In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

In [4]:
```
data = pd.read_csv('E-commerce Website Logs.csv')
```

In [5]:
```
data
```

Out[5]:

	access_date	duration_(seconds)	Proto	IP	Src IP type	S
0	11/1/2016 9:58	2533	TCP	1.10.195.126	EXT_SERVER	80
1	11/1/2016 9:59	4034	TCP	1.1.217.211	OPENSTACK_NET	569
2	11/1/2016 9:59	1525	TCP	1.115.198.107	EXT_SERVER	80
3	11/1/2016 10:00	4572	TCP	1.121.152.143	OPENSTACK_NET	569
4	11/1/2016 10:00	3652	TCP	1.123.135.213	EXT_SERVER	80
...	...	...	...	...	...	
172833	12/31/2016 10:15	4372	TCP	94.197.121.229	EXT_SERVER	80
172834	12/31/2016 10:15	2167	TCP	129.110.241.72	EXT_SERVER	80
172835	12/31/2016 10:16	2725	TCP	185.119.252.121	EXT_SERVER	80
172836	12/31/2016 10:16	3728	TCP	185.145.107.23	EXT_SERVER	80
172837	12/31/2016 10:17	3420	TCP	202.69.12.251	EXT_SERVER	80

172838 rows × 14 columns

In [6]:
```
data.head()
```

Out[6]:

	access_date	duration_(seconds)	Proto	IP	Src IP type	Src Pt	By
0	11/1/2016 9:58	2533	TCP	1.10.195.126	EXT_SERVER	8082	20
1	11/1/2016 9:59	4034	TCP	1.1.217.211	OPENSTACK_NET	56978	20
2	11/1/2016 9:59	1525	TCP	1.115.198.107	EXT_SERVER	8082	90
3	11/1/2016 10:00	4572	TCP	1.121.152.143	OPENSTACK_NET	56979	100
4	11/1/2016 10:00	3652	TCP	1.123.135.213	EXT_SERVER	8082	270

In [7]:
```
data.info()
```

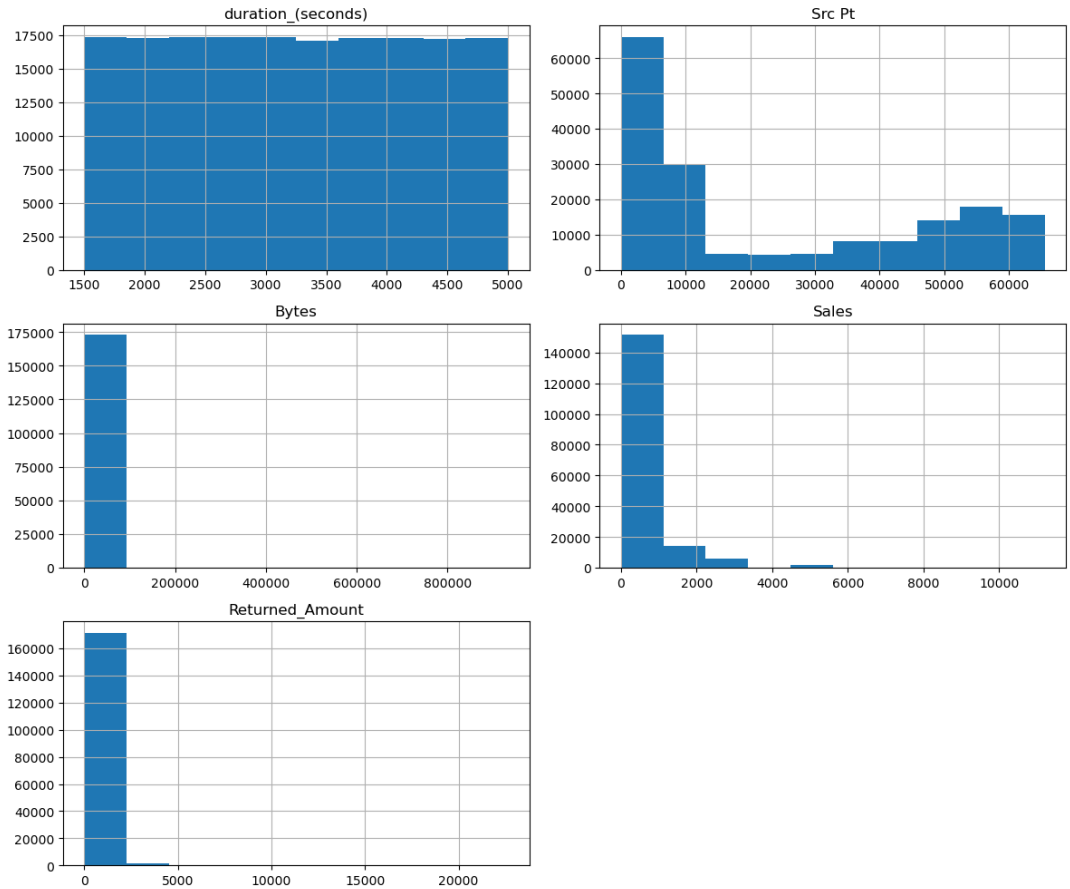
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 172838 entries, 0 to 172837
Data columns (total 14 columns):
 #   Column              Non-Null Count   Dtype  
---  ------              --------------   -----  
 0   access_date         172838 non-null  object 
 1   duration_(seconds)  172838 non-null  int64  
 2   Proto               172838 non-null  object 
 3   IP                  172838 non-null  object 
 4   Src IP type         172838 non-null  object 
 5   Src Pt              172838 non-null  int64  
 6   Bytes               172838 non-null  int64  
 7   Accessed_From       172838 non-null  object 
 8   country             172838 non-null  object 
 9   membership          172838 non-null  object 
 10  Languages           172838 non-null  object 
 11  Sales               172838 non-null  float64
 12  Returned            172838 non-null  object 
 13  Returned_Amount     172838 non-null  float64
dtypes: float64(2), int64(3), object(9)
memory usage: 18.5+ MB
```

In [8]:
```
data.describe()
```

Out[8]:

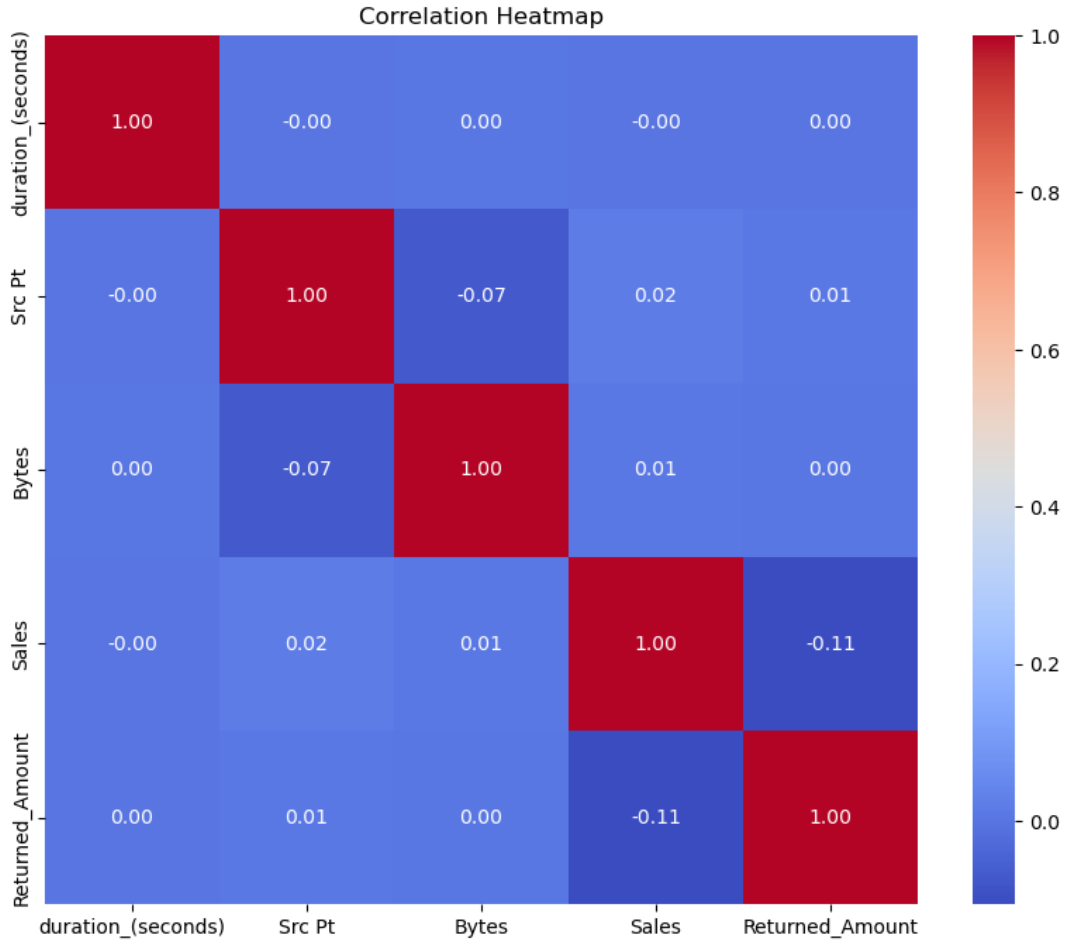
	duration_(seconds)	Src Pt	Bytes	Sales	Returned_Amo
count	172838.000000	172838.000000	172838.000000	172838.000000	172838.000
mean	3248.031827	22445.197526	1535.206858	411.346449	74.012
std	1010.872270	23810.373191	6349.555845	785.537868	364.446
min	1500.000000	0.000000	28.000000	0.000000	0.000
25%	2371.000000	23.000000	264.000000	5.230000	0.000
50%	3246.000000	8000.000000	589.000000	46.920000	0.000
75%	4124.000000	49144.750000	2430.000000	600.160000	0.000
max	5000.000000	65535.000000	932858.000000	11199.968000	22638.480

In [10]:
```
data.hist(figsize=(12, 10))
plt.tight_layout()
plt.show()
```

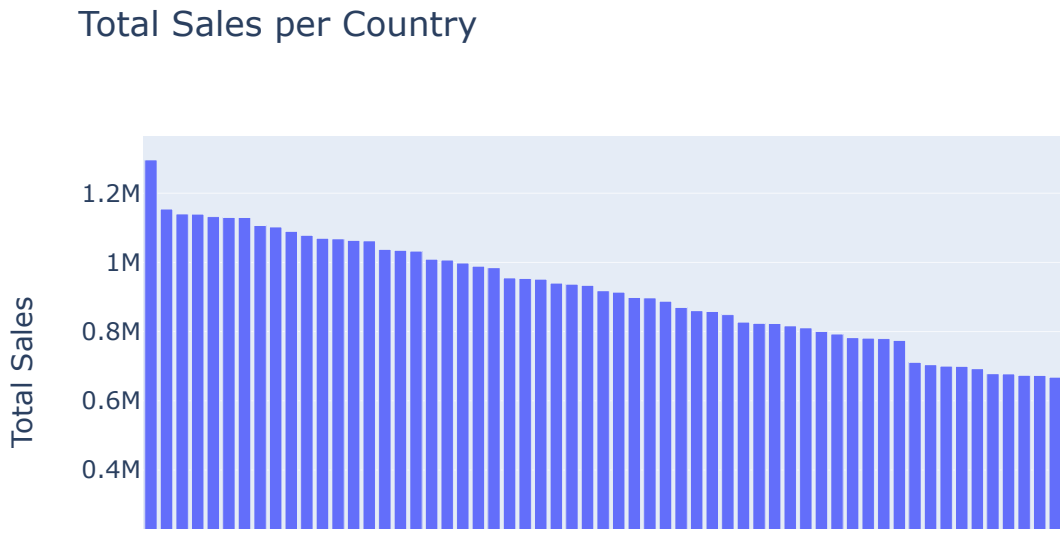


In [9]:

```
correlation = data.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap')
plt.show()
```

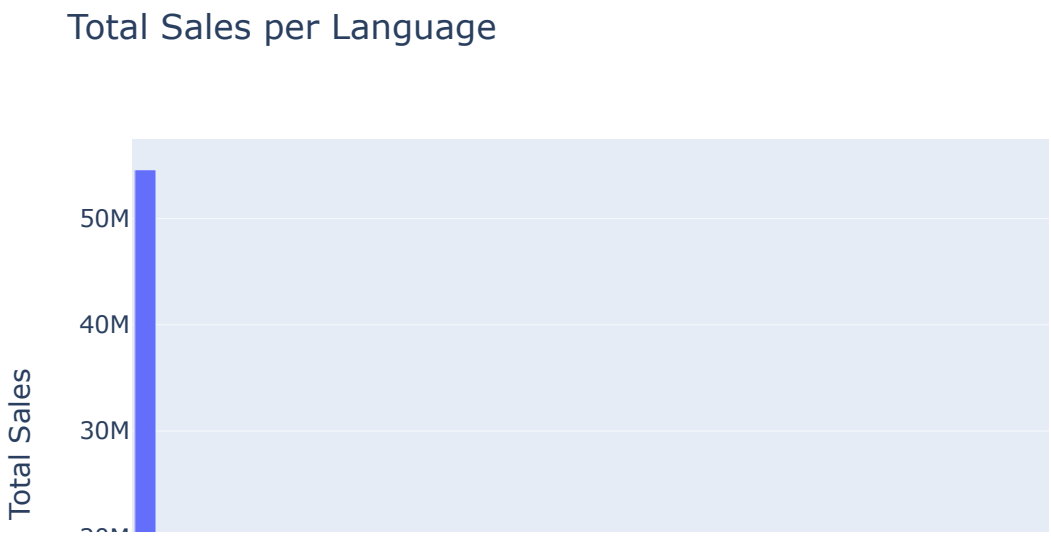


In [10]:
```
sales_country = data.groupby('country')['Sales'].sum().reset_index()
sales_country = sales_country.sort_values(by='Sales', ascending=False)
fig = px.bar(sales_country, x='country', y='Sales', title='Total Sales
fig.update_xaxes(title='Country')
fig.update_yaxes(title='Total Sales')
fig.show()
```

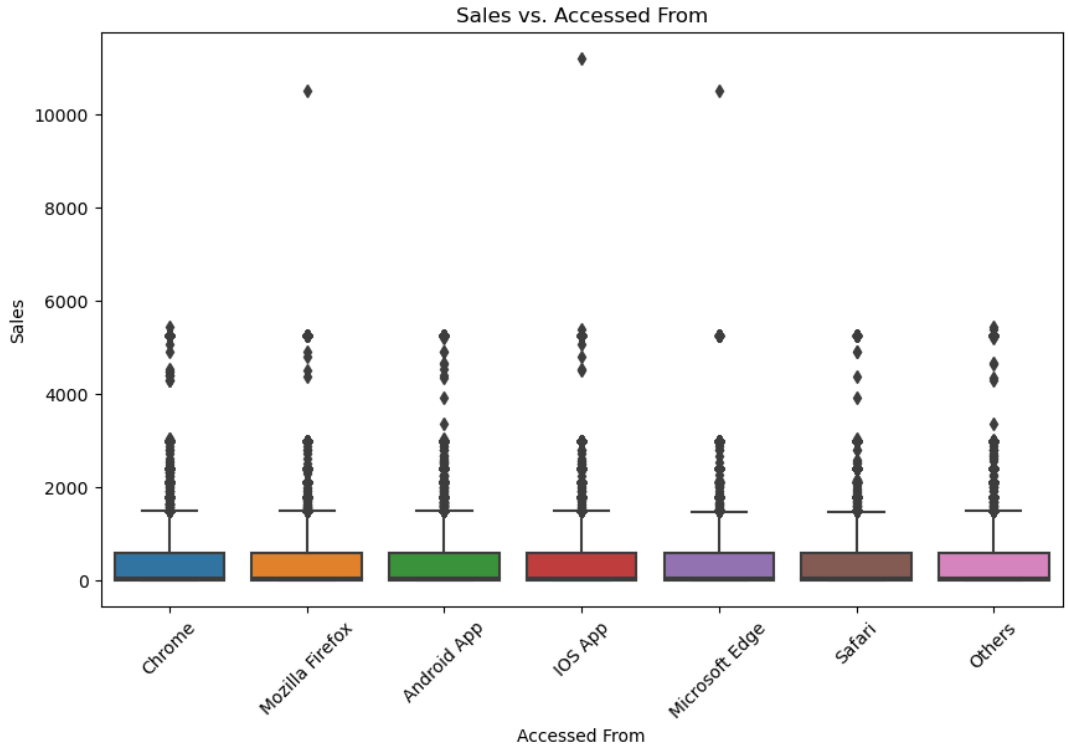


In [11]:

```
sales_language = data.groupby('Languages')['Sales'].sum().reset_index()
sales_language = sales_language.sort_values(by='Sales', ascending=False
fig = px.bar(sales_language, x='Languages', y='Sales', title='Total Sal
fig.update_xaxes(title='Language')
fig.update_yaxes(title='Total Sales')
fig.show()
```



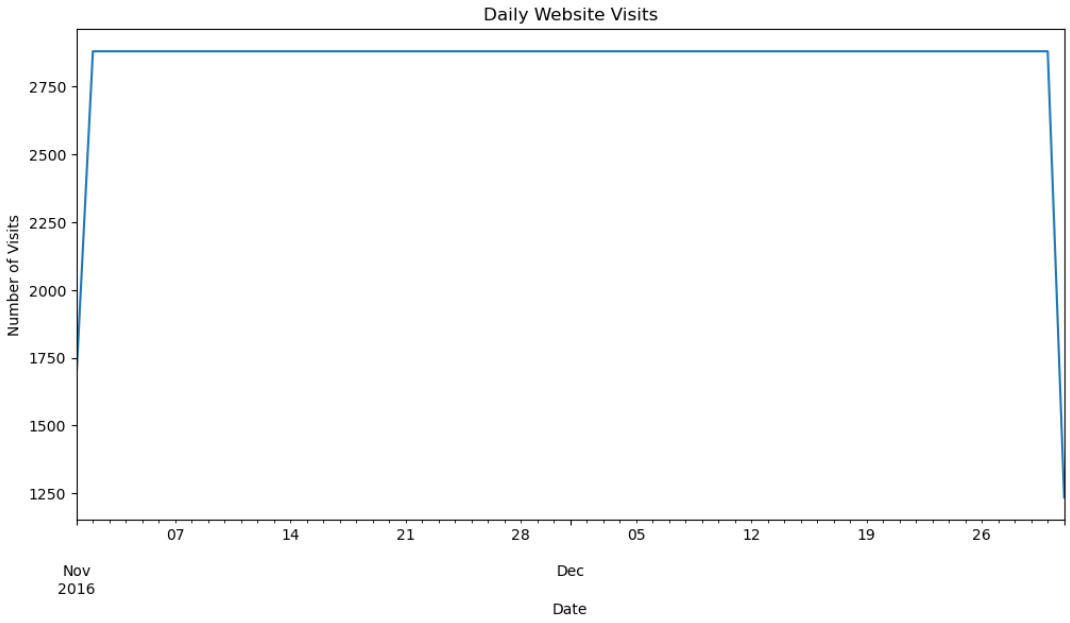
In [12]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(x='Accessed_From', y='Sales', data=data)
plt.title('Sales vs. Accessed From')
plt.xlabel('Accessed From')
plt.ylabel('Sales')
plt.xticks(rotation=45)
plt.show()
```



In [13]:
```
#TIME SERIES ANALYSIS OF THE DATASET
data['access_date'] = pd.to_datetime(data['access_date'])
```

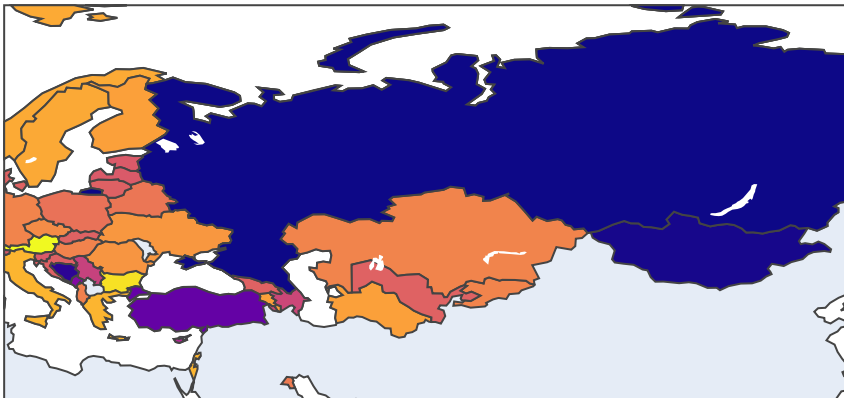
In [14]:
```
daily_visits = data.resample('D', on='access_date').size()
```

In [17]:
```
plt.figure(figsize=(12, 6))
daily_visits.plot(title='Daily Website Visits')
plt.xlabel('Date')
plt.ylabel('Number of Visits')
plt.show()
```

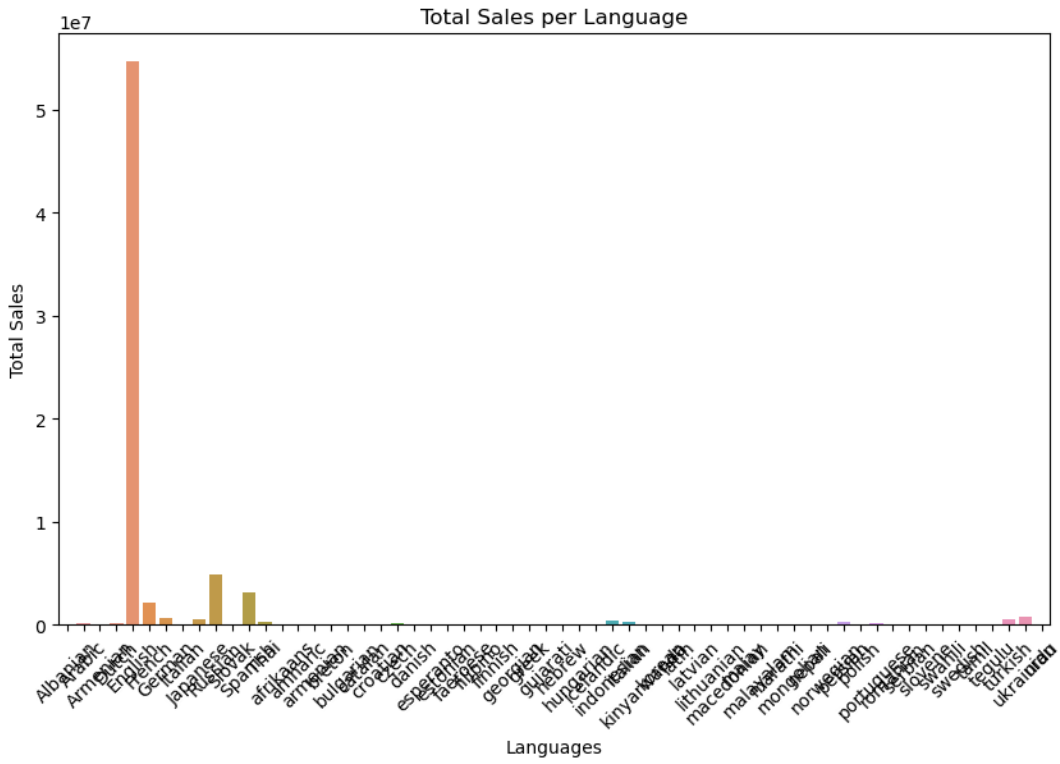


In [17]:
```
visits_by_country = data['country'].value_counts().reset_index()
visits_by_country.columns = ['Country', 'Visits']
```

In [18]:
```
# Create a map showing website access by country using Plotly
fig = px.choropleth(visits_by_country,
                    locations='Country',
                    locationmode='country names',
                    color='Visits',
                    hover_name='Country',
                    title='Website Visits by Country')
fig.show()
```



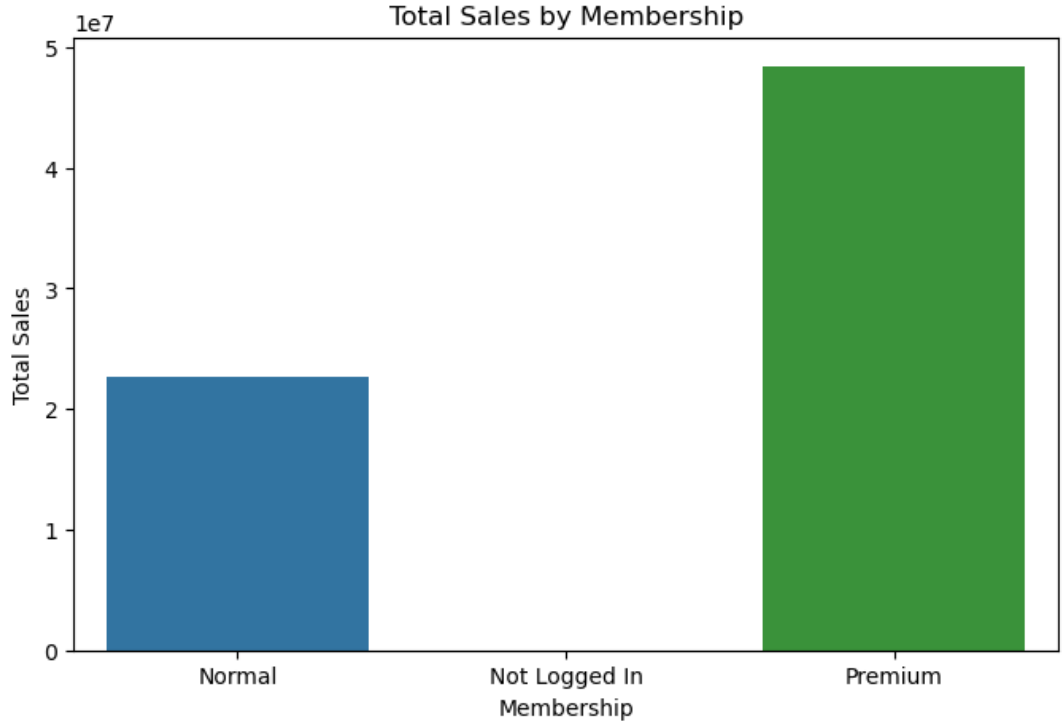
In [20]:



In [19]:

```
# Grouping data by membership and calculating total sales
sales_membership = data.groupby('membership')['Sales'].sum().reset_inde

# Visualize using a bar chart
plt.figure(figsize=(8, 5))
sns.barplot(x='membership', y='Sales', data=sales_membership)
plt.title('Total Sales by Membership')
plt.xlabel('Membership')
plt.ylabel('Total Sales')
plt.show()
```



In [ ]: