders']
custom_palette = sns.color_palette("hls", len(state_segmentation))
plt.figure(figsize=(14, 7))
sns.barplot(data=state_segmentation, x='State', y='Number of Orders',palette=custom_palette)
plt.title('Customer Segmentation by State')
plt.xlabel('State')
plt.ylabel('Number of Orders')
plt.xticks(rotation=90)
plt.show()
```



```
sns.barplot(data=state_segmentation, x='State', y='Number of Orders',palette=custom_palette)
```

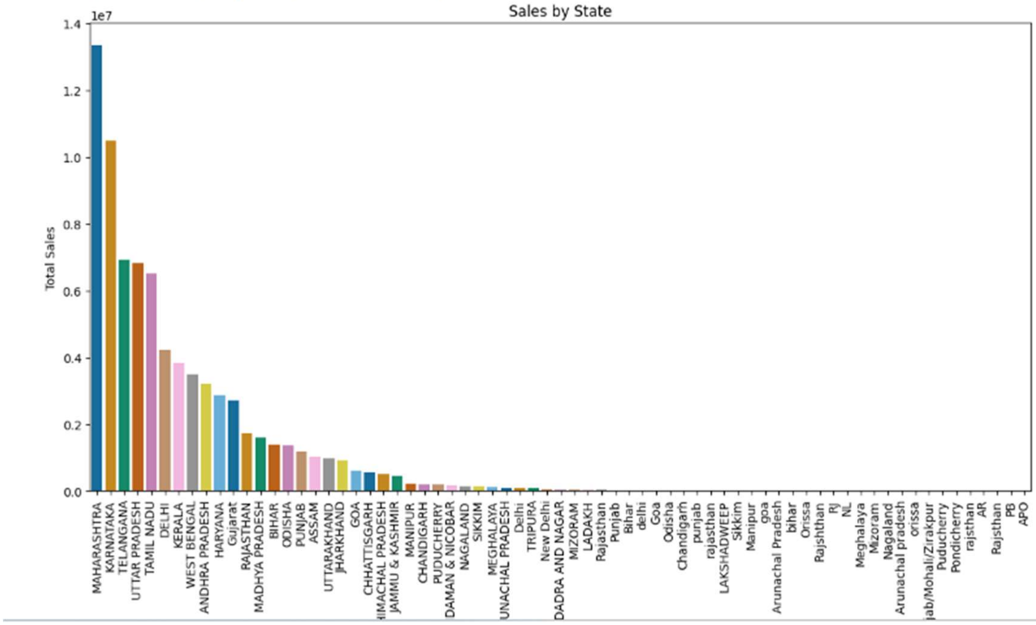


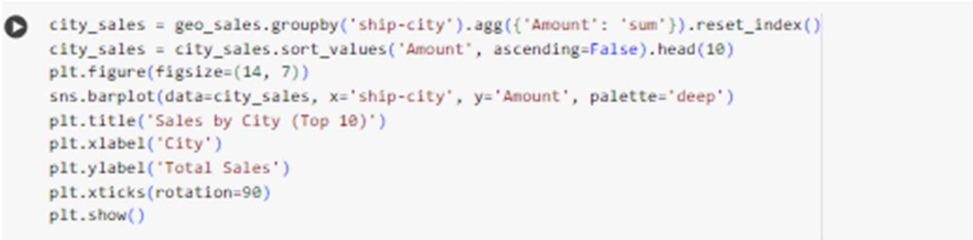
```
geo_sales = df.groupby(['ship-state', 'ship-city']).agg({'Amount': 'sum'}).reset_index()
```



```
state_sales = geo_sales.groupby('ship-state').agg({'Amount': 'sum'}).reset_index()
state_sales = state_sales.sort_values('Amount', ascending=False)
plt.figure(figsize=(14, 7))
sns.barplot(data=state_sales, x='ship-state', y='Amount', palette='colorblind')
plt.title('Sales by State')
plt.xlabel('State')
plt.ylabel('Total Sales')
plt.xticks(rotation=90)
plt.show()
```

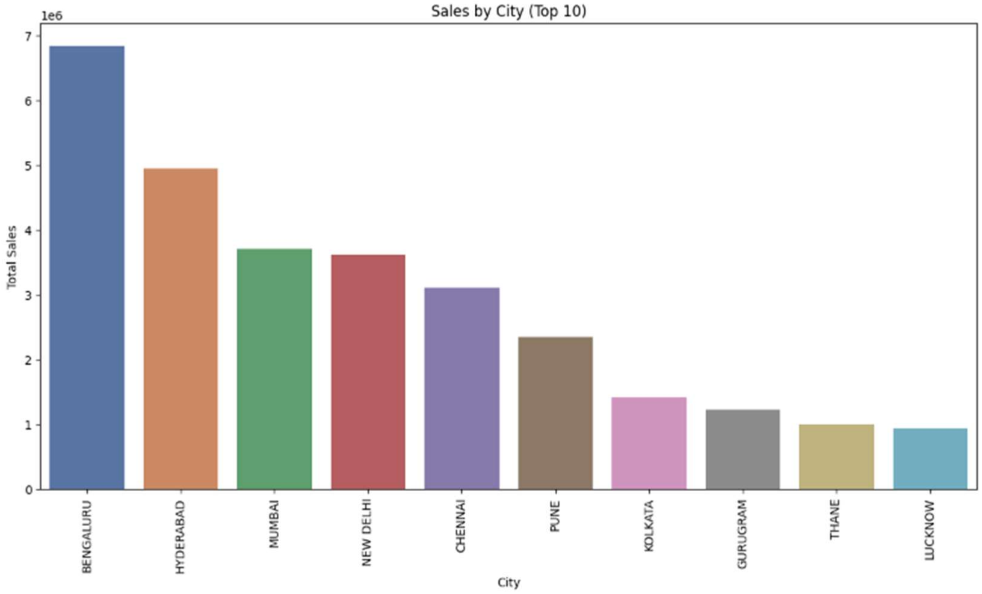
```
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(data=state_sales, x='ship-state', y='Amount', palette='colorblind')
```





```python
city_sales = geo_sales.groupby('ship-city').agg({'Amount': 'sum'}).reset_index()
city_sales = city_sales.sort_values('Amount', ascending=False).head(10)
plt.figure(figsize=(14, 7))
sns.barplot(data=city_sales, x='ship-city', y='Amount', palette='deep')
plt.title('Sales by City (Top 10)')
plt.xlabel('City')
plt.ylabel('Total Sales')
plt.xticks(rotation=90)
plt.show()
```

```
<ipython-input-39-ea183934f2b2>:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(data=city_sales, x='ship-city', y='Amount', palette='deep')
```


Sales by City (Top 10)
1e6
Total Sales
BENGALURU
HYDERABAD
MUMBAI
NEW DELHI
CHENNAI
PUNE
KOLKATA
GURUGRAM
THANE
LUCKNOW
City