In [13]:
```python
import pandas as pd

# Load the sales data into a DataFrame
sales_data = pd.read_csv("C:/Users/SHOPINVERSE/Downloads/sales_data.csv")
```

In [ ]:
```python
#DATA EXPLORATION
```

In [19]:
```python
import pandas as pd
from tabulate import tabulate

# Load the sales data into a DataFrame
sales_data = pd.read_csv("C:/Users/SHOPINVERSE/Downloads/sales_data.csv")

# Display the first few rows of the DataFrame
print("First few rows of the DataFrame:")
print(tabulate(sales_data.head(), headers='keys', tablefmt='fancy_grid'))

# Get an overview of the dataset
print("Dataset Overview:")
print(tabulate(sales_data.info(), headers='keys', tablefmt='fancy_grid'))
print("Summary Statistics:")
print(tabulate(sales_data.describe(), headers='keys', tablefmt='fancy_grid'))
```

First few rows of the DataFrame:

	Transaction_Date	Brand	Product_name	Store_type	Country	State	City	Sales Revenue	Cost	Profit/Loss
0	January 1, 2018	Best Choice	Grape Fruit Roll	Gourmet Supermarket	USA	CA	Beverly Hills	11	5	6
1	January 1, 2018	Fort West	Fudge Cookies	Gourmet Supermarket	USA	CA	Beverly Hills	7	3	4
2	January 1, 2018	Tell Tale	Canned Peanuts	Gourmet Supermarket	USA	CA	Beverly Hills	14	7	7
3	January 1, 2018	Tell Tale	Prepared Salad	Gourmet Supermarket	USA	CA	Beverly Hills	19	9	10
4	January 1, 2018	High Top	Dried Mushrooms	Gourmet Supermarket	USA	CA	Beverly Hills	15	6	9

```
Dataset Overview:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 80000 entries, 0 to 79999
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Transaction_Date  80000 non-null  object
 1   Brand             80000 non-null  object
 2   Product_name      80000 non-null  object
 3   Store_type        80000 non-null  object
 4   Country           80000 non-null  object
 5   State             80000 non-null  object
 6   City              80000 non-null  object
 7   Sales Revenue     80000 non-null  int64
 8   Cost              80000 non-null  int64
 9   Profit/Loss       80000 non-null  int64
dtypes: int64(3), object(7)
memory usage: 6.1+ MB
```

Summary Statistics:

	Sales Revenue	Cost	Profit/Loss
count	80000	80000	80000
mean	6.85292	2.7667	4.07182
std	3.41627	1.47008	2.08388
min	1	0	0
25%	4	2	3
50%	7	3	4
75%	9	4	5
max	24	11	15

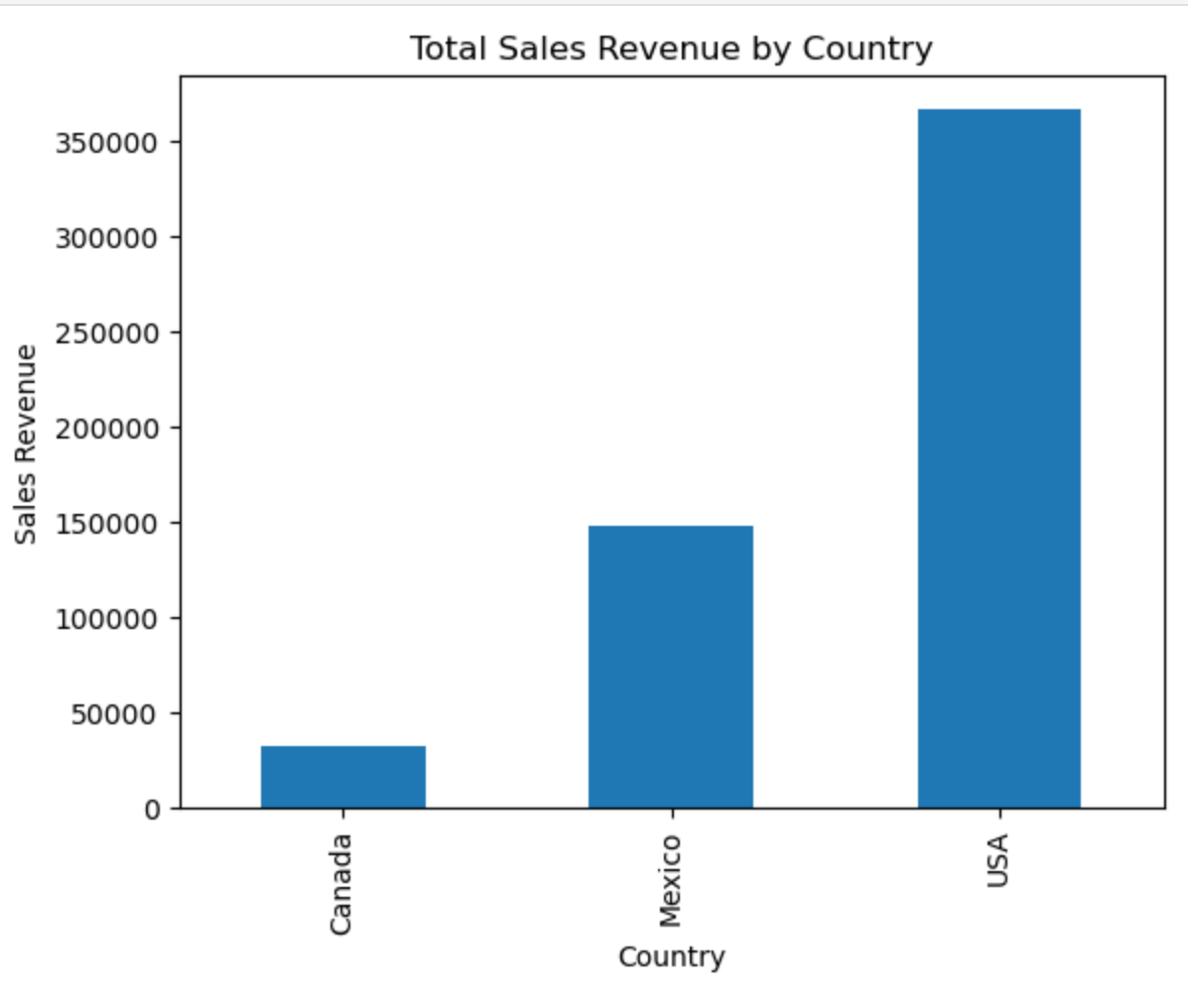
In [ ]:
```python
#DATA ANALYSIS TASK
```

In [15]:
```python
# Calculate total sales revenue
total_revenue = sales_data['Sales Revenue'].sum()
print("Total sales revenue:", total_revenue)

# Calculate total cost
total_cost = sales_data['Cost'].sum()
print("Total cost:", total_cost)

# Calculate total profit
total_profit = sales_data['Profit/Loss'].sum()
print("Total profit:", total_profit)

# Group data by a specific column and perform aggregations
grouped_data = sales_data.groupby('Country').agg({'Sales Revenue': 'sum', 'Cost': 'sum', 'Profit/Loss': 'sum'})
print("Grouped data:\n", grouped_data)
```

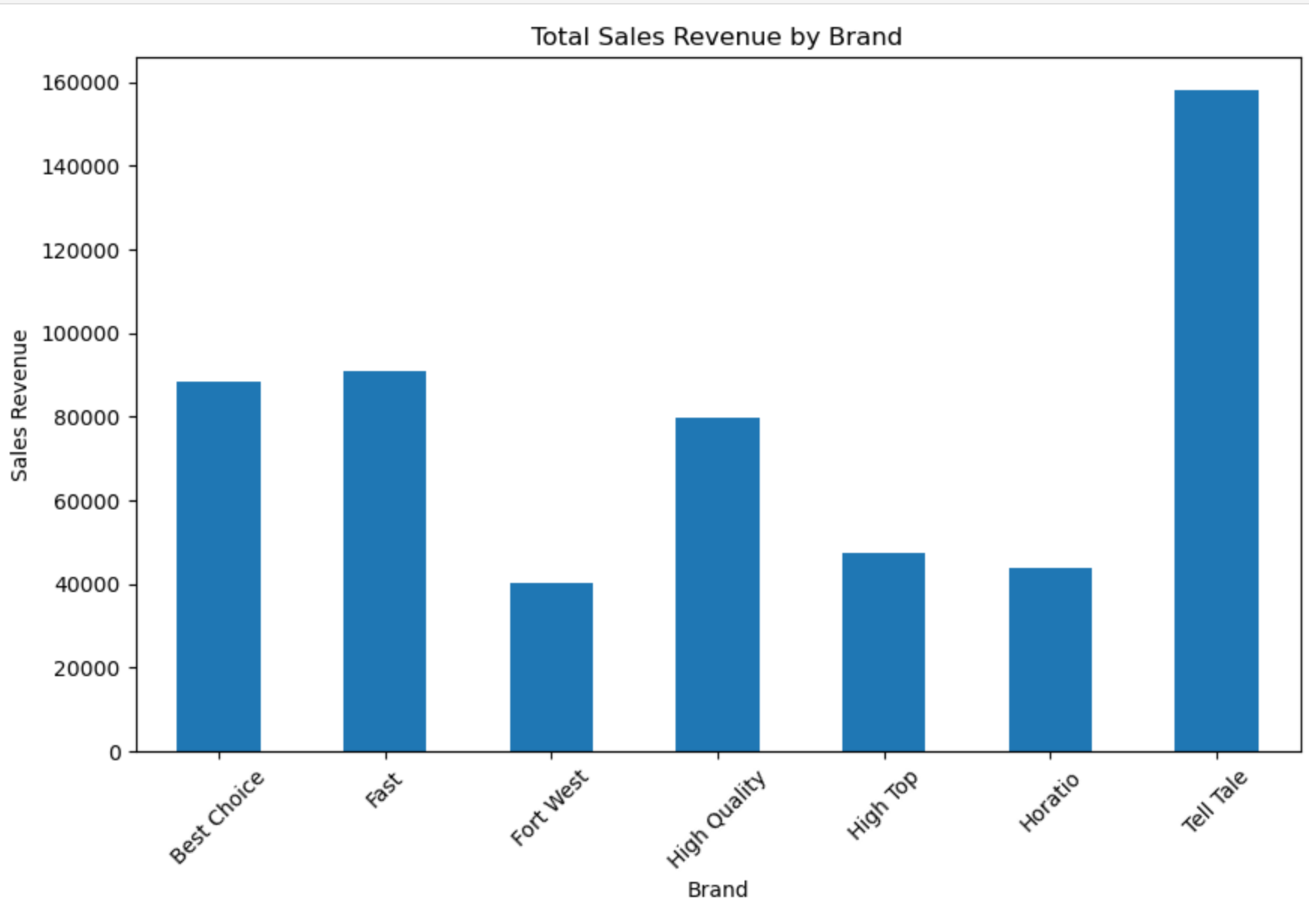
```
Total sales revenue: 548234
Total cost: 221336
Total profit: 325746
Grouped data:
          Sales Revenue    Cost  Profit/Loss
Country
Canada            33197   13371        19823
Mexico           148442   60053        88270
USA              366595  147912       217653
```

In [ ]:
```python
#VISULIZATION USING MATPLOTLIB
```

In [16]:
```python
import matplotlib.pyplot as plt

# Plot total sales revenue by country
sales_by_country = sales_data.groupby('Country')['Sales Revenue'].sum()
sales_by_country.plot(kind='bar', xlabel='Country', ylabel='Sales Revenue', title='Total Sales Revenue by Country')
plt.show()
```



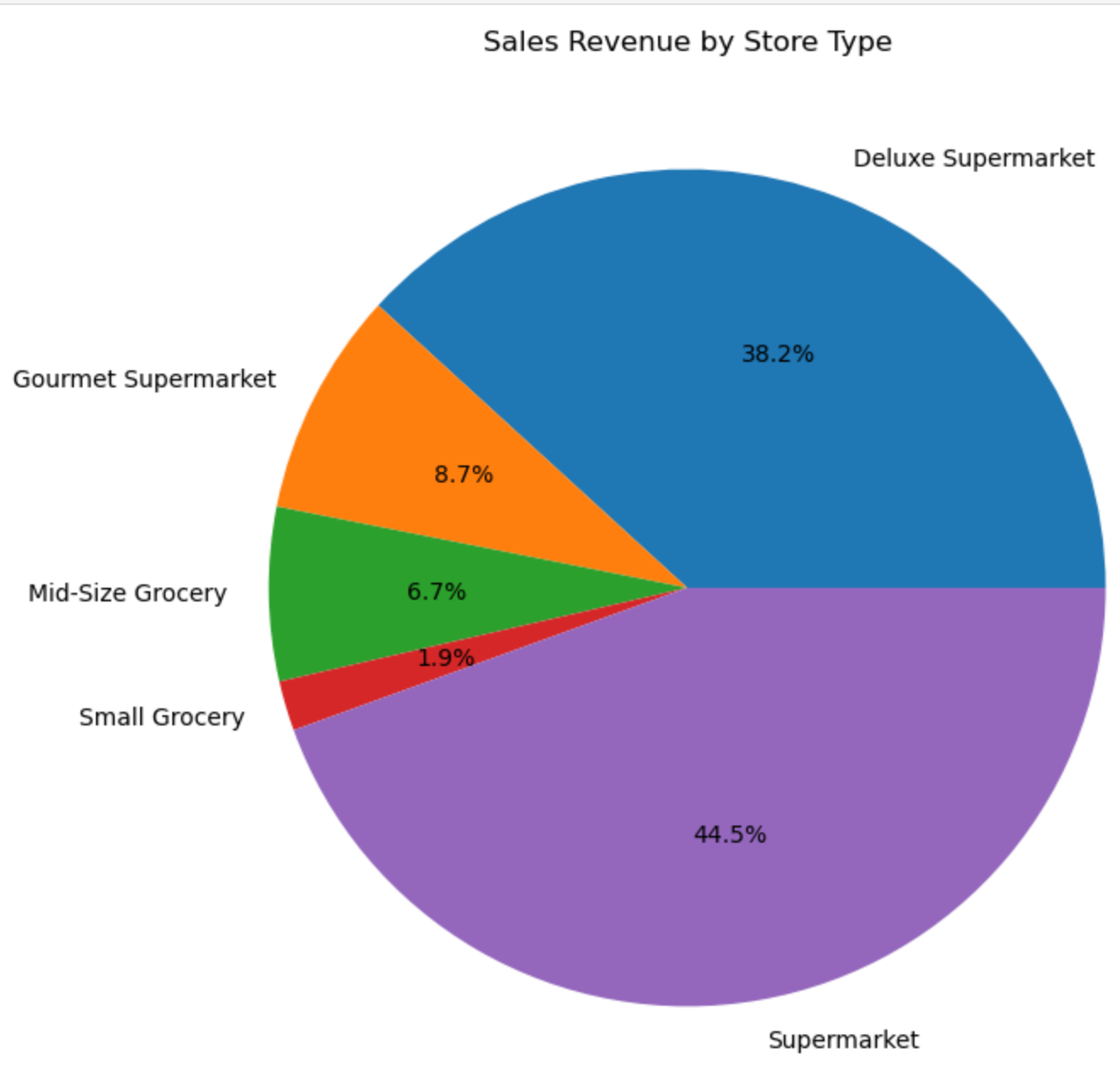
In [22]:
```python
import matplotlib.pyplot as plt

# Calculate total sales revenue by brand
brand_sales = sales_data.groupby('Brand')['Sales Revenue'].sum()

# Create a bar plot
plt.figure(figsize=(10, 6))
brand_sales.plot(kind='bar')
plt.title('Total Sales Revenue by Brand')
plt.xlabel('Brand')
plt.ylabel('Sales Revenue')
plt.xticks(rotation=45)
plt.show()
```



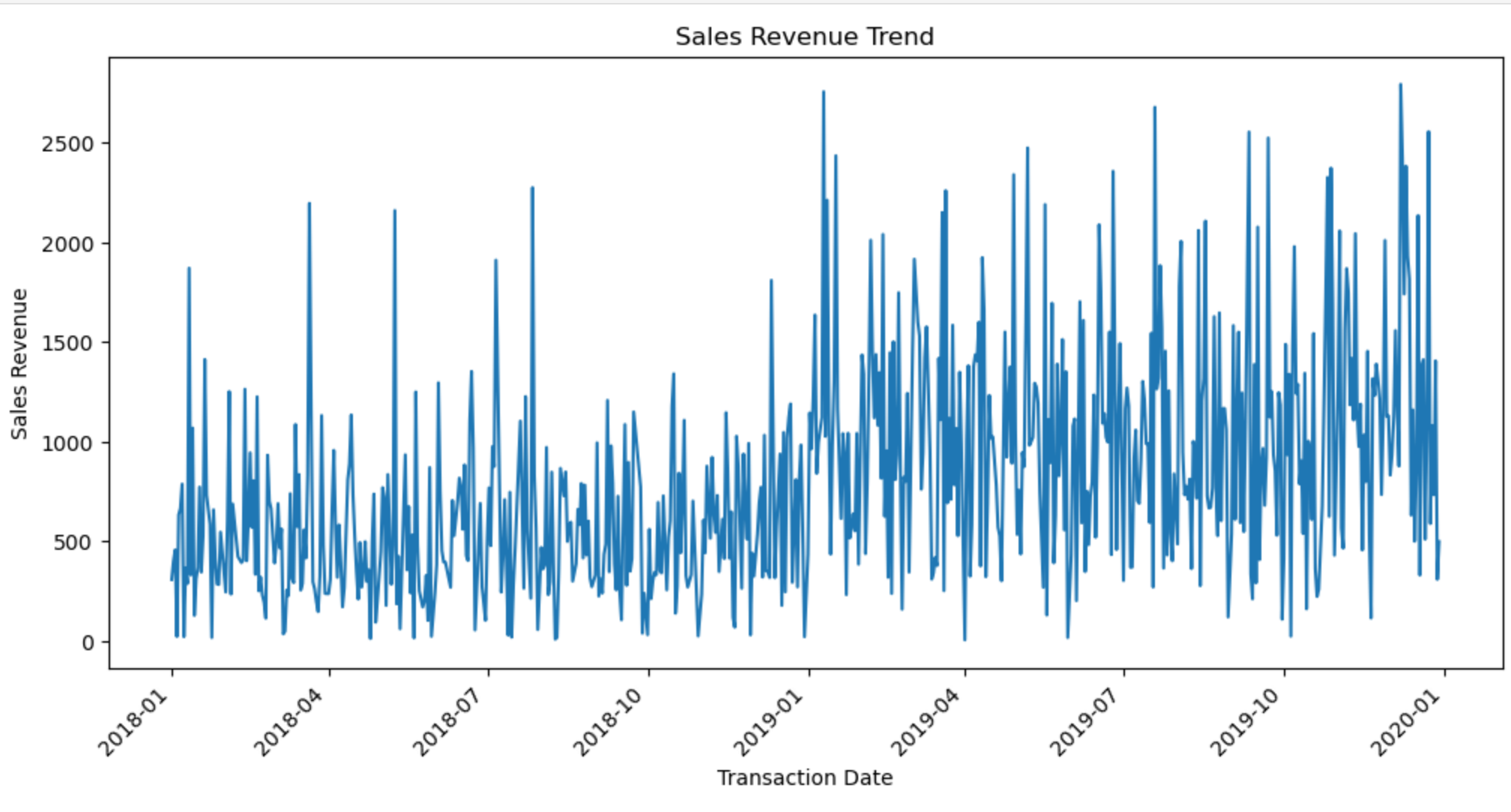
In [24]:
```python
print(sales_data.columns)
```

```
Index(['Transaction_Date', 'Brand', 'Product_name', 'Store_type', 'Country',
       'State', 'City', 'Sales Revenue', 'Cost', 'Profit/Loss'],
      dtype='object')
```

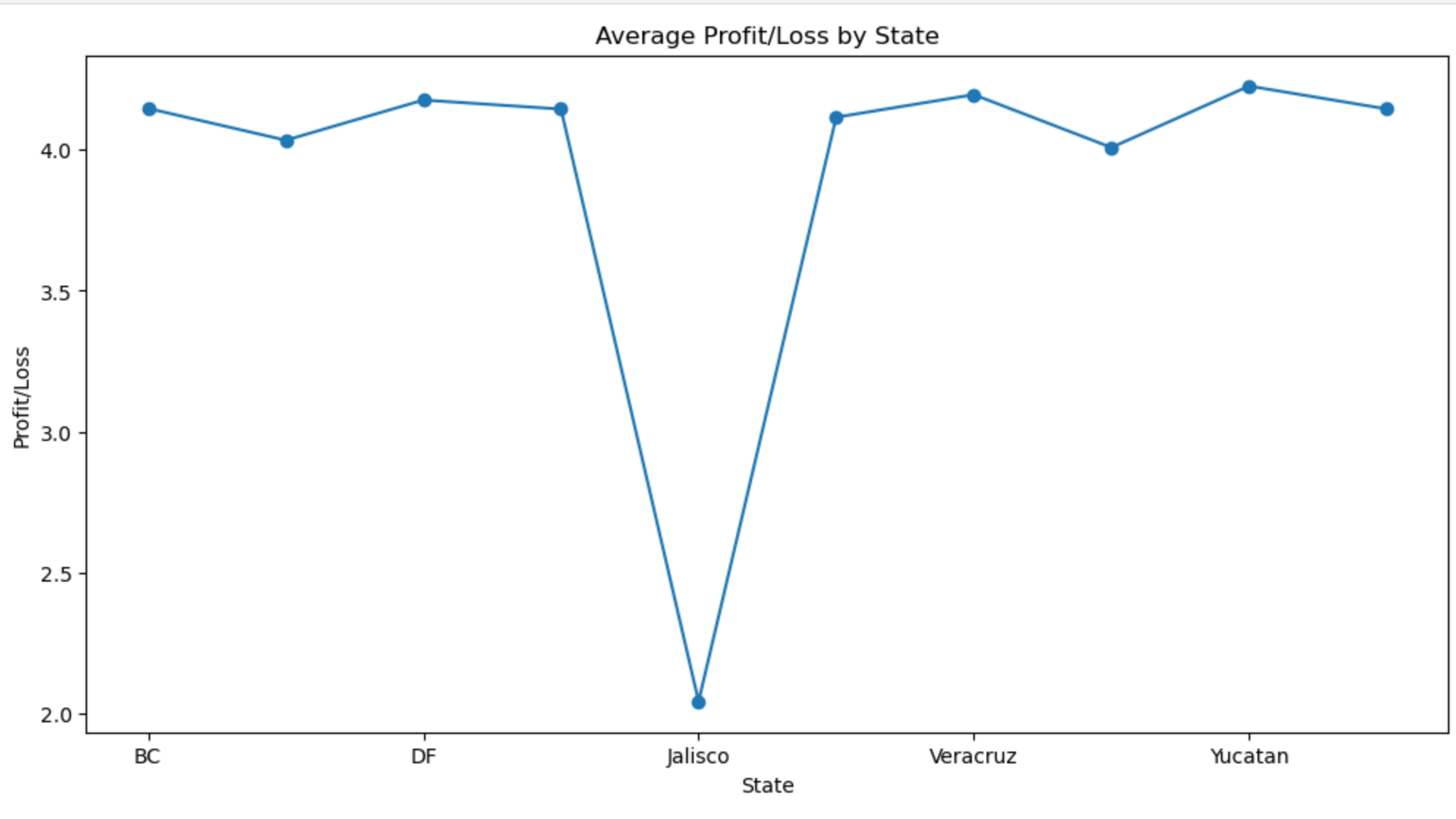
In [25]:
```python
# Calculate the total sales revenue by store type
store_type_sales = sales_data.groupby('Store_type')['Sales Revenue'].sum()

# Create a pie chart
plt.figure(figsize=(8, 8))
plt.pie(store_type_sales, labels=store_type_sales.index, autopct='%1.1f%%')
plt.title('Sales Revenue by Store Type')
plt.show()
```



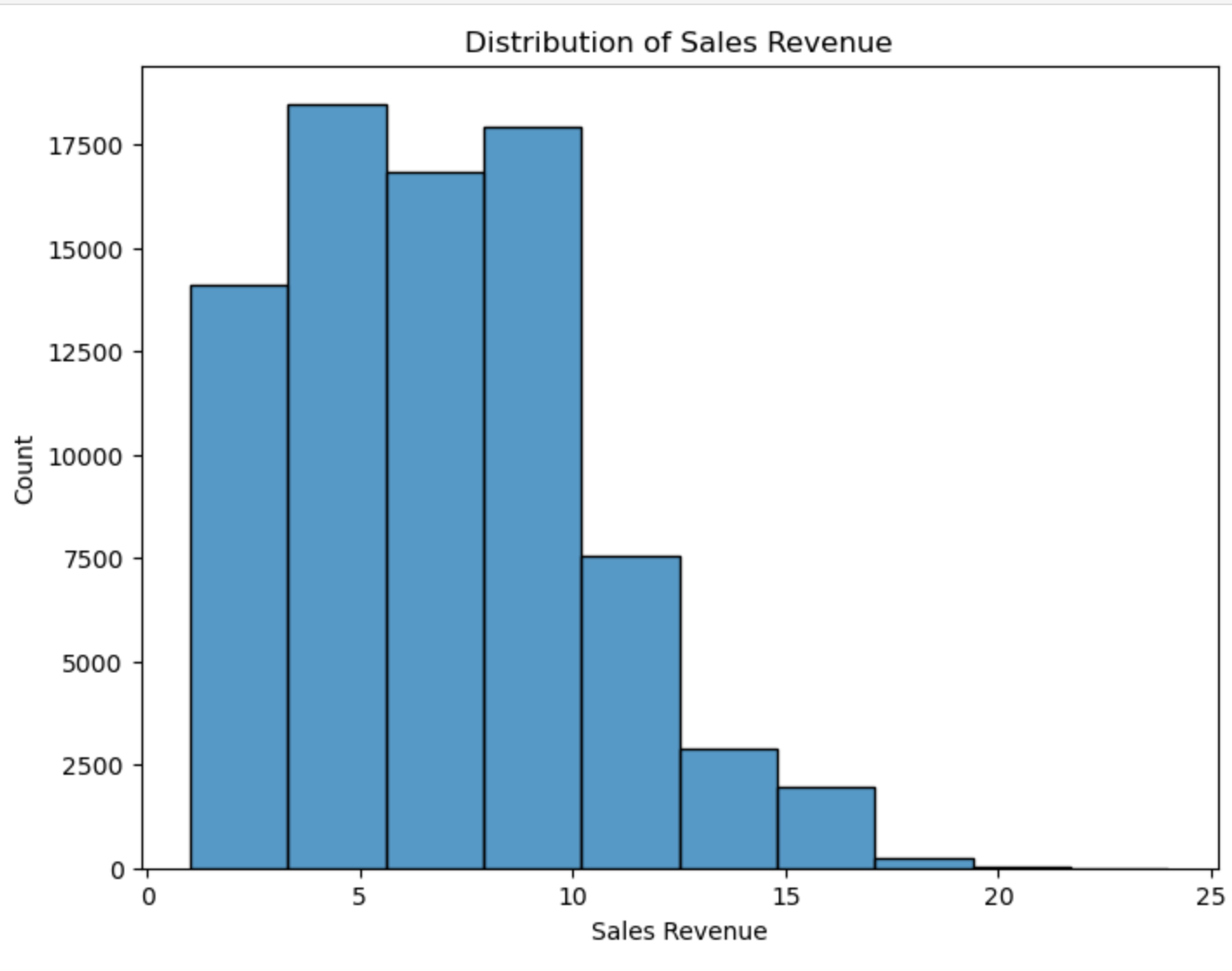
In [26]:
```python
# Convert the Transaction_Date column to datetime format
sales_data['Transaction_Date'] = pd.to_datetime(sales_data['Transaction_Date'])

# Calculate the total sales revenue by transaction date
date_sales = sales_data.groupby('Transaction_Date')['Sales Revenue'].sum()

# Create a line plot
plt.figure(figsize=(12, 6))
date_sales.plot(kind='line')
plt.title('Sales Revenue Trend')
plt.xlabel('Transaction Date')
plt.ylabel('Sales Revenue')
plt.xticks(rotation=45)
plt.show()
```



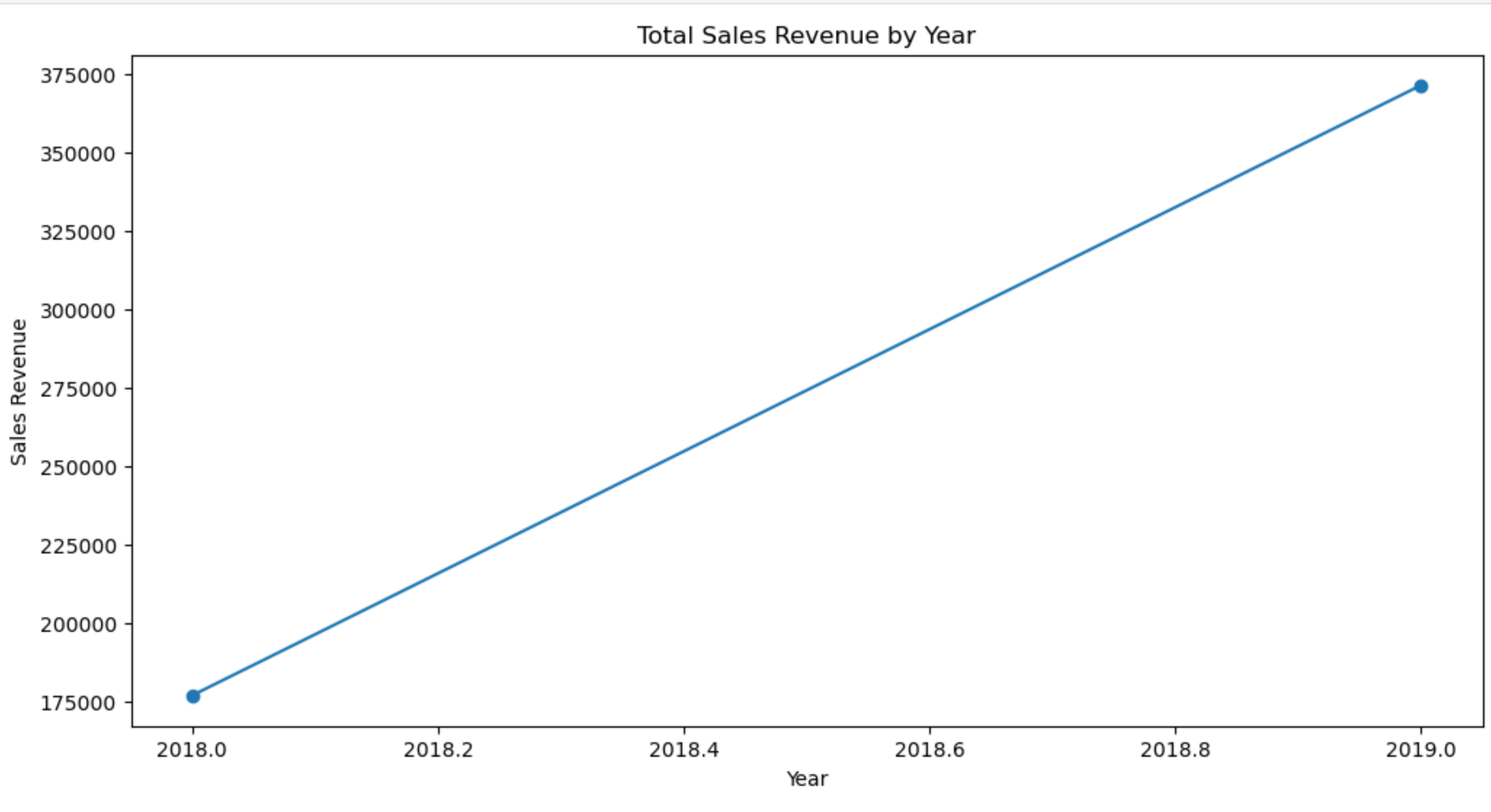
In [27]:
```python
state_profit = sales_data.groupby('State')['Profit/Loss'].mean()

# Create a line plot
plt.figure(figsize=(12, 6))
state_profit.plot(marker='o')
plt.title('Average Profit/Loss by State')
plt.xlabel('State')
plt.ylabel('Profit/Loss')
plt.show()
```



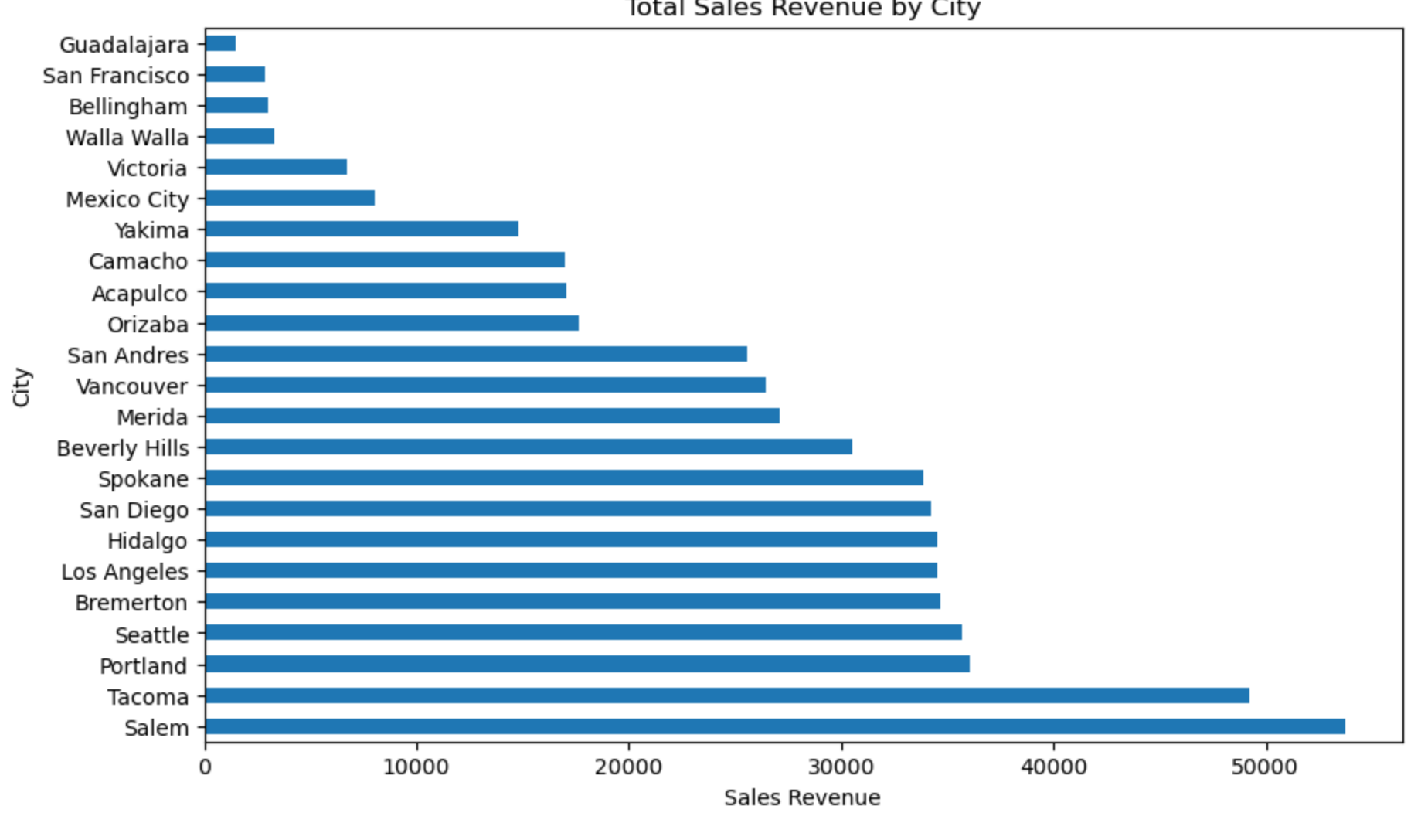
In [28]:
```python
plt.figure(figsize=(8, 6))
sns.histplot(sales_data['Sales Revenue'], bins=10)
plt.title('Distribution of Sales Revenue')
plt.xlabel('Sales Revenue')
plt.ylabel('Count')
plt.show()
```



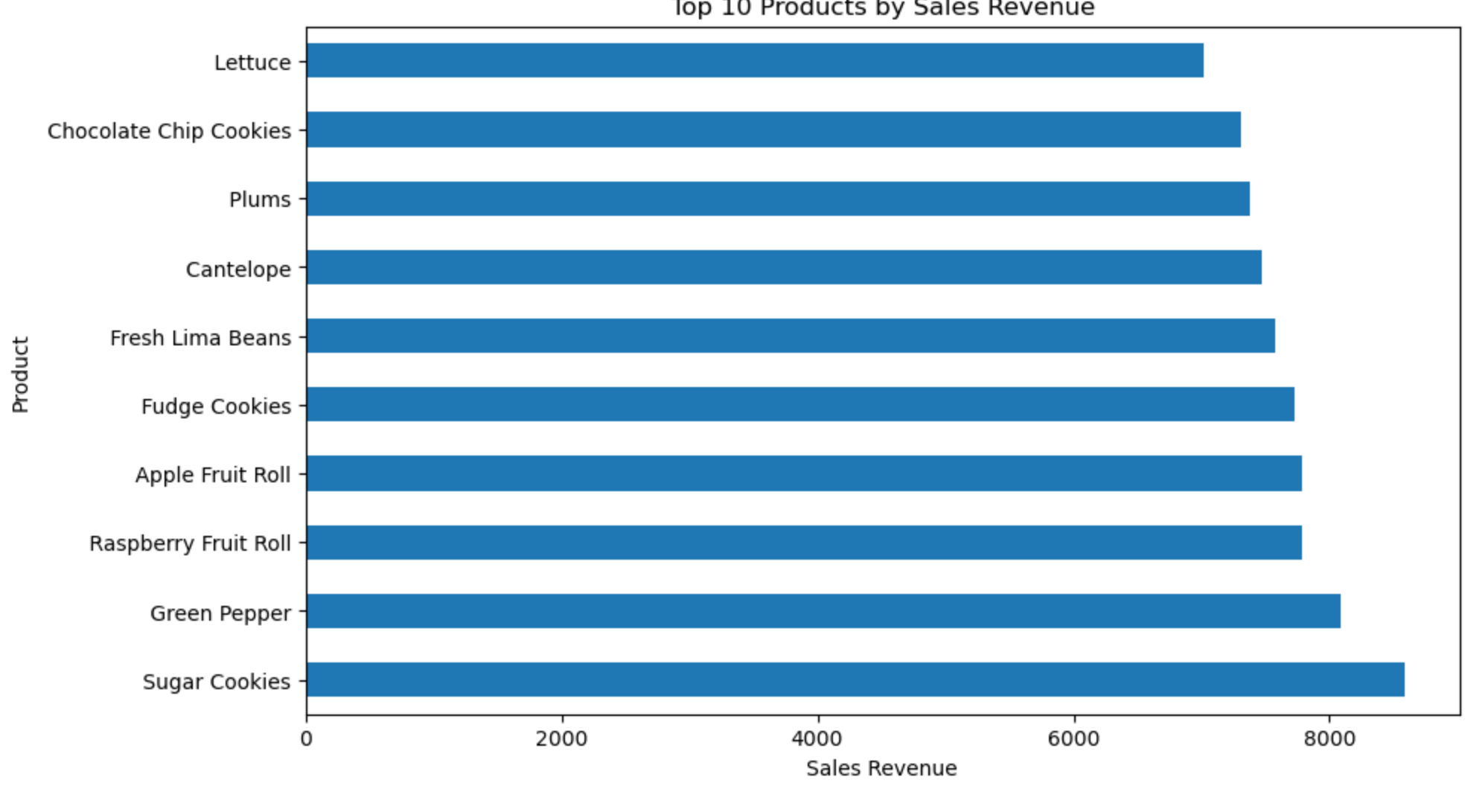
In [29]:
```python
# Convert the 'Transaction_Date' column to datetime format
sales_data['Transaction_Date'] = pd.to_datetime(sales_data['Transaction_Date'])

# Extract the year from the 'Transaction_Date' column
sales_data['Year'] = sales_data['Transaction_Date'].dt.year

yearly_sales = sales_data.groupby('Year')['Sales Revenue'].sum()

# Create a line plot
plt.figure(figsize=(12, 6))
yearly_sales.plot(marker='o')
plt.title('Total Sales Revenue by Year')
plt.xlabel('Year')
plt.ylabel('Sales Revenue')
plt.show()
```



In [30]:
```python
city_sales = sales_data.groupby('City')['Sales Revenue'].sum()

# Create a horizontal bar plot
plt.figure(figsize=(10, 6))
city_sales.sort_values(ascending=False).plot(kind='barh')
plt.title('Total Sales Revenue by City')
plt.xlabel('Sales Revenue')
plt.ylabel('City')
plt.show()
```



In [32]:
```python
N = 10  # Number of top products to show

# Calculate the total sales revenue by product name
product_sales = sales_data.groupby('Product_name')['Sales Revenue'].sum()

# Select the top N products with the highest sales revenue
top_products = product_sales.nlargest(N)

# Create a horizontal bar plot
plt.figure(figsize=(10, 6))
top_products.sort_values(ascending=False).plot(kind='barh')
plt.title(f'Top {N} Products by Sales Revenue')
plt.xlabel('Sales Revenue')
plt.ylabel('Product')
plt.show()
```



In [ ]: