In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
data=pd.read_csv("Amazon Sales data.csv")
data
```

Out[2]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Re
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	5/28/2010	669165933	6/27/2010	9925	255.28	159.42	2533
1	Central America and the Caribbean	Grenada	Cereal	Online	C	8/22/2012	963881480	9/15/2012	2804	205.70	117.11	576
2	Europe	Russia	Office Supplies	Offline	L	5/2/2014	341417157	5/8/2014	1779	651.21	524.96	1158
3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	6/20/2014	514321792	7/5/2014	8102	9.33	6.92	75
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2/1/2013	115456712	2/6/2013	5062	651.21	524.96	3296
...	...	...	...	...	...	...	...	...	...	...	...	
95	Sub-Saharan Africa	Mali	Clothes	Online	M	7/26/2011	512878119	9/3/2011	888	109.28	35.84	97
96	Asia	Malaysia	Fruits	Offline	L	11/11/2011	810711038	12/28/2011	6267	9.33	6.92	58
97	Sub-Saharan Africa	Sierra Leone	Vegetables	Offline	C	6/1/2016	728815257	6/29/2016	1485	154.06	90.93	228
98	North America	Mexico	Personal Care	Offline	M	7/30/2015	559427106	8/8/2015	5767	81.73	56.67	471
99	Sub-Saharan Africa	Mozambique	Household	Offline	L	2/10/2012	665095412	2/15/2012	5367	668.27	502.54	3586

100 rows × 14 columns

Amazon Sales data refers to sales, high performing sellers and several other data points. There are millions of Amazon sellers around the world. Amazon sales data Analysis focuseson the process of analyzing consumer behavior, sales, and several other attributes in order to make improved, data-driven decisions. It is key to successfully sustaining their businesses and earning profits and for this purpose, they analyze different metrics like sales, Sales Quantity, Discount rate, Sales over years etc. By analyzing different metrics, you will be able to increase and improve your performance in terms of sales, Items to be sold and discount rates etc. Analysis of the sales data the main factor that contributes to sellers improving their business and increasing their revenue. They can better understand the market trends and customers' buying behaviors and help them cater to what the customers really want. In the world of rising new technology and innovation, E-commerce industry is advancing with the role of Data Analytics. Data analysis can help them to understand their business in a quiet different manner and helps to improve the quality of the service by identifying the weak areas of the business. This study demonstrates the how different analysis help to make better business decisions and help analyze customer trends and satisfaction, which can lead to new and better products and services. Different analysis performed to get the key insights from this data based on which business decisions will be taken.

In [3]:
```
data.head()
```

Out[3]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	5/28/2010	669165933	6/27/2010	9925	255.28	159.42	2533654.00	1
1	Central America and the Caribbean	Grenada	Cereal	Online	C	8/22/2012	963881480	9/15/2012	2804	205.70	117.11	576782.80	
2	Europe	Russia	Office Supplies	Offline	L	5/2/2014	341417157	5/8/2014	1779	651.21	524.96	1158502.59	
3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	6/20/2014	514321792	7/5/2014	8102	9.33	6.92	75591.66	
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2/1/2013	115456712	2/6/2013	5062	651.21	524.96	3296425.02	2

In [4]:
```
data.shape
```

Out[4]: (100, 14)

In [5]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 14 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Region          100 non-null    object 
 1   Country         100 non-null    object 
 2   Item Type       100 non-null    object 
 3   Sales Channel   100 non-null    object 
 4   Order Priority  100 non-null    object 
 5   Order Date      100 non-null    object 
 6   Order ID        100 non-null    int64  
 7   Ship Date       100 non-null    object 
 8   Units Sold      100 non-null    int64  
 9   Unit Price      100 non-null    float64
 10  Unit Cost       100 non-null    float64
 11  Total Revenue   100 non-null    float64
 12  Total Cost      100 non-null    float64
 13  Total Profit    100 non-null    float64
dtypes: float64(5), int64(2), object(7)
memory usage: 11.1+ KB
```

In [6]:
```
data.columns
```

Out[6]:
```
Index(['Region', 'Country', 'Item Type', 'Sales Channel', 'Order Priority',
       'Order Date', 'Order ID', 'Ship Date', 'Units Sold', 'Unit Price',
       'Unit Cost', 'Total Revenue', 'Total Cost', 'Total Profit'],
      dtype='object')
```

In [7]:
```
data[['Units Sold','Unit Price','Unit Cost','Total Revenue','Total Cost','Total Profit']].de
```

Out[7]:

	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Total Profit
count	100.000000	100.000000	100.000000	1.000000e+02	1.000000e+02	1.000000e+02
mean	5128.710000	276.761300	191.048000	1.373488e+06	9.318057e+05	4.416820e+05
std	2794.484562	235.592241	188.208181	1.460029e+06	1.083938e+06	4.385379e+05
min	124.000000	9.330000	6.920000	4.870260e+03	3.612240e+03	1.258020e+03
25%	2836.250000	81.730000	35.840000	2.687212e+05	1.688680e+05	1.214436e+05
50%	5382.500000	179.880000	107.275000	7.523144e+05	3.635664e+05	2.907680e+05
75%	7369.000000	437.200000	263.330000	2.212045e+06	1.613870e+06	6.358288e+05
max	9925.000000	668.270000	524.960000	5.997055e+06	4.509794e+06	1.719922e+06

In [8]:
```
data.duplicated().sum()
```

Out[8]: 0

In [9]:
```
data.isnull().sum()
```

Out[9]:
```
Region            0
Country           0
Item Type         0
Sales Channel     0
Order Priority    0
Order Date        0
Order ID          0
Ship Date         0
Units Sold        0
Unit Price        0
Unit Cost         0
Total Revenue     0
Total Cost        0
Total Profit      0
dtype: int64
```

Now we are changing date and time format of order date and ship date for training

In [10]:
```
data["Order Date"]=pd.to_datetime(data['Order Date'])
data["Ship Date"]=pd.to_datetime(data['Ship Date'])

```

Changing the data type column of different columns for training the model

In [11]:
```
data['Region']=data['Region'].astype(str)
data['Country']=data['Country'].astype(str)
data['Item Type']=data['Item Type'].astype(str)
data['Sales Channel']=data['Sales Channel'].astype(str)
data['Order Priority']=data['Order Priority'].astype(str)
```

In [12]:
```
pd.set_option('display.max_rows', None)
data['Country'].value_counts()
```

```
Out[12]: The Gambia                          4
         Sierra Leone                        3
         Sao Tome and Principe               3
         Mexico                              3
         Australia                           3
         Djibouti                            3
         Switzerland                         2
         Myanmar                             2
         Norway                              2
         Turkmenistan                        2
         Cameroon                            2
         Bulgaria                            2
         Honduras                            2
         Azerbaijan                          2
         Libya                               2
         Rwanda                              2
         Mali                                2
         Gabon                               1
         Belize                              1
         Haiti                               1
         Lithuania                           1
         San Marino                          1
         United Kingdom                      1
         Austria                             1
         Fiji                                1
         Madagascar                          1
         Cote d'Ivoire                       1
         Tuvalu                              1
         Democratic Republic of the Congo    1
         Zambia                              1
         Malaysia                            1
         Nicaragua                           1
         Romania                             1
         Slovenia                            1
         Kuwait                              1
         Kenya                               1
         Iran                                1
         Pakistan                            1
         Lebanon                             1
         Spain                               1
         Samoa                               1
         Monaco                              1
         Laos                                1
         Saudi Arabia                        1
         Federated States of Micronesia      1
         Slovakia                            1
         Lesotho                             1
         Albania                             1
         Russia                              1
         Solomon Islands                     1
         Angola                              1
         Burkina Faso                        1
         Republic of the Congo               1
         Senegal                             1
         Kyrgyzstan                          1
         Cape Verde                          1
         Bangladesh                          1
         Mongolia                            1
         Sri Lanka                           1
         East Timor                          1
         Portugal                            1
         New Zealand                         1
         Moldova                             1
         France                              1
         Kiribati                            1
         South Sudan                         1
```

```
Costa Rica                         1
Syria                              1
Brunei                             1
Niger                              1
Grenada                            1
Comoros                            1
Iceland                            1
Macedonia                          1
Mauritania                         1
Mozambique                         1
Name: Country, dtype: int64
```

In [13]:
```
data['Item Type'].value_counts()
```

Out[13]:
```
Clothes            13
Cosmetics          13
Office Supplies    12
Fruits             10
Personal Care      10
Household           9
Beverages           8
Baby Food           7
Cereal              7
Vegetables          6
Snacks              3
Meat                2
Name: Item Type, dtype: int64
```

In [14]:
```
data['Sales Channel'].value_counts()
```

Out[14]:
```
Offline    50
Online     50
Name: Sales Channel, dtype: int64
```

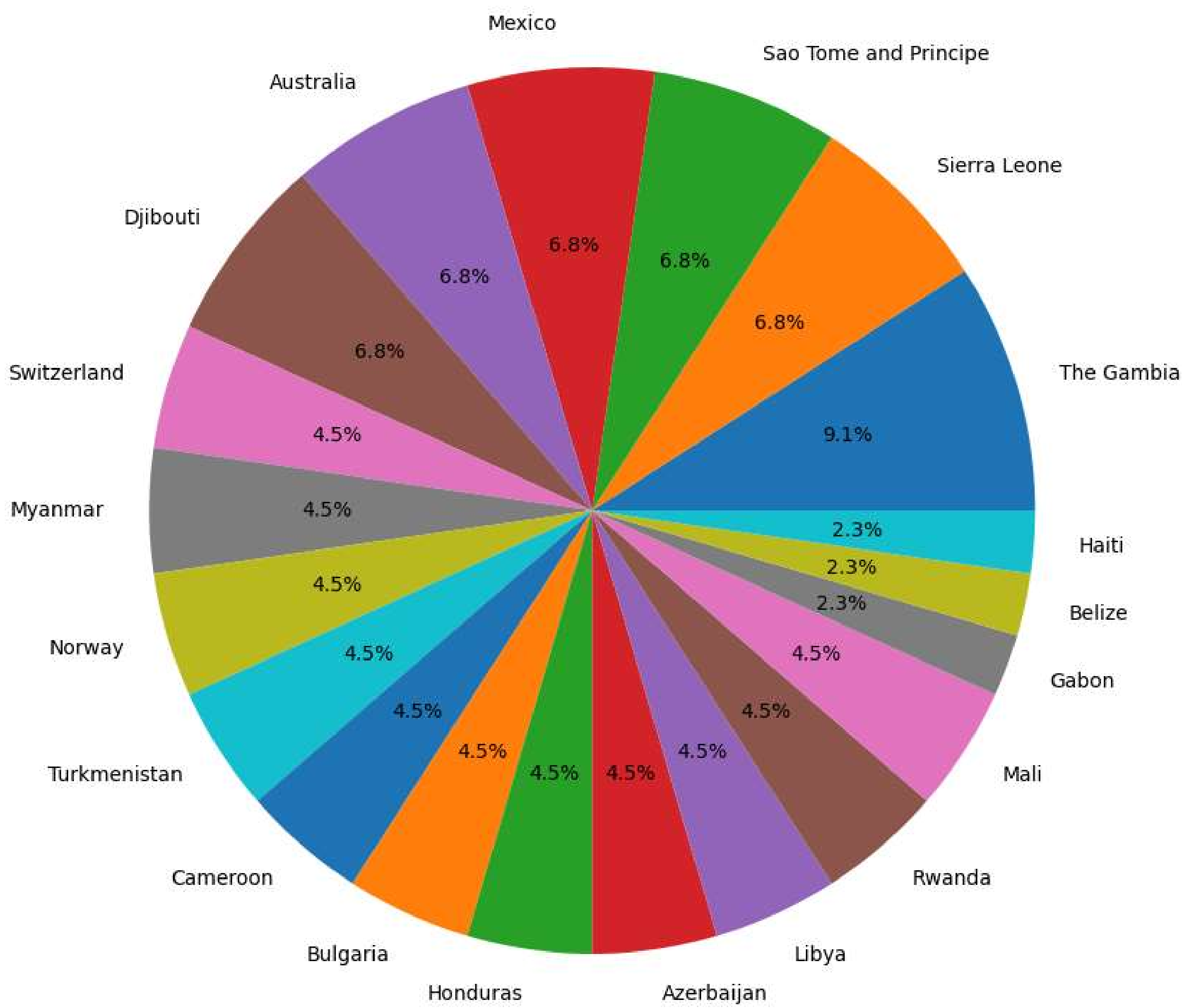
In [15]:
```
data['Order Priority'].value_counts()

```

Out[15]:
```
H    30
L    27
C    22
M    21
Name: Order Priority, dtype: int64
```

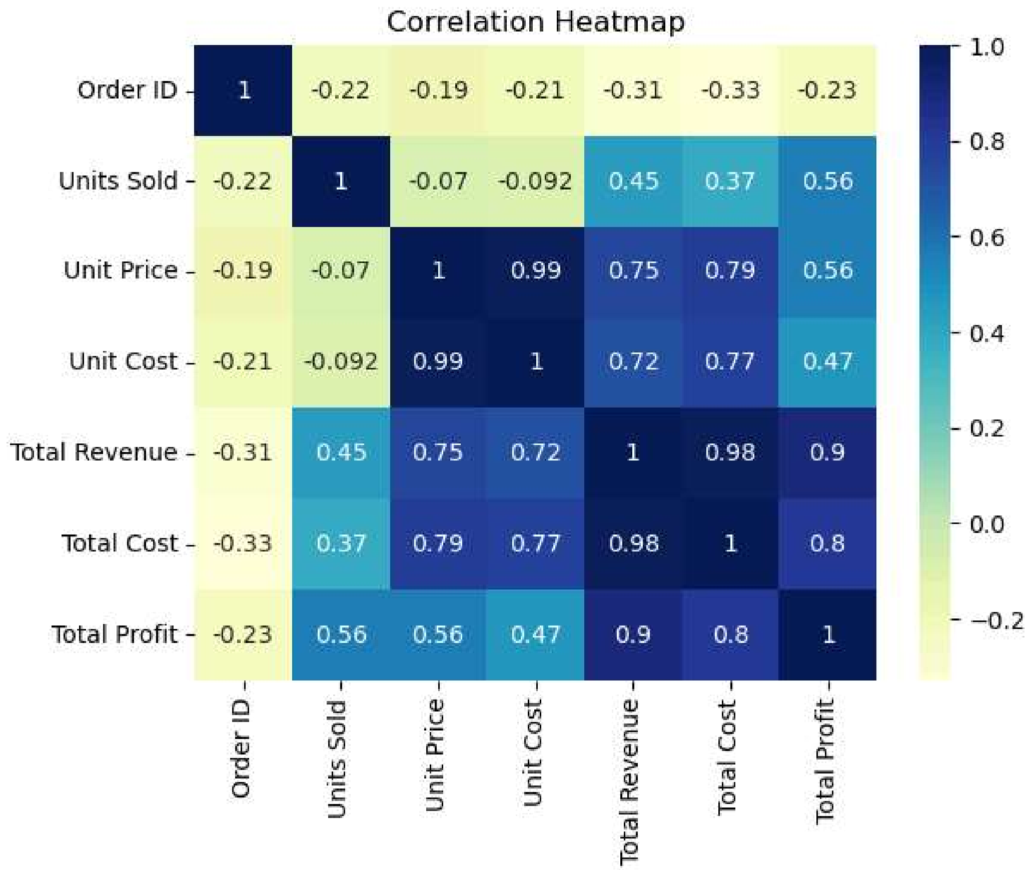
Let's see in pie chart for top 20 country

In [16]:
```
country_names = data.Country.value_counts().index
country_val = data.Country.value_counts().values
fig,ax = plt.subplots(figsize=(10,10))
ax.pie(country_val[:20],labels=country_names[:20],autopct='%1.1f%%')
plt.title("Top 20 Countries")
plt.show()
```

Top 20 Countries



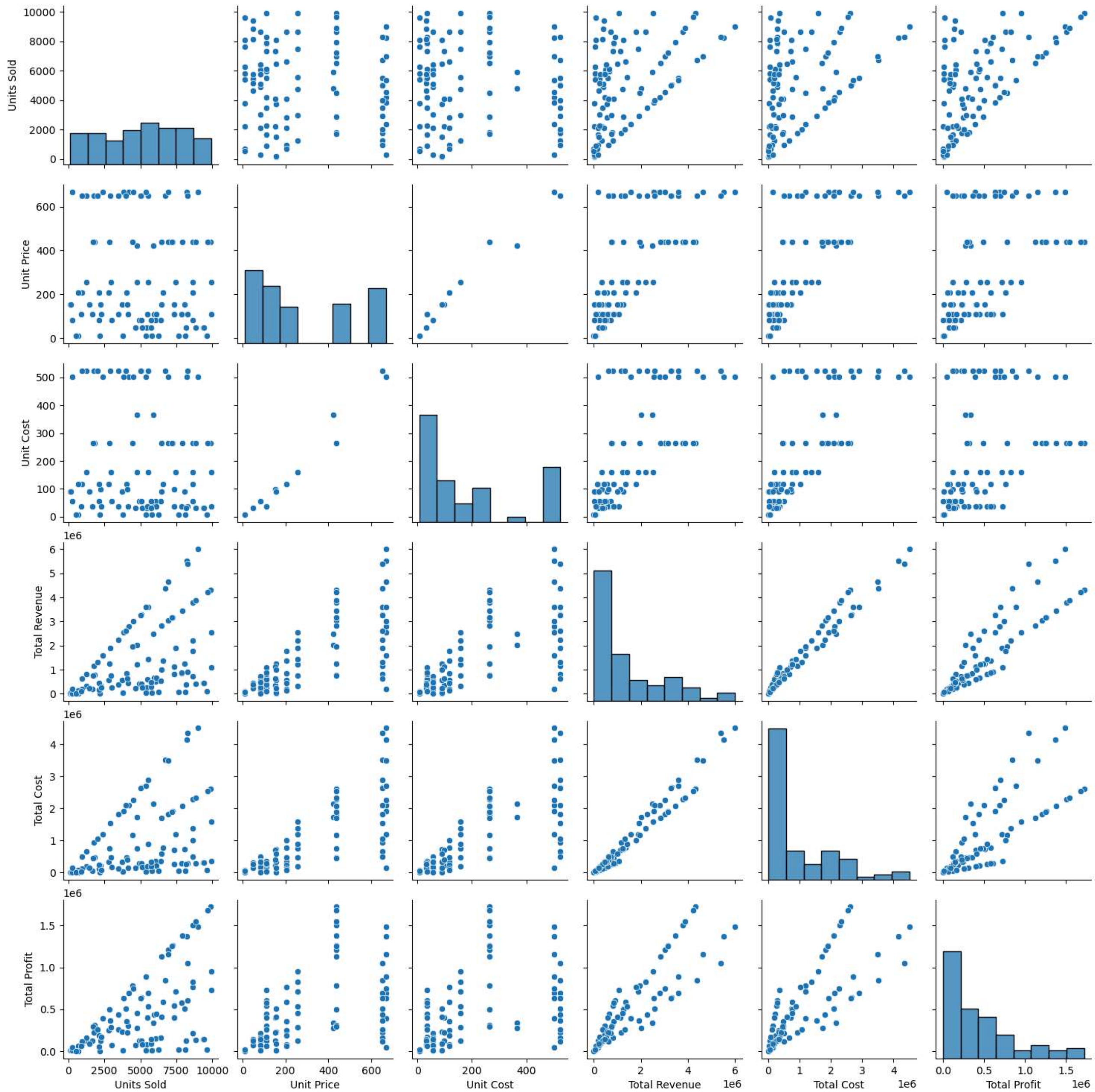
In [17]:
```
sns.heatmap(data.corr(),annot=True ,cmap='YlGnBu',linecolor='black')
plt.title('Correlation Heatmap')
plt.show()
```



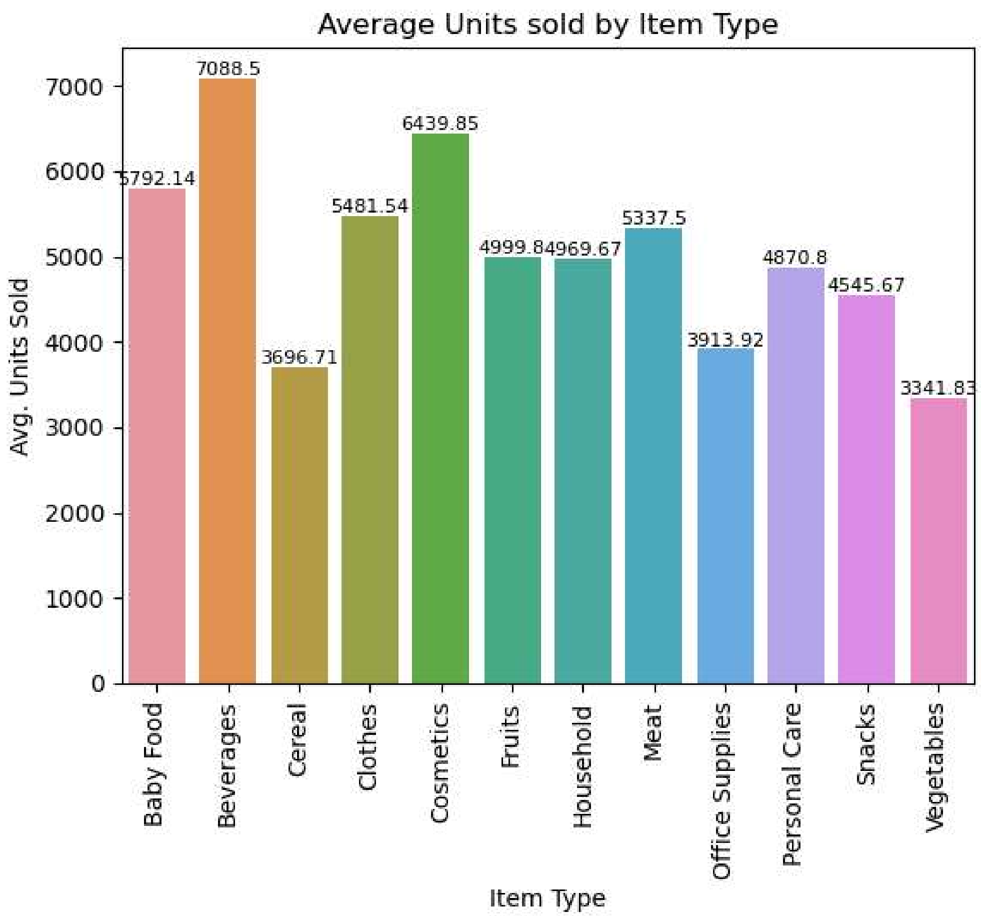
- We can see in the above heatmap Unit Price and Unit Cost are stronger corelated.
- Unit Price also related to Total Revenue and Total Cost.

In [18]:
```
Variables=["Units Sold",'Unit Price',"Unit Cost","Total Revenue","Total Cost","Total Profit"
```

In [19]:
```
sns.pairplot(data[Variables])
plt.show()
```

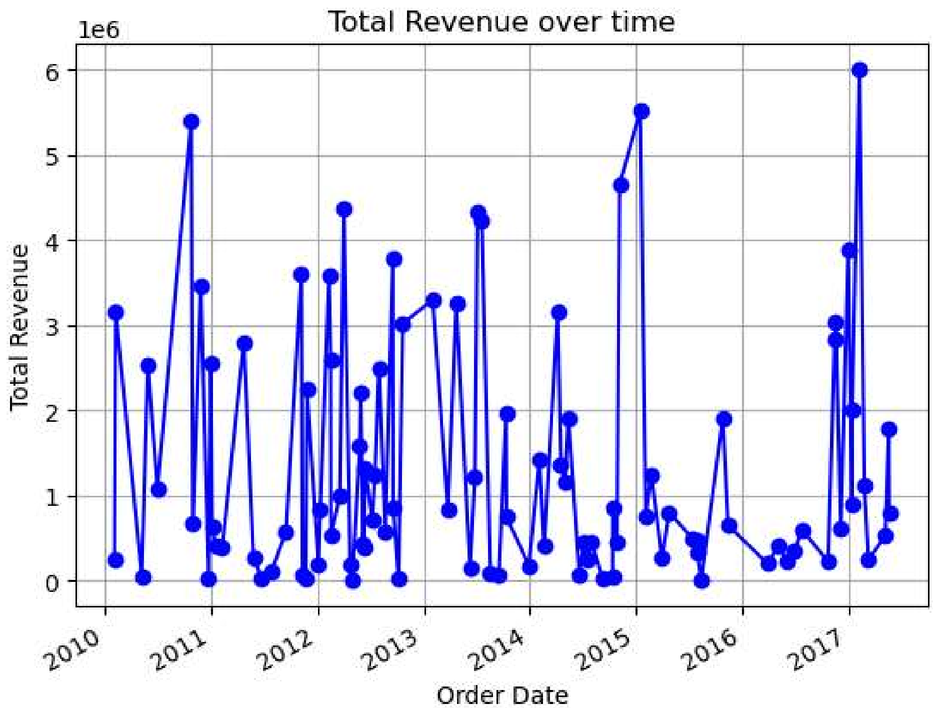


In [20]:
```
Avg_unitsold=data.groupby('Item Type')['Units Sold'].mean().reset_index()
ax=sns.barplot(x='Item Type',y='Units Sold',data=Avg_unitsold)
ax.bar_label(ax.containers[0],fontsize=8)

plt.xlabel('Item Type')
plt.ylabel('Avg. Units Sold')
plt.title('Average Units sold by Item Type')
plt.xticks(rotation=90)
plt.show()
```



In [21]:
```
data.groupby('Order Date').sum()['Total Revenue'].plot(kind = 'line',color = 'blue',marker='
plt.xlabel('Order Date')
plt.ylabel('Total Revenue')

plt.title('Total Revenue over time')
print('Total Revenue:',data['Total Revenue'].sum())
plt.show()
```

```
Total Revenue: 137348768.31
```

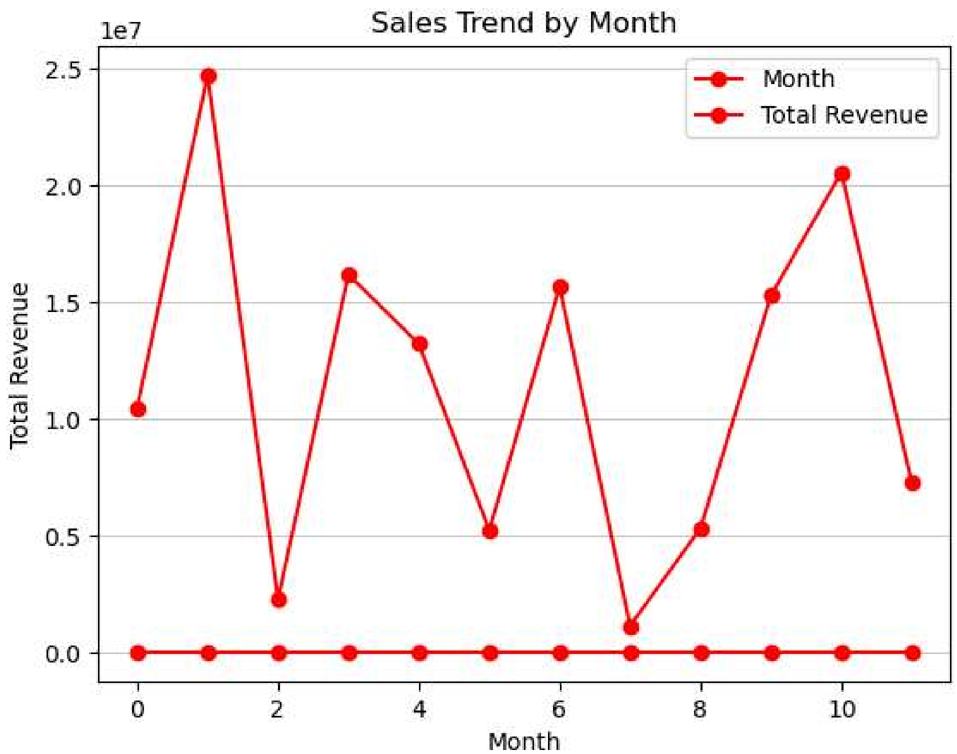


In the above line chart we can see that total revenue over all year from 2010 till 2017 is 137348768.31

In [22]:
```
data['Month']=data['Order Date'].dt.month
data['Year']=data['Order Date'].dt.year
```

In [23]:
```
monthly_sales=data.groupby(data['Month'])['Total Revenue'].sum().reset_index()
```

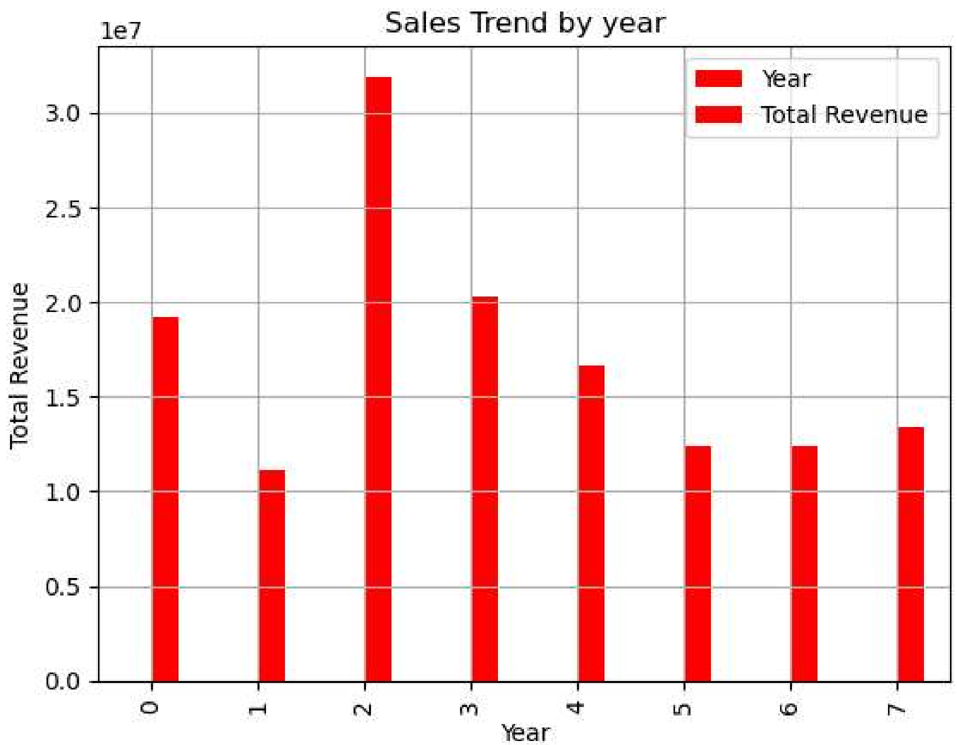
In [24]:
```
ax=monthly_sales.plot(kind = 'line',marker='o',color = 'red')
ax.grid(axis='y', linewidth=0.5)
ax.set_xlabel('Month')
ax.set_ylabel('Total Revenue')
ax.set_title('Sales Trend by Month')
plt.show()
```



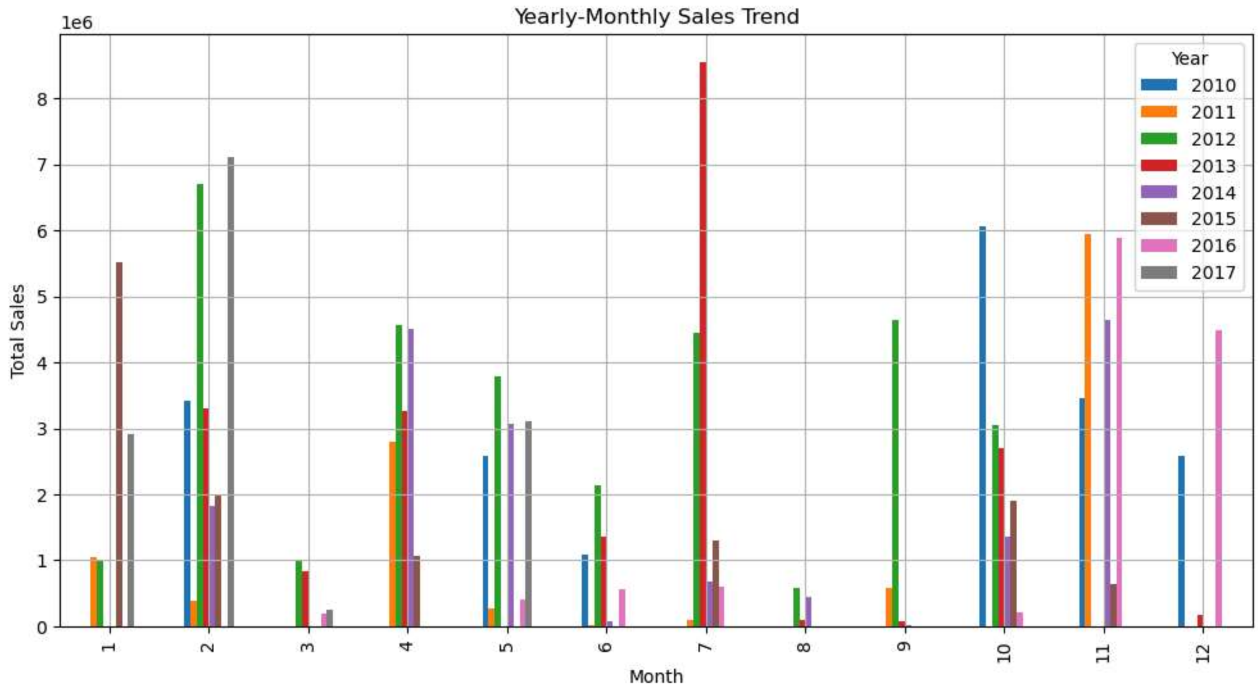
In [25]:
```
year_sales=data.groupby(data['Year'])['Total Revenue'].sum().reset_index()
```

In [26]:

```
year_sales.plot(kind = 'bar',color = 'red',grid=True)
plt.xlabel('Year')
plt.ylabel('Total Revenue')
plt.title('Sales Trend by year')
plt.show()
```



In [27]:
```
Yearly_Monthly_Sales = data.groupby(['Month','Year'])['Total Revenue'].sum().unstack()
Yearly_Monthly_Sales.plot(kind='bar', figsize=(12, 6))
plt.title('Yearly-Monthly Sales Trend')
plt.xlabel('Month')
plt.ylabel('Total Sales')
plt.legend(title='Year')
plt.grid(True)
plt.show()
```

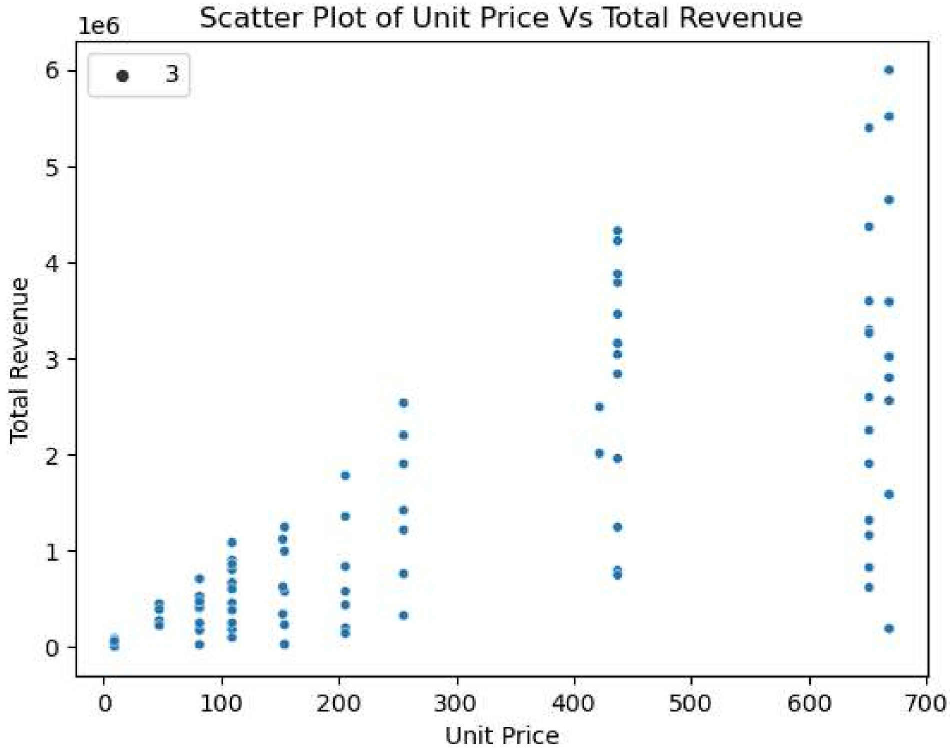


In [28]:
```
total_revenue = data['Total Revenue'].sum()
average_order_value = data['Total Revenue'].mean()
print(f'Total Revenue: ${total_revenue:.2f}')
print(f'Average Order Value: ${average_order_value:.2f}')
```

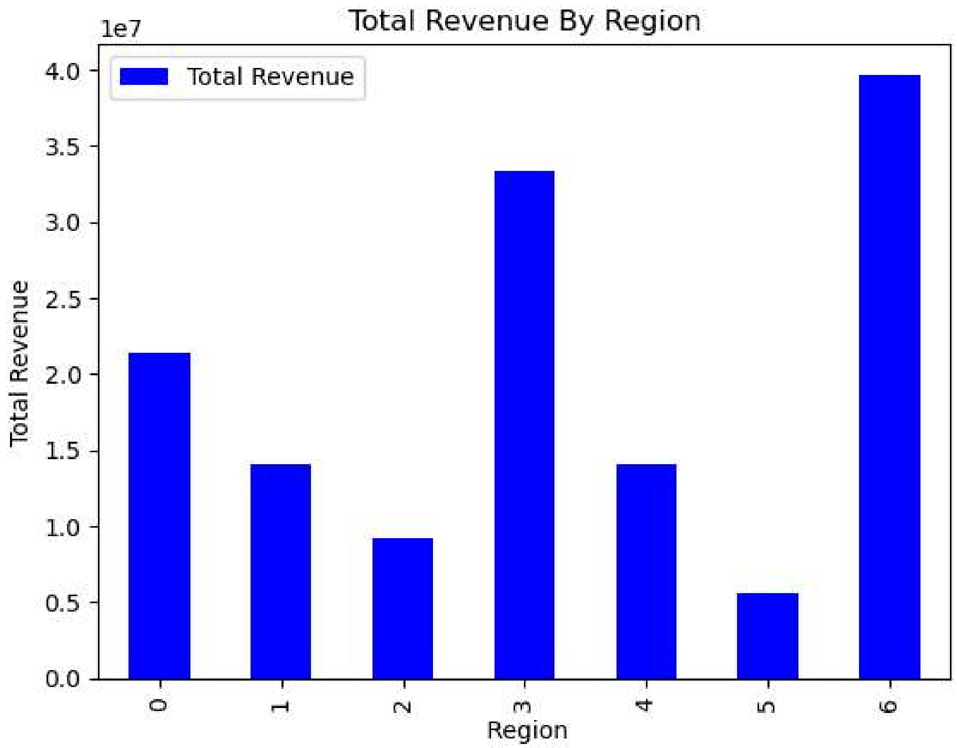
```
Total Revenue: $137348768.31
Average Order Value: $1373487.68
```

In [29]:
```
#Relation between Unit Price and Total Revenue.
```

In [30]:
```
sns.scatterplot(data=data,x='Unit Price',y='Total Revenue',marker='o',size=3)
plt.xlabel('Unit Price')
plt.ylabel('Total Revenue')
plt.title('Scatter Plot of Unit Price Vs Total Revenue')
plt.show()
```



In [31]:
```
region=data.groupby('Region')['Total Revenue'].sum().reset_index()
region.plot(kind="bar",color='blue')
plt.xlabel(" Region")
plt.ylabel("Total Revenue")
plt.title("Total Revenue By Region")
plt.show()
```



In [32]:
```
sns.barplot(x='Order Priority',y='Total Profit',data=data)
plt.xlabel(" Order Priority")
plt.ylabel("Total Profit")
plt.title("Total Revenue By Order Priority")
plt.show()
```

Observation based on anaylsis

- Total revenue has been increasing steadily over the past few years.
- There is a positive correlation between Unit price and total revenue, indicating that higher-priced items contribute more to revenue.
- The average order value is within an acceptable range, suggesting that customers are making purchases of reasonable value.

Recommendations:-

- Explore strategies to further increase total revenue, such as introducing premium-priced products or expanding into new markets.
- Consider optimizing pricing strategies to maximize revenue without sacrificing customer satisfaction.
- Monitor and analyze sales data regularly to identify trends and opportunities for improvement.

In [ ]:
```

```