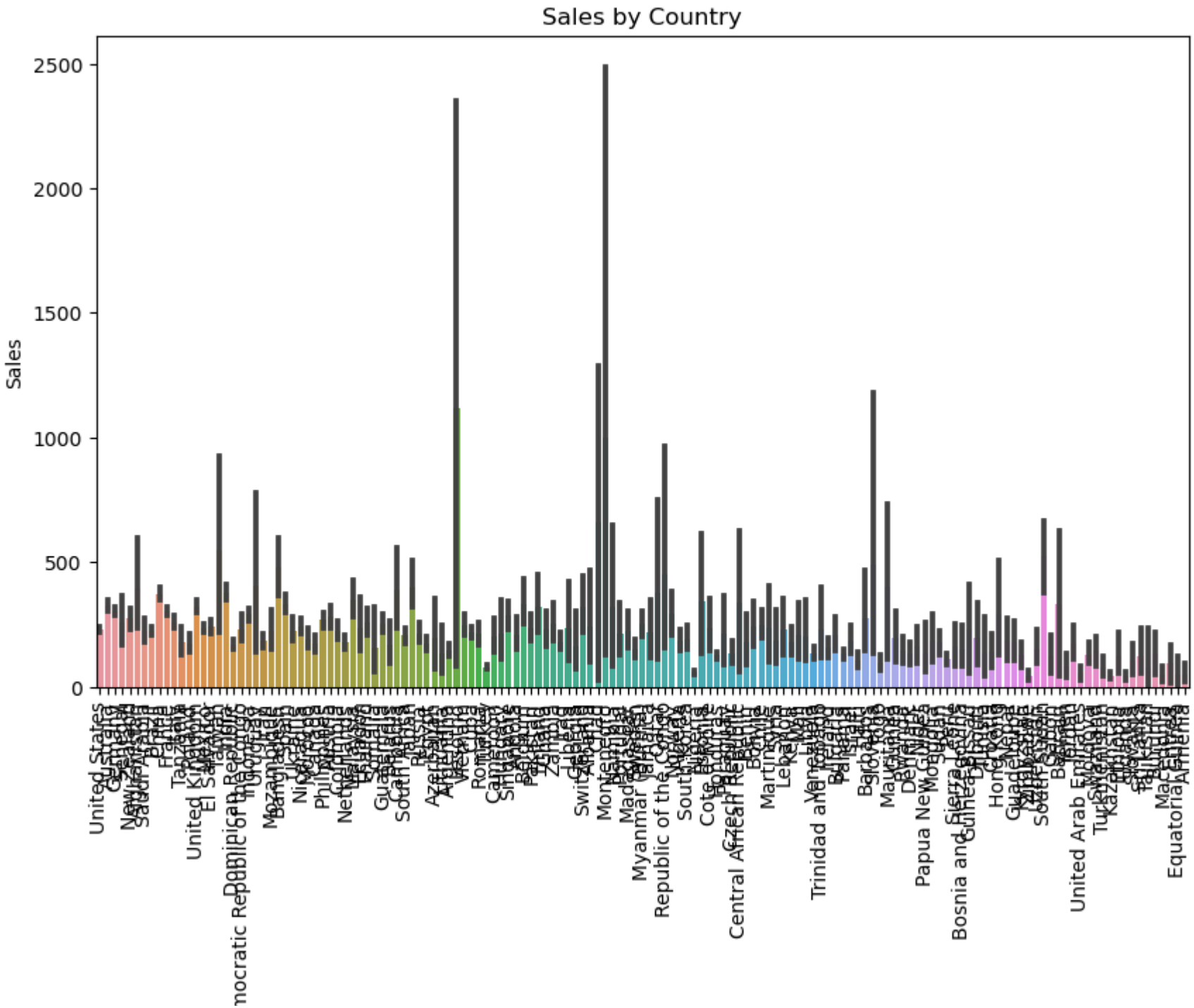
In [1]:
```
import pandas as pd

# Load the dataset
url = "https://docs.google.com/spreadsheets/d/1Ez8RhO1rE2QikfdddOZ19eEZhN4r6eW7WHb9qB05T7E/export?format=csv"
df = pd.read_csv(url)

# Display basic information
df.info()

# Check for missing values
missing_data = df.isnull().sum()

# Drop duplicates if any
df.drop_duplicates(inplace=True)

# Handle missing values (either drop or fill, depending on the importance of the column)
df.fillna(method='ffill', inplace=True)  # This fills missing data forward
```

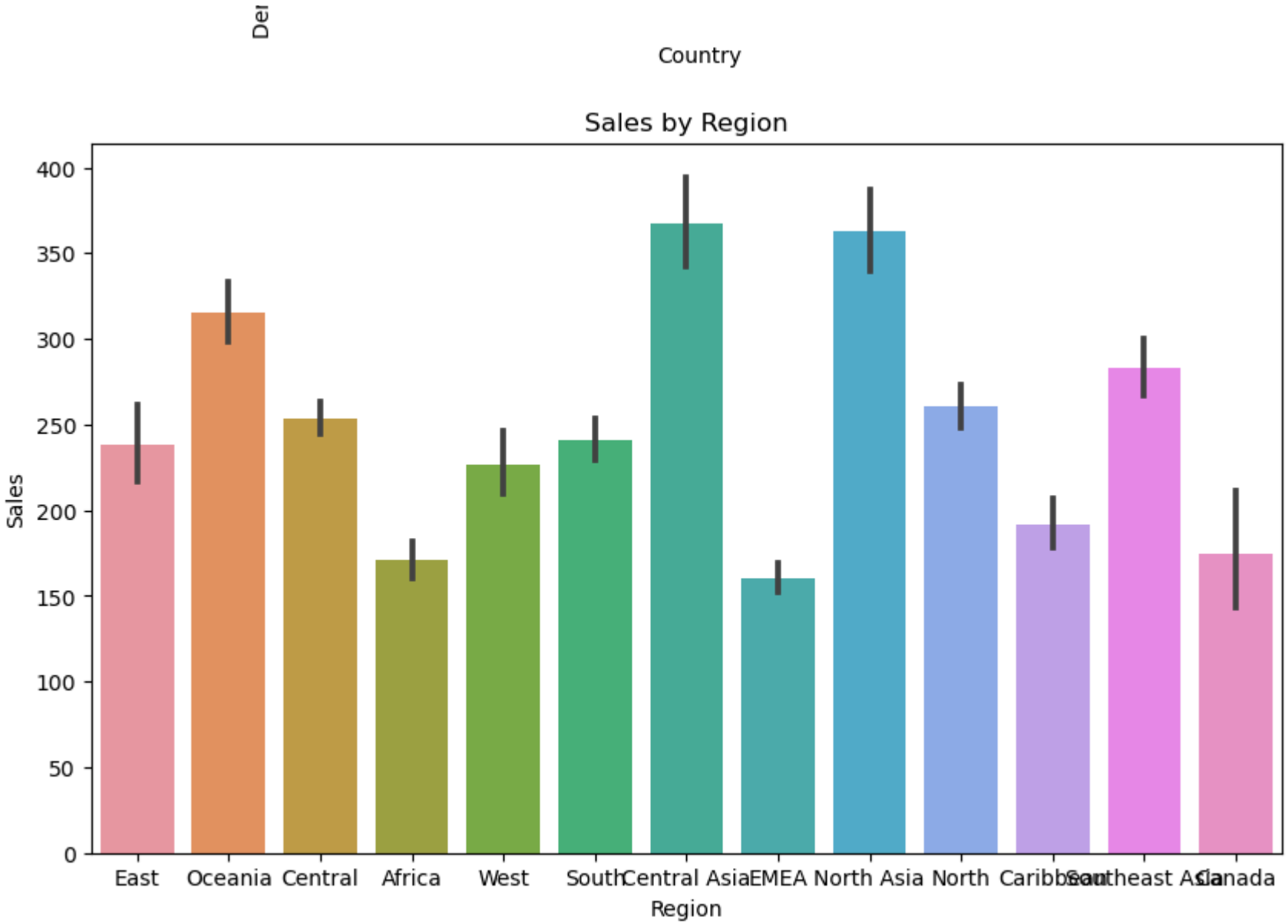
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51290 entries, 0 to 51289
Data columns (total 24 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Row ID          51290 non-null  int64  
 1   Order ID        51290 non-null  object 
 2   Order Date      51290 non-null  object 
 3   Ship Date       51290 non-null  object 
 4   Ship Mode       51290 non-null  object 
 5   Customer ID     51290 non-null  object 
 6   Customer Name   51290 non-null  object 
 7   Segment         51290 non-null  object 
 8   City            51290 non-null  object 
 9   State           51290 non-null  object 
 10  Country         51290 non-null  object 
 11  Postal Code     9994 non-null   float64
 12  Market          51290 non-null  object 
 13  Region          51290 non-null  object 
 14  Product ID      51290 non-null  object 
 15  Category        51290 non-null  object 
 16  Sub-Category    51290 non-null  object 
 17  Product Name    51290 non-null  object 
 18  Sales           51290 non-null  float64
 19  Quantity        51290 non-null  int64  
 20  Discount        51290 non-null  float64
 21  Profit          51290 non-null  float64
 22  Shipping Cost   51290 non-null  float64
 23  Order Priority  51290 non-null  object 
dtypes: float64(5), int64(2), object(17)
memory usage: 9.4+ MB
```

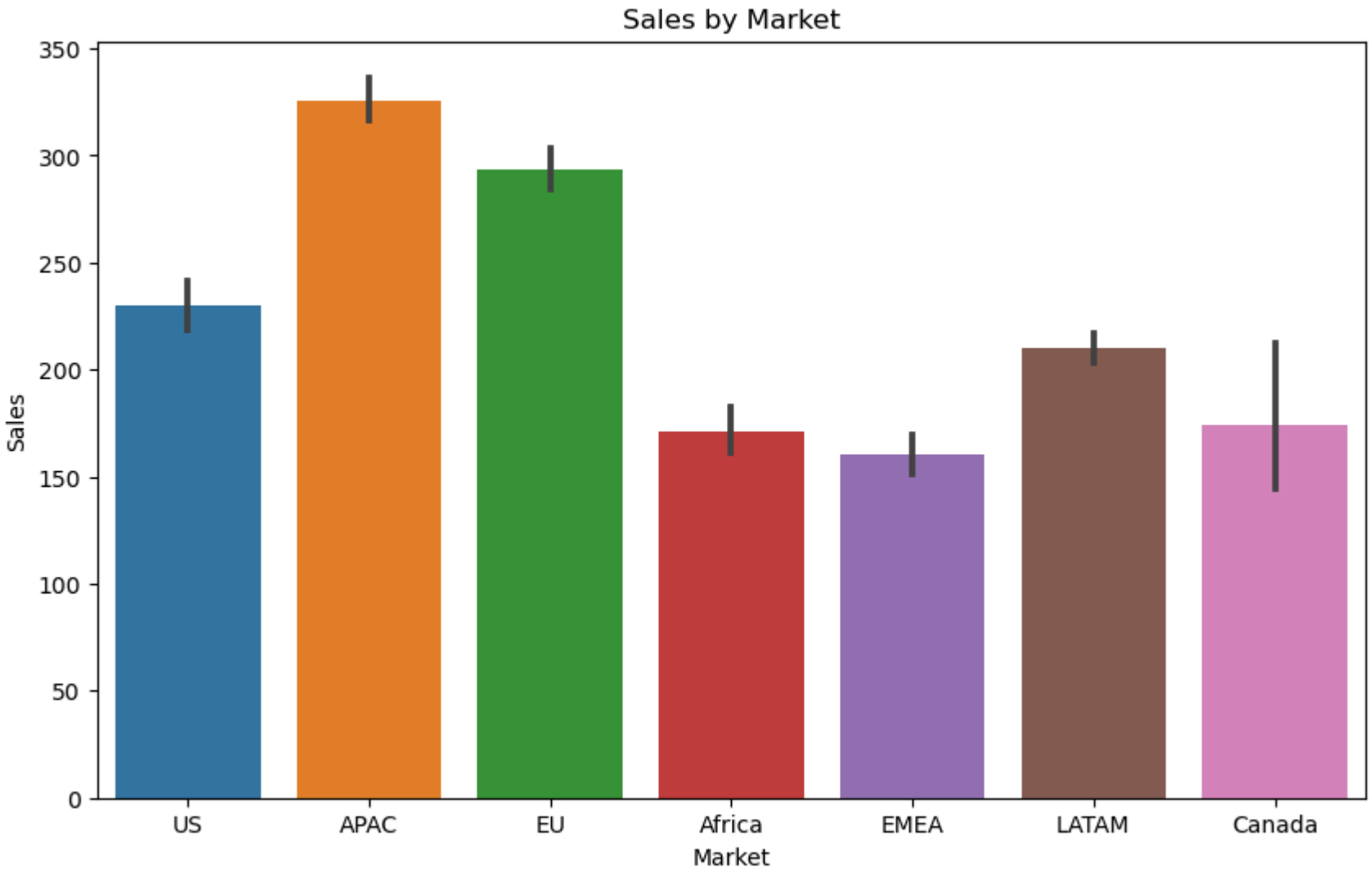
In [2]:

```
import matplotlib.pyplot as plt
import seaborn as sns

# Sales by Country
plt.figure(figsize=(10,6))
sns.barplot(x='Country', y='Sales', data=df)
plt.xticks(rotation=90)
plt.title('Sales by Country')
plt.show()

# Sales by Region
plt.figure(figsize=(10,6))
```

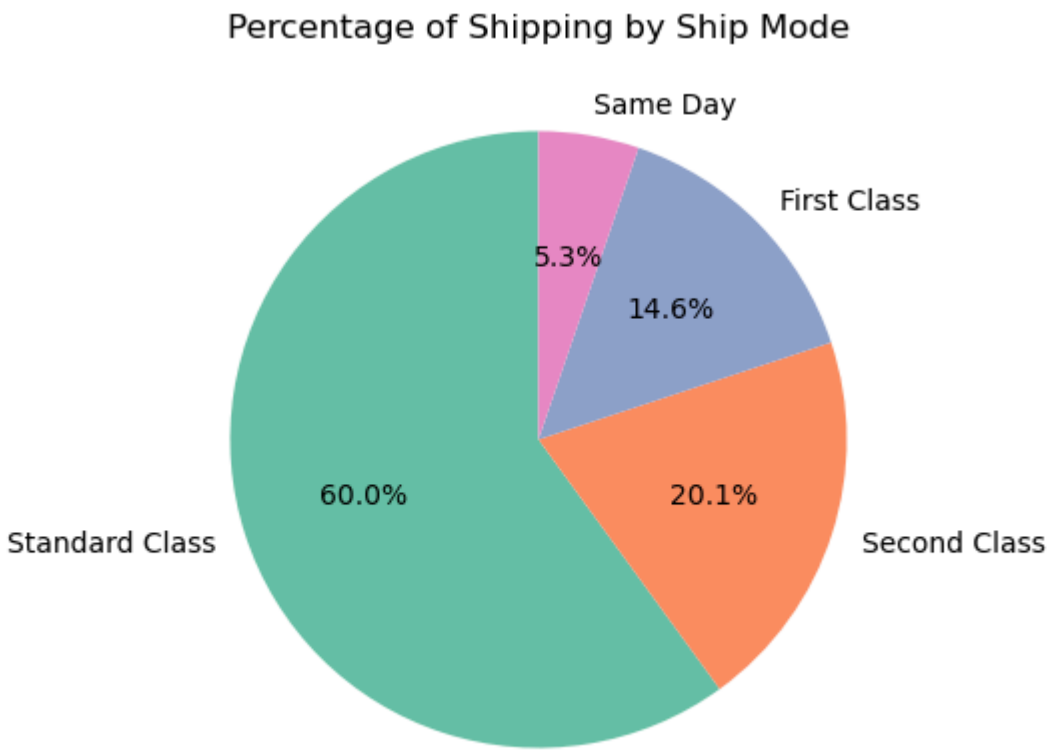
```
sns.barplot(x='Region', y='Sales', data=df)
plt.title('Sales by Region')
plt.show()

# Sales by Market
plt.figure(figsize=(10,6))
sns.barplot(x='Market', y='Sales', data=df)
plt.title('Sales by Market')
plt.show()
```


Sales by Country
Sales
2500
2000
1500
1000
500
0

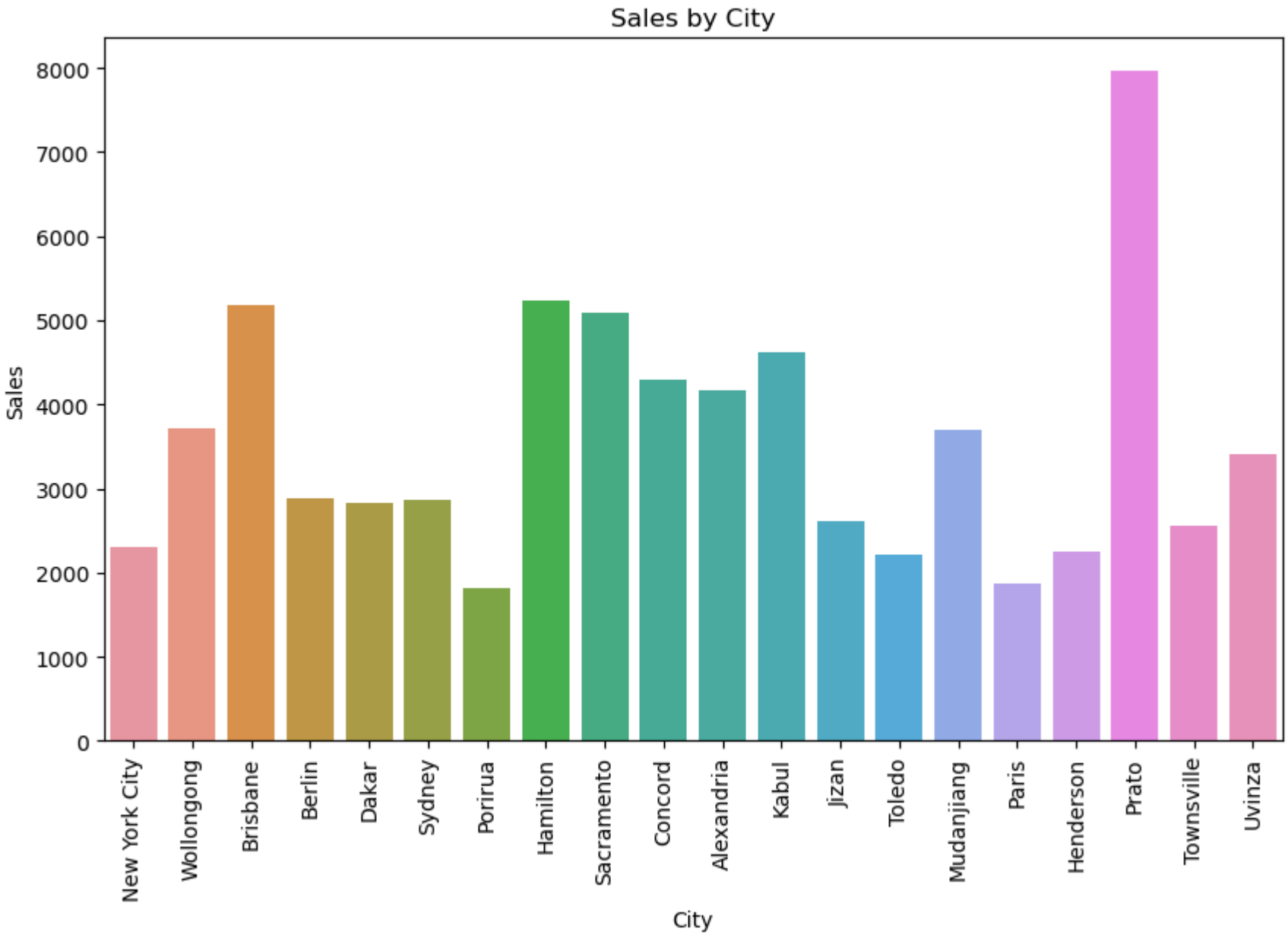

Country
Sales by Region
Sales
400
350
300
250
200
150
100
50
0
East
Oceania
Central
Africa
West
South
Central Asia
EMEA
North Asia
North
Caribbean
Southeast Asia
Canada
Region

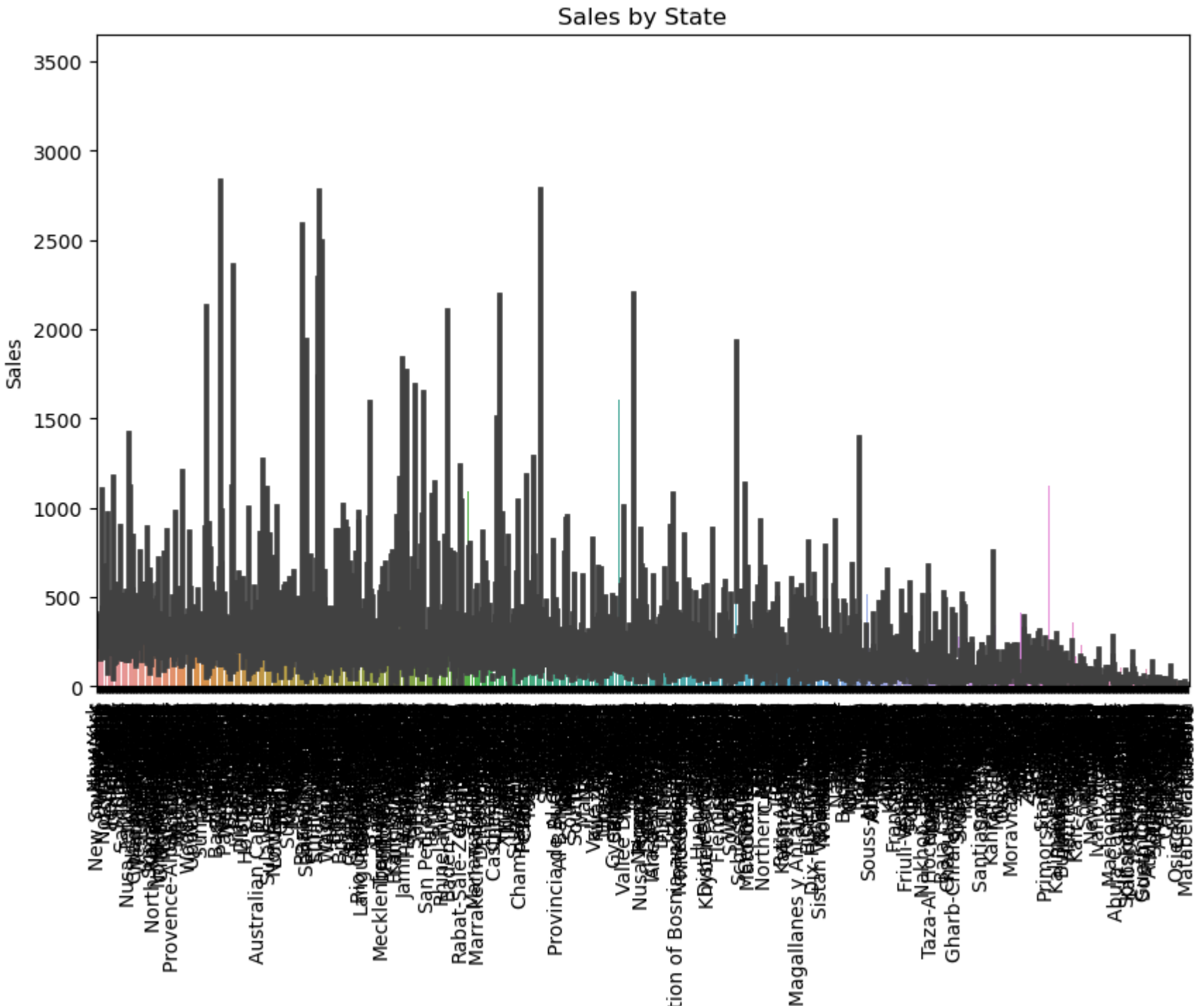


In [3]:
```
# Plot percentage of shipping by ship mode
ship_mode_percentage = df['Ship Mode'].value_counts(normalize=True) * 100

plt.figure(figsize=(8,5))
ship_mode_percentage.plot(kind='pie', autopct='%1.1f%%', startangle=90, colors=sns.color_palette('Set2'))
plt.title('Percentage of Shipping by Ship Mode')
plt.ylabel('')
plt.show()
```



In [4]:
```
# Sales by City
plt.figure(figsize=(10,6))
sns.barplot(x='City', y='Sales', data=df.head(20))  # Showing top 20 cities
plt.xticks(rotation=90)
plt.title('Sales by City')
plt.show()

# Sales by State
plt.figure(figsize=(10,6))
sns.barplot(x='State', y='Sales', data=df)
plt.xticks(rotation=90)
plt.title('Sales by State')
plt.show()

# You can repeat this for Region and Market similarly as done for State and City
```


Sales by City
Sales
8000
7000
6000
5000
4000
3000
2000
1000
0
New York City
Wollongong
Brisbane
Berlin
Dakar
Sydney
Porirua
Hamilton
Sacramento
Concord
Alexandria
Kabul
Jizan
Toledo
Mudanjiang
Paris
Henderson
Prato
Townsville
Uvinza
City


Sales by State
Sales
3500
3000
2500
2000
1500
1000
500
0

In [6]:
```
# Example table for Sales by Region
sales_by_region = df.groupby('Region')['Sales'].sum().reset_index()
print(sales_by_region)
```

```
            Region        Sales
0           Africa  7.837732e+05
1           Canada  6.692817e+04
2        Caribbean  3.242809e+05
3          Central  2.822303e+06
4     Central Asia  7.528266e+05
5             EMEA  8.061613e+05
6             East  6.787812e+05
7            North  1.248166e+06
8       North Asia  8.483098e+05
9          Oceania  1.100185e+06
10           South  1.600907e+06
11  Southeast Asia  8.844232e+05
12            West  7.254578e+05
```

In [ ]: