In [1]:
```python
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```python
data= pd.read_csv('E:/DataAnalyst/Python Project/Amazon Sales data.csv')
data= pd.DataFrame(data= data)
print('Shape before deleting duplicate values:', data.shape)

data=data.drop_duplicates()
print('Shape After deleting duplicate values:', data.shape)
data.head()
```

```
Shape before deleting duplicate values: (100, 14)
Shape After deleting duplicate values: (100, 14)
```

Out[3]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total C
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	5/28/2010	669165933	6/27/2010	9925	255.28	159.42	2533654.00	1582243
1	Central America and the Caribbean	Grenada	Cereal	Online	C	8/22/2012	963881480	9/15/2012	2804	205.70	117.11	576782.80	328376
2	Europe	Russia	Office Supplies	Offline	L	05-02-2014	341417157	05-08-2014	1779	651.21	524.96	1158502.59	933903
3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	6/20/2014	514321792	07-05-2014	8102	9.33	6.92	75591.66	56065
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	02-01-2013	115456712	02-06-2013	5062	651.21	524.96	3296425.02	2657347

Basic Data Exploration

In [4]:

```
data.head()
```

Out[4]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total C
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	5/28/2010	669165933	6/27/2010	9925	255.28	159.42	2533654.00	1582243
1	Central America and the Caribbean	Grenada	Cereal	Online	C	8/22/2012	963881480	9/15/2012	2804	205.70	117.11	576782.80	328376
2	Europe	Russia	Office Supplies	Offline	L	05-02-2014	341417157	05-08-2014	1779	651.21	524.96	1158502.59	933903
3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	6/20/2014	514321792	07-05-2014	8102	9.33	6.92	75591.66	56065
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	02-01-2013	115456712	02-06-2013	5062	651.21	524.96	3296425.02	2657347

In [5]:
```
data.columns
```

Out[5]:
```
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 [6]:
```
data.shape
```

Out[6]:
```
(100, 14)
```

In [7]:
```
data.size
```

Out[7]:
```
1400
```

In [8]:
```
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 [9]:
```
data.describe()
```

Out[9]:

	Order ID	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Total Profit
count	1.000000e+02	100.000000	100.000000	100.000000	1.000000e+02	1.000000e+02	1.000000e+02
mean	5.550204e+08	5128.710000	276.761300	191.048000	1.373488e+06	9.318057e+05	4.416820e+05
std	2.606153e+08	2794.484562	235.592241	188.208181	1.460029e+06	1.083938e+06	4.385379e+05
min	1.146066e+08	124.000000	9.330000	6.920000	4.870260e+03	3.612240e+03	1.258020e+03
25%	3.389225e+08	2836.250000	81.730000	35.840000	2.687212e+05	1.688680e+05	1.214436e+05
50%	5.577086e+08	5382.500000	179.880000	107.275000	7.523144e+05	3.635664e+05	2.907680e+05
75%	7.907551e+08	7369.000000	437.200000	263.330000	2.212045e+06	1.613870e+06	6.358288e+05
max	9.940222e+08	9925.000000	668.270000	524.960000	5.997055e+06	4.509794e+06	1.719922e+06

In [10]:
```python
data.isna().sum() # NO ANY NULL VALUE PRESENT IN OUR DATASET.
```

Out[10]:
```
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
```

In [11]:
```python
data.dtypes
```

Out[11]:
```
Region             object
Country            object
Item Type          object
Sales Channel      object
Order Priority     object
Order Date         object
Order ID            int64
Ship Date          object
Units Sold          int64
Unit Price        float64
Unit Cost         float64
Total Revenue     float64
Total Cost        float64
Total Profit      float64
dtype: object
```

In [12]:
```python
data['Order Date']=pd.to_datetime(data['Order Date'],format='mixed')
data['Ship Date']=pd.to_datetime(data['Ship Date'],format='mixed')
#'Ship Date'
```

In [13]:
```python
data.dtypes
```

Out[13]:
```
Region                    object
Country                   object
Item Type                 object
Sales Channel             object
Order Priority            object
Order Date        datetime64[ns]
Order ID                   int64
Ship Date         datetime64[ns]
Units Sold                 int64
Unit Price               float64
Unit Cost                float64
Total Revenue            float64
Total Cost               float64
Total Profit             float64
dtype: object
```

In [14]:
```
data['Total Cost']= data['Total Cost'].astype('Float64')
data.head()
```

Out[14]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	2010-05-28	669165933	2010-06-27	9925	255.28	159.42	2533654.00	1582243.5	951
1	Central America and the Caribbean	Grenada	Cereal	Online	C	2012-08-22	963881480	2012-09-15	2804	205.70	117.11	576782.80	328376.44	248
2	Europe	Russia	Office Supplies	Offline	L	2014-05-02	341417157	2014-05-08	1779	651.21	524.96	1158502.59	933903.84	224
3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	2014-06-20	514321792	2014-07-05	8102	9.33	6.92	75591.66	56065.84	19
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2013-02-01	115456712	2013-02-06	5062	651.21	524.96	3296425.02	2657347.52	639

In [ ]:

Data Analysis:

Queries:

1. Which region(s) has the highest total sales revenue?

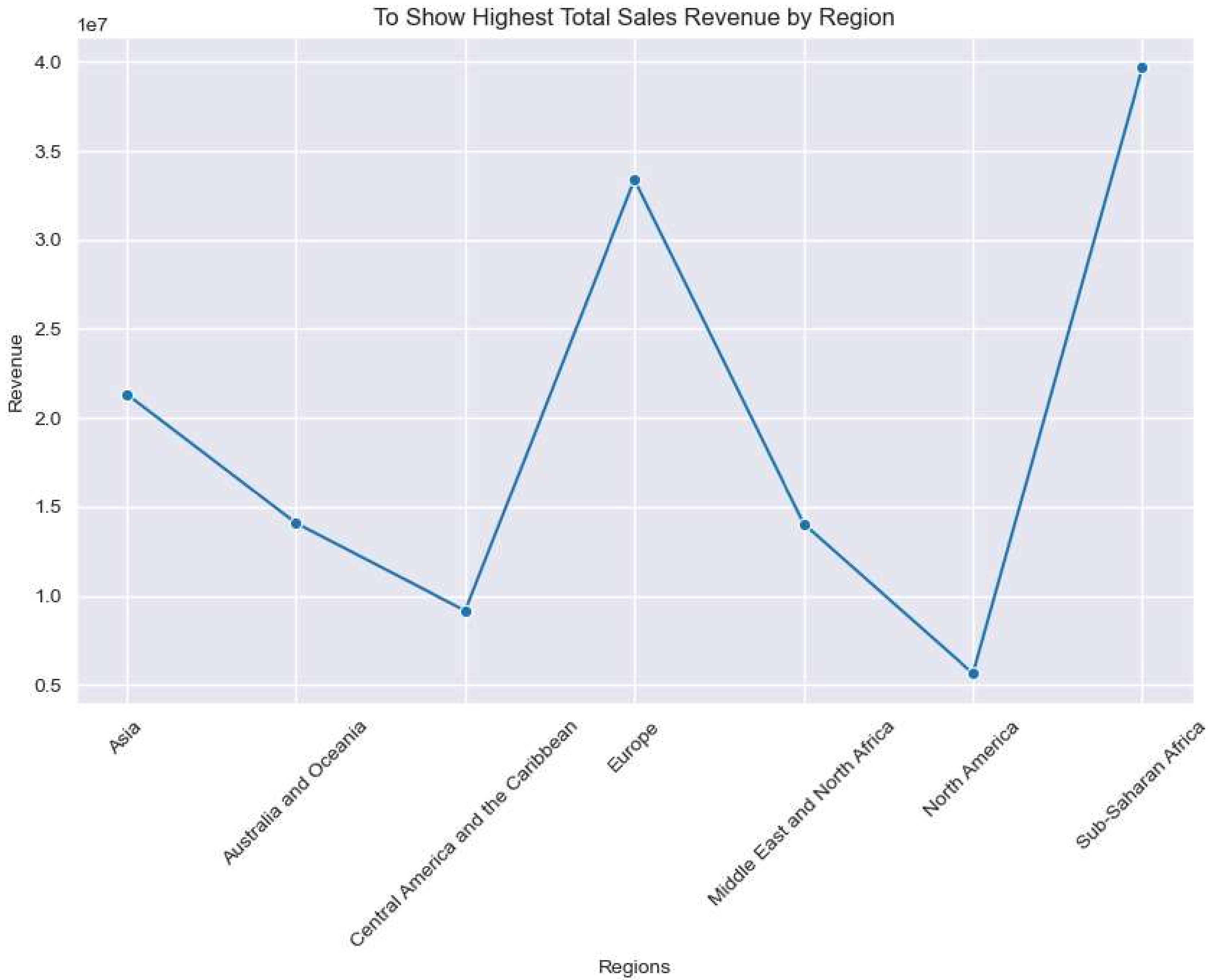
In [15]:

```
Highest_Total_Revenue= data.groupby(data['Region'])['Total Revenue'].sum()
Highest_Total_Revenue.idxmax()
```

Out[15]: 'Sub-Saharan Africa'

In [16]:

```python
group_data= data.groupby(data['Region'])['Total Revenue'].sum()

plt.figure(figsize=(10,6))
sns.set_style('darkgrid')
sns.lineplot(data= group_data, marker='o')
plt.xticks(rotation= 45)
plt.title('To Show Highest Total Sales Revenue by Region')
plt.xlabel('Regions')
plt.ylabel('Revenue')
plt.show()


# 1e7 is scientific form. it means 1*10**7= 10,000,000
```



To Show Highest Total Sales Revenue by Region
1e7
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
Revenue
Asia
Australia and Oceania
Central America and the Caribbean
Europe
Middle East and North Africa
North America
Sub-Saharan Africa
Regions

2. What is the average unit price and unit cost for each item type?

In [17]:
```
Avg_Unit_Price= data.groupby(data['Item Type'])['Unit Price'].mean()
Avg_Unit_Cost= data.groupby(data['Item Type'])['Unit Cost'].mean()

Avg_Price_Cost= pd.DataFrame({'Average Unit Price': Avg_Unit_Price,
                              'Average Unit Cost': Avg_Unit_Cost})

Avg_Price_Cost
```

Out[17]:

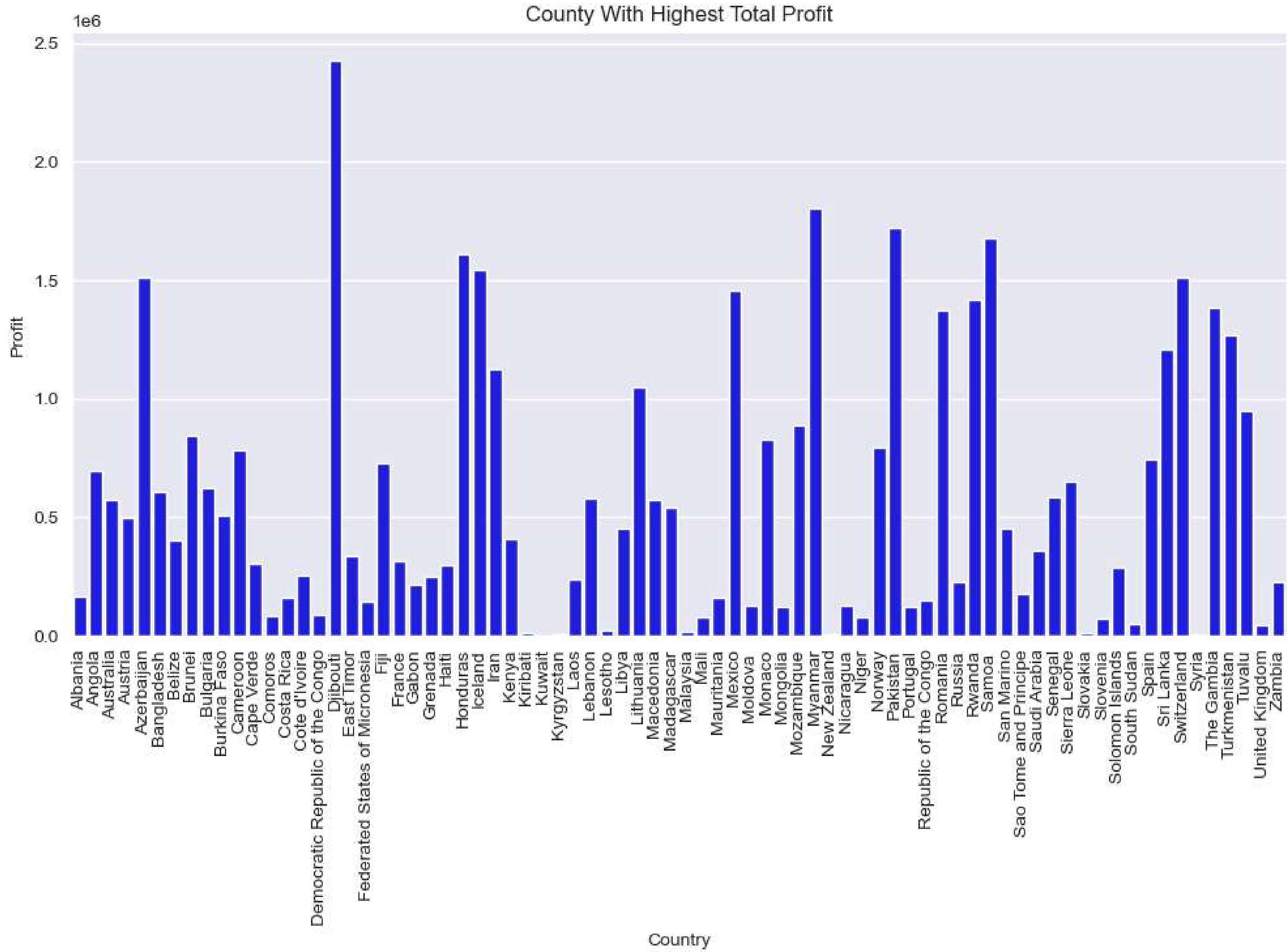
Item Type	Average Unit Price	Average Unit Cost
Baby Food	255.28	159.42
Beverages	47.45	31.79
Cereal	205.70	117.11
Clothes	109.28	35.84
Cosmetics	437.20	263.33
Fruits	9.33	6.92
Household	668.27	502.54
Meat	421.89	364.69
Office Supplies	651.21	524.96
Personal Care	81.73	56.67
Snacks	152.58	97.44
Vegetables	154.06	90.93

3. Which country has the highest total profit?

In [18]:
```
Total_Profit_By_Comapany= data.groupby(data['Country']) ['Total Profit'].sum()
Highest_Total_Profit_County= Total_Profit_By_Comapany.idxmax()

print("Country with the highest total profit:",Highest_Total_Profit_County)
```

```
Country with the highest total profit: Djibouti
```

In [19]:
```
group_data= data.groupby(data['Country']) ['Total Profit'].sum()

plt.figure(figsize=(12,6))
sns.set_style('darkgrid')
sns.barplot(x= group_data.index, y= group_data, color='blue')
plt.xticks(rotation= 90)
plt.title('County With Highest Total Profit' )
plt.xlabel('Country')
plt.ylabel('Profit')
plt.show()
```



4. How does the sales channel affect the order priority distribution?

In [20]:
```
Sales_Channel_Order_Priority_Distribution= data.groupby(data['Sales Channel']) ['Order Priority'].value_counts()
Sales_Channel_Order_Priority_Distribution
```

Out[20]:
```
Sales Channel  Order Priority
Offline        H                 17
               C                 13
               L                 12
               M                  8
Online         L                 15
               H                 13
               M                 13
               C                  9
Name: count, dtype: int64
```

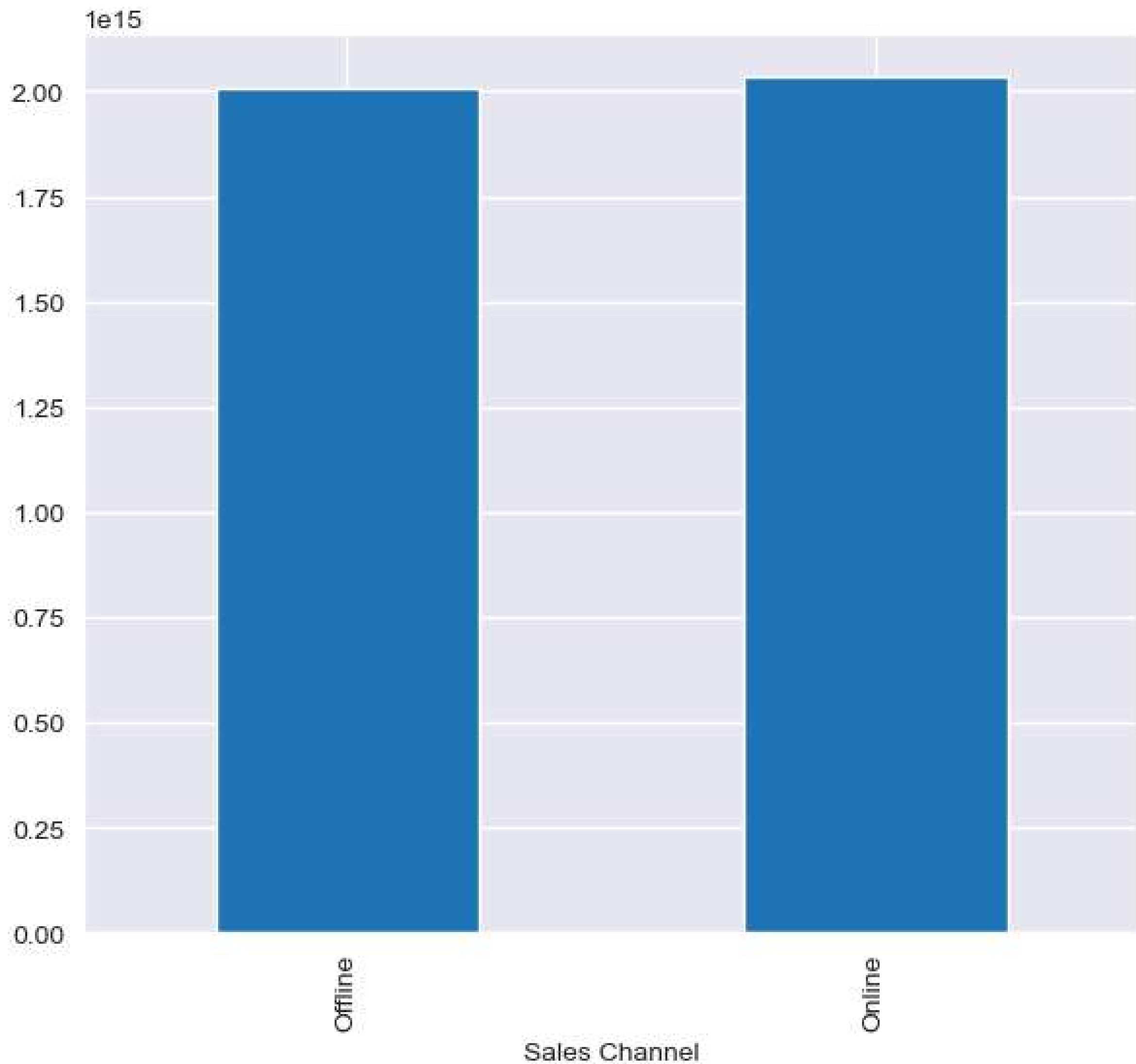
5. What is the average order processing time (duration between order and ship dates) for each sales channel?

In [21]:
```
data['Processing Time']= data['Ship Date']-data['Order Date']

Avg_Processing_Time= data.groupby(data['Sales Channel'])['Processing Time'].mean()
Avg_Processing_Time
```

Out[21]:
```
Sales Channel
Offline   23 days 04:48:00
Online    23 days 12:28:48
Name: Processing Time, dtype: timedelta64[ns]
```

In [22]:
```
plt.figure(figsize=(7, 6))

Avg_Processing_Time.plot(kind='bar',grid='bool')
```

Out[22]:
```
<Axes: xlabel='Sales Channel'>
```



6. Which item types have the highest and lowest total sales?

In [23]:
```
group_item_type= data.groupby(data['Item Type'])['Total Revenue'].sum()

highest_sales_revenue_item_type= group_item_type.idxmax()
lowest_sales_revenue_item_type= group_item_type.idxmin()
```

```
print("{'Highest Sales Revenue By Item Type':", highest_sales_revenue_item_type,
      "\n'Lowest Sales Revenue By Item Type':", lowest_sales_revenue_item_type, "}")
```

```
{'Highest Sales Revenue By Item Type': Cosmetics
'Lowest Sales Revenue By Item Type': Fruits }
```

7. How does the order priority vary across different regions?

In [24]:
```
Diff_regions_by_order_priority= data.groupby(data['Region'])['Order Priority'].value_counts()
Diff_regions_by_order_priority
```

Out[24]:
```
Region                             Order Priority
Asia                               L                  4
                                   H                  3
                                   M                  2
                                   C                  2
Australia and Oceania              H                  5
                                   C                  4
                                   L                  1
                                   M                  1
Central America and the Caribbean  L                  2
                                   H                  2
                                   C                  2
                                   M                  1
Europe                             H                  7
                                   L                  6
                                   C                  5
                                   M                  4
Middle East and North Africa       M                  4
                                   L                  4
                                   H                  2
North America                      C                  1
                                   L                  1
                                   M                  1
Sub-Saharan Africa                 H                 11
                                   L                  9
                                   M                  8
                                   C                  8
Name: count, dtype: int64
```

In [ ]:

8. What is the correlation between unit price and total profit?

In [25]:
```
Correlation_Unit_Price_Total_Profit= data['Unit Price'].corr(data['Total Profit'])

print("Correlation between Unit Price and Total Profit:", Correlation_Unit_Price_Total_Profit)
```

```
Correlation between Unit Price and Total Profit: 0.5573652488121269
```

9. Are there any seasonal trends or patterns in the sales data?

In [26]:
```
month_names= {1: 'JAN',
              2: 'FEB',
              3: 'MAR',
              4: 'APR',
              5: 'MAY',
              6: 'JUN',
              7: 'JUL',
              8: 'AUG',
              9: 'SEPT',
             10: 'OCT',
             11: 'NOV',
             12: 'DEC'}
monthly_sales = data.groupby(data['Order Date'].dt.month)['Total Revenue'].sum()
monthly_sales.index= monthly_sales.index.map(month_names)

monthly_sales
```

Out[26]:
```
Order Date
JAN     10482467.12
FEB     24740517.77
MAR      2274823.87
APR     16187186.33
MAY     13215739.99
JUN      5230325.77
JUL     15669518.50
AUG      1128164.91
SEPT     5314762.56
OCT     15287576.61
NOV     20568222.76
DEC      7249462.12
Name: Total Revenue, dtype: float64
```

In [27]:
```
monthly_sales.plot(kind='bar',xlabel='Months', ylabel='Total Revenue', title ='Month wise Total Revenue',grid='bool')
```

Out[27]:
```
<Axes: title={'center': 'Month wise Total Revenue'}, xlabel='Months', ylabel='Total Revenue'>
```

10. How does the number of units sold vary across different countries?

In [28]:
```
Diff_countries_by_unit_sold= data.groupby(data['Country'])['Units Sold'].sum().reset_index(name= 'Unit Sold')
pd.set_option('display.max_rows',None)
Diff_countries_by_unit_sold
```

Out[28]:

	Country	Unit Sold
0	Albania	2269
1	Angola	4187
2	Australia	12995
3	Austria	2847
4	Azerbaijan	9255
5	Bangladesh	8263
6	Belize	5498
7	Brunei	6708
8	Bulgaria	5660
9	Burkina Faso	8082
10	Cameroon	10948
11	Cape Verde	4168
12	Comoros	962
13	Costa Rica	6409
14	Cote d'Ivoire	3482
15	Democratic Republic of the Congo	5741
16	Djibouti	23198
17	East Timor	5908
18	Federated States of Micronesia	9379
19	Fiji	9905
20	France	1815
21	Gabon	8656

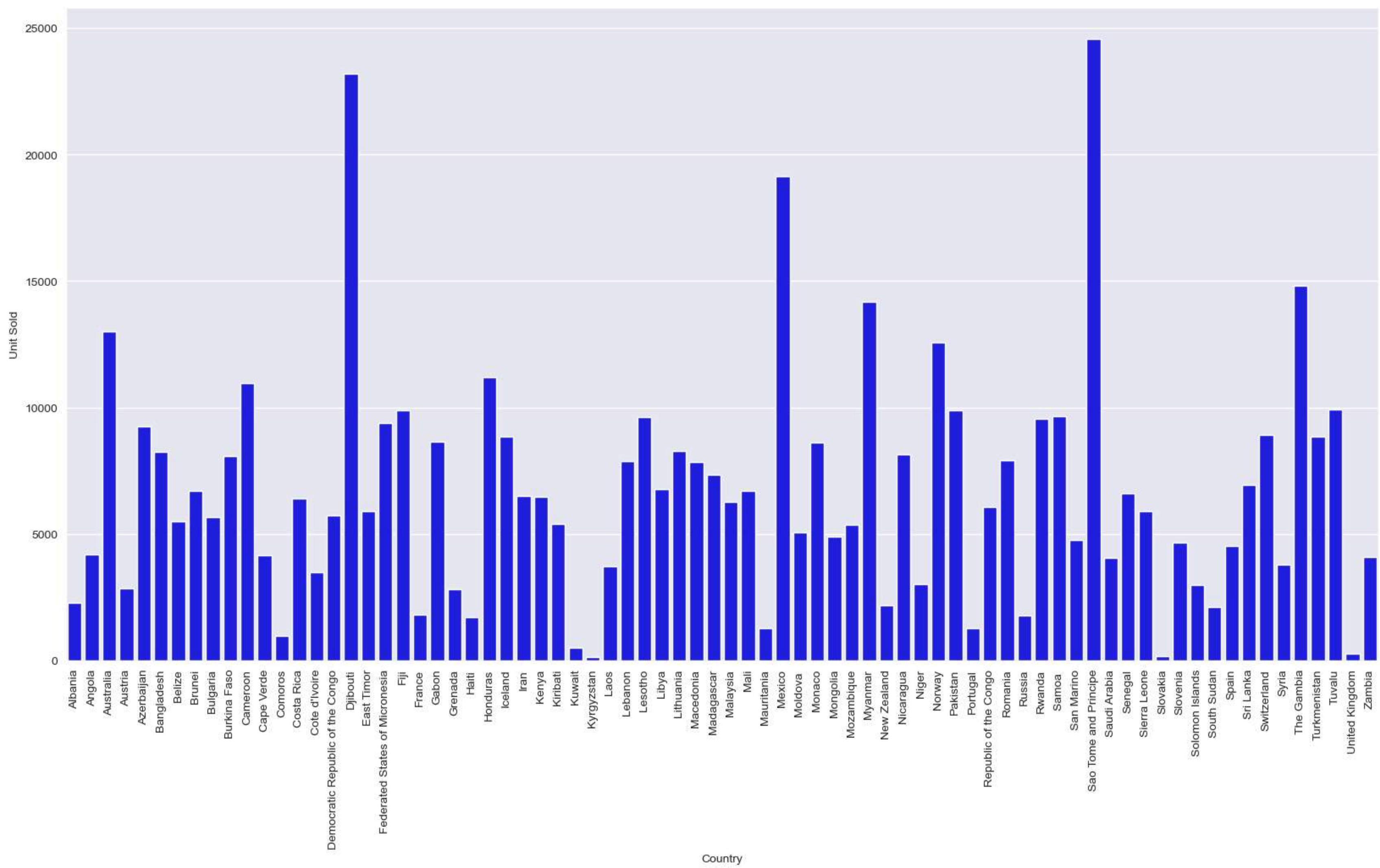
	Country	Unit Sold
22	Grenada	2804
23	Haiti	1705
24	Honduras	11199
25	Iceland	8867
26	Iran	6489
27	Kenya	6457
28	Kiribati	5398
29	Kuwait	522
30	Kyrgyzstan	124
31	Laos	3732
32	Lebanon	7884
33	Lesotho	9606
34	Libya	6789
35	Lithuania	8287
36	Macedonia	7842
37	Madagascar	7342
38	Malaysia	6267
39	Mali	6710
40	Mauritania	1266
41	Mexico	19143
42	Moldova	5070
43	Monaco	8614

	Country	Unit Sold
44	Mongolia	4901
45	Mozambique	5367
46	Myanmar	14180
47	New Zealand	2187
48	Nicaragua	8156
49	Niger	3015
50	Norway	12574
51	Pakistan	9892
52	Portugal	1273
53	Republic of the Congo	6070
54	Romania	7910
55	Russia	1779
56	Rwanda	9539
57	Samoa	9654
58	San Marino	4750
59	Sao Tome and Principe	24568
60	Saudi Arabia	4063
61	Senegal	6593
62	Sierra Leone	5890
63	Slovakia	171
64	Slovenia	4660
65	Solomon Islands	2974

	Country	Unit Sold
66	South Sudan	2125
67	Spain	4513
68	Sri Lanka	6952
69	Switzerland	8934
70	Syria	3784
71	The Gambia	14813
72	Turkmenistan	8840
73	Tuvalu	9925
74	United Kingdom	282
75	Zambia	4085

In [29]:

```
plt.figure(figsize=(20,10))
sns.set_style('darkgrid')
sns.barplot(x= Diff_countries_by_unit_sold['Country'], y= Diff_countries_by_unit_sold['Unit Sold'], color='blue')
plt.xticks(rotation= 90)
plt.show()
```



11. How does the total sales revenue vary across different countries?

In [30]:
```
sales_revenue_by_countries=  data.groupby(data['Country']) ['Total Revenue'].sum().reset_index(name= 'Total Revenue')
sales_revenue_by_countries
```

Out[30]:

	Country	Total Revenue
0	Albania	247956.32
1	Angola	2798046.49
2	Australia	2489933.49
3	Austria	1244708.40
4	Azerbaijan	4478800.21
5	Bangladesh	902980.64
6	Belize	600821.44
7	Brunei	4368316.68
8	Bulgaria	2779199.71
9	Burkina Faso	1245112.92
10	Cameroon	3851030.28
11	Cape Verde	455479.04
12	Comoros	197883.40
13	Costa Rica	523807.57
14	Cote d'Ivoire	380512.96
15	Democratic Republic of the Congo	272410.45
16	Djibouti	6052890.86
17	East Timor	2492526.12
18	Federated States of Micronesia	445033.55
19	Fiji	1082418.40
20	France	793518.00
21	Gabon	707454.88

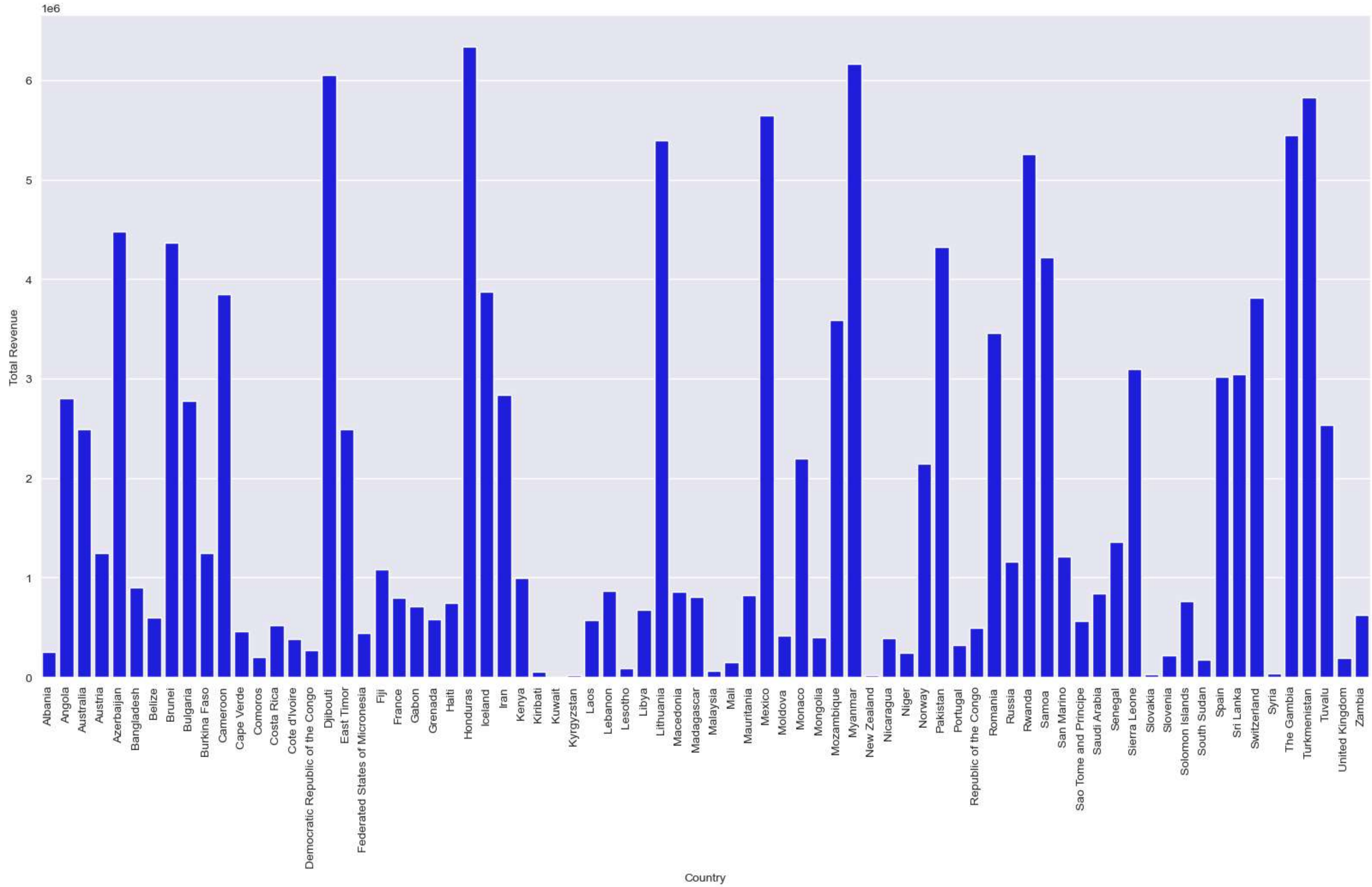
	Country	Total Revenue
22	Grenada	576782.80
23	Haiti	745426.00
24	Honduras	6336545.48
25	Iceland	3876652.40
26	Iran	2836990.80
27	Kenya	994765.42
28	Kiribati	50363.34
29	Kuwait	4870.26
30	Kyrgyzstan	19103.44
31	Laos	574951.92
32	Lebanon	861563.52
33	Lesotho	89623.98
34	Libya	674635.57
35	Lithuania	5396577.27
36	Macedonia	856973.76
37	Madagascar	802333.76
38	Malaysia	58471.11
39	Mali	151359.90
40	Mauritania	824431.86
41	Mexico	5643356.55
42	Moldova	414371.10
43	Monaco	2198981.92

	Country	Total Revenue
44	Mongolia	400558.73
45	Mozambique	3586605.09
46	Myanmar	6161257.90
47	New Zealand	20404.71
48	Nicaragua	387002.20
49	Niger	246415.95
50	Norway	2144969.80
51	Pakistan	4324782.40
52	Portugal	324971.44
53	Republic of the Congo	496101.10
54	Romania	3458252.00
55	Russia	1158502.59
56	Rwanda	5253769.42
57	Samoa	4220728.80
58	San Marino	1212580.00
59	Sao Tome and Principe	565780.92
60	Saudi Arabia	835759.10
61	Senegal	1356180.10
62	Sierra Leone	3097359.15
63	Slovakia	26344.26
64	Slovenia	221117.00
65	Solomon Islands	759202.72

	Country	Total Revenue
66	South Sudan	173676.25
67	Spain	3015902.51
68	Sri Lanka	3039414.40
69	Switzerland	3808901.49
70	Syria	35304.72
71	The Gambia	5449517.95
72	Turkmenistan	5822036.20
73	Tuvalu	2533654.00
74	United Kingdom	188452.14
75	Zambia	623289.30

In [31]:

```
plt.figure(figsize=(20,10))
sns.set_style('darkgrid')
sns.barplot(x= sales_revenue_by_countries['Country'], y= sales_revenue_by_countries['Total Revenue'], color='blue')
plt.xticks(rotation= 90)
plt.show()
```



12. What is the distribution of unit prices for each item type?

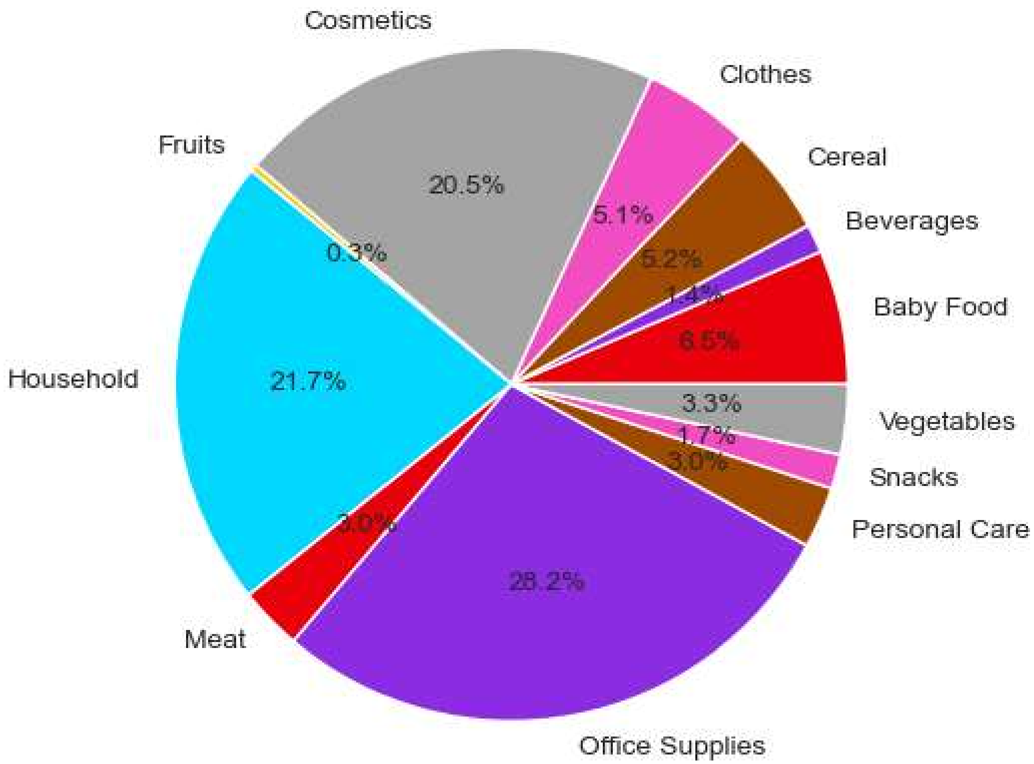
In [32]:
```
unit_price_and_item_type_distribution= data.groupby(data['Item Type'])['Unit Price'].sum().reset_index(name= 'Unit Pr
unit_price_and_item_type_distribution
```

Out[32]:

	Item Type	Unit Price
0	Baby Food	1786.96
1	Beverages	379.60
2	Cereal	1439.90
3	Clothes	1420.64
4	Cosmetics	5683.60
5	Fruits	93.30
6	Household	6014.43
7	Meat	843.78
8	Office Supplies	7814.52
9	Personal Care	817.30
10	Snacks	457.74
11	Vegetables	924.36

In [33]:

```
colors = sns.color_palette('bright')[3:16]
plt.pie(x= unit_price_and_item_type_distribution['Unit Price'],
        labels= unit_price_and_item_type_distribution['Item Type'],autopct='%1.1f%%',colors=colors)
plt.axis('equal')
plt.show()
```



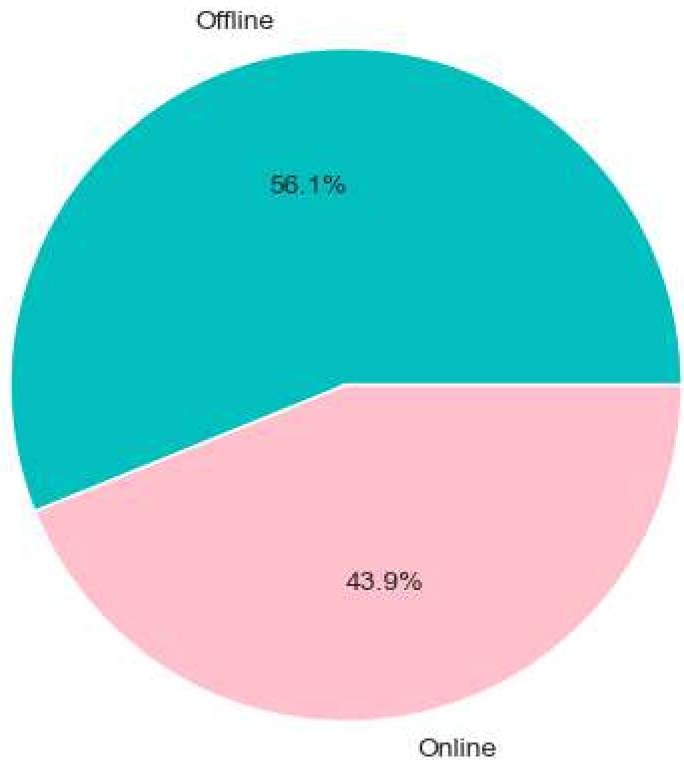
13. Which sales channel has the highest average unit price?

In [34]:
```
Highest_avg_unit_price_for_sales_channel= data.groupby(data['Sales Channel']) ['Unit Price'].mean().reset_index(name=
Highest_avg_unit_price_for_sales_channel
```

Out[34]:

	Sales Channel	new
0	Offline	310.7206
1	Online	242.8020

In [35]:
```
colors =['c', 'pink']
plt.pie(x= Highest_avg_unit_price_for_sales_channel['new'],labels=Highest_avg_unit_price_for_sales_channel['Sales Cha
plt.axis('equal')
plt.show()
```



14. Are there any outliers in the total cost distribution?

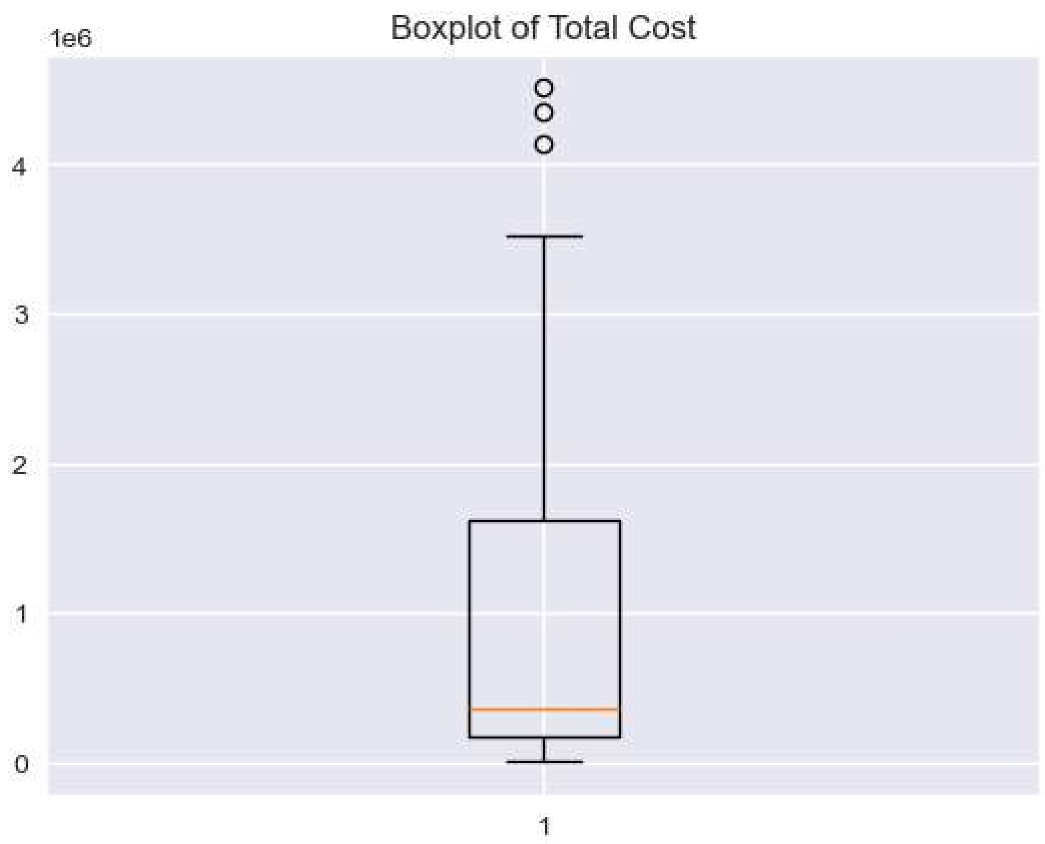
In [36]:
```
q1= data['Total Cost'].quantile(0.25)
q3= data['Total Cost'].quantile(0.75)

iqr= q3-q1

lower_fence= q1-1.5*iqr
upper_fence= q3+1.5*iqr

outliers= data[(data['Total Cost']<lower_fence)| (data['Total Cost']>upper_fence)].reset_index(drop= True)
outliers
```

Out[36]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost
0	Central America and the Caribbean	Honduras	Household	Offline	H	2017-02-08	522840487	2017-02-13	8974	668.27	502.54	5997054.98	4509793.96
1	Asia	Myanmar	Household	Offline	H	2015-01-16	177713572	2015-03-01	8250	668.27	502.54	5513227.50	4145955.0
2	Europe	Lithuania	Office Supplies	Offline	H	2010-10-24	166460740	2010-11-17	8287	651.21	524.96	5396577.27	4350343.52

In [37]:

```
sns.set_style('darkgrid')
plt.boxplot(data['Total Cost'])
plt.title('Boxplot of Total Cost')
plt.show()
```



15. How does the total profit vary across different item types?

In [38]:
```
total_profit_and_diff_item_types= data.groupby(data['Item Type'])['Total Profit'].sum().reset_index(name='Total Profi
total_profit_and_diff_item_types
```

Out[38]:

	Item Type	Total Profit
0	Baby Food	3886643.70
1	Beverages	888047.28
2	Cereal	2292443.43
3	Clothes	5233334.40
4	Cosmetics	14556048.66
5	Fruits	120495.18
6	Household	7412605.71
7	Meat	610610.00
8	Office Supplies	5929583.75
9	Personal Care	1220622.48
10	Snacks	751944.18
11	Vegetables	1265819.63

16. What is the average order processing time for each country?

In [39]:

```
Avg_Processing_Time_by_country= data.groupby(data['Country'])['Processing Time'].mean()
Avg_Processing_Time_by_country
```

```
Out[39]: Country
Albania                               44 days 00:00:00
Angola                                 4 days 00:00:00
Australia                             18 days 16:00:00
Austria                                7 days 00:00:00
Azerbaijan                            30 days 00:00:00
Bangladesh                            47 days 00:00:00
Belize                                44 days 00:00:00
Brunei                                37 days 00:00:00
Bulgaria                              26 days 12:00:00
Burkina Faso                          10 days 00:00:00
Cameroon                              12 days 12:00:00
Cape Verde                            17 days 00:00:00
Comoros                               31 days 00:00:00
Costa Rica                            13 days 00:00:00
Cote d'Ivoire                         19 days 00:00:00
Democratic Republic of the Congo      50 days 00:00:00
Djibouti                              13 days 08:00:00
East Timor                            42 days 00:00:00
Federated States of Micronesia        18 days 00:00:00
Fiji                                  32 days 00:00:00
France                                14 days 00:00:00
Gabon                                  1 days 00:00:00
Grenada                               24 days 00:00:00
Haiti                                 34 days 00:00:00
Honduras                              15 days 12:00:00
Iceland                                0 days 00:00:00
Iran                                  23 days 00:00:00
Kenya                                 20 days 00:00:00
Kiribati                              28 days 00:00:00
Kuwait                                18 days 00:00:00
Kyrgyzstan                            18 days 00:00:00
Laos                                  38 days 00:00:00
Lebanon                               20 days 00:00:00
Lesotho                               31 days 00:00:00
Libya                                 32 days 12:00:00
Lithuania                             24 days 00:00:00
Macedonia                             31 days 00:00:00
Madagascar                            33 days 00:00:00
Malaysia                              47 days 00:00:00
Mali                                  21 days 00:00:00
Mauritania                             2 days 00:00:00
```

```
Mexico                        25 days 16:00:00
Moldova                        3 days 00:00:00
Monaco                         4 days 00:00:00
Mongolia                       4 days 00:00:00
Mozambique                     5 days 00:00:00
Myanmar                       24 days 00:00:00
New Zealand                   26 days 00:00:00
Nicaragua                     41 days 00:00:00
Niger                         17 days 00:00:00
Norway                        28 days 12:00:00
Pakistan                      42 days 00:00:00
Portugal                      34 days 00:00:00
Republic of the Congo         42 days 00:00:00
Romania                       29 days 00:00:00
Russia                         6 days 00:00:00
Rwanda                        25 days 00:00:00
Samoa                         18 days 00:00:00
San Marino                     5 days 00:00:00
Sao Tome and Principe         19 days 00:00:00
Saudi Arabia                   3 days 00:00:00
Senegal                       42 days 00:00:00
Sierra Leone                  26 days 00:00:00
Slovakia                      35 days 00:00:00
Slovenia                      33 days 00:00:00
Solomon Islands               17 days 00:00:00
South Sudan                   30 days 00:00:00
Spain                         40 days 00:00:00
Sri Lanka                     29 days 00:00:00
Switzerland                   36 days 00:00:00
Syria                         11 days 00:00:00
The Gambia                    17 days 06:00:00
Turkmenistan                  24 days 00:00:00
Tuvalu                        30 days 00:00:00
United Kingdom                40 days 00:00:00
Zambia                         1 days 00:00:00
Name: Processing Time, dtype: timedelta64[ns]
```

17. Which region has the highest average total revenue per order?

In [40]:

```
data['avg total revenue']= data['Total Revenue']/data['Units Sold']
highest_avg_total_revenue_per_order= data.groupby(data['Region']) ['avg total revenue'].mean()
```

```
highest_avg_total_revenue_per_order.sort_values(ascending=True)
highest_avg_total_revenue_per_order.head(1)
```

Out[40]:

```
Region
Asia    335.809091
Name: avg total revenue, dtype: float64
```

18. Is there a relationship between the number of units sold and the total profit?

In [41]:

```
Correlation_unit_sold_and_total_profit= data['Units Sold'].corr(data['Total Profit'])
print(f"Correlation coefficient: {Correlation_unit_sold_and_total_profit}")
```

```
Correlation coefficient: 0.5645504620845977
```

19. How does the order priority vary based on the item type?

In [42]:

```
Order_priority_vary_on_item_type= data.groupby(data['Order Priority'])['Item Type'].value_counts().reset_index(name=
Order_priority_vary_on_item_type
```

Out[42]:

	Order Priority	Item Type	No. Of Items
0	C	Beverages	7
1	C	Clothes	4
2	C	Vegetables	2
3	C	Personal Care	2
4	C	Office Supplies	2
5	C	Baby Food	1
6	C	Household	1
7	C	Fruits	1
8	C	Cosmetics	1
9	C	Cereal	1
10	H	Cosmetics	8
11	H	Cereal	5
12	H	Baby Food	3
13	H	Clothes	3
14	H	Vegetables	3
15	H	Office Supplies	2
16	H	Household	2
17	H	Fruits	2
18	H	Personal Care	1
19	H	Beverages	1
20	L	Household	5
21	L	Fruits	5

	Order Priority	Item Type	No. Of Items
22	L	Personal Care	4
23	L	Office Supplies	3
24	L	Clothes	3
25	L	Baby Food	2
26	L	Snacks	2
27	L	Vegetables	1
28	L	Meat	1
29	L	Cosmetics	1
30	M	Office Supplies	5
31	M	Clothes	3
32	M	Personal Care	3
33	M	Cosmetics	3
34	M	Fruits	2
35	M	Baby Food	1
36	M	Cereal	1
37	M	Household	1
38	M	Meat	1
39	M	Snacks	1

20. Are there any trends or patterns in the order dates?

In [43]:

```
# Extract year, month, day, or weekday
data['Year'] = data['Order Date'].dt.year
data['Month'] = data['Order Date'].dt.month
```

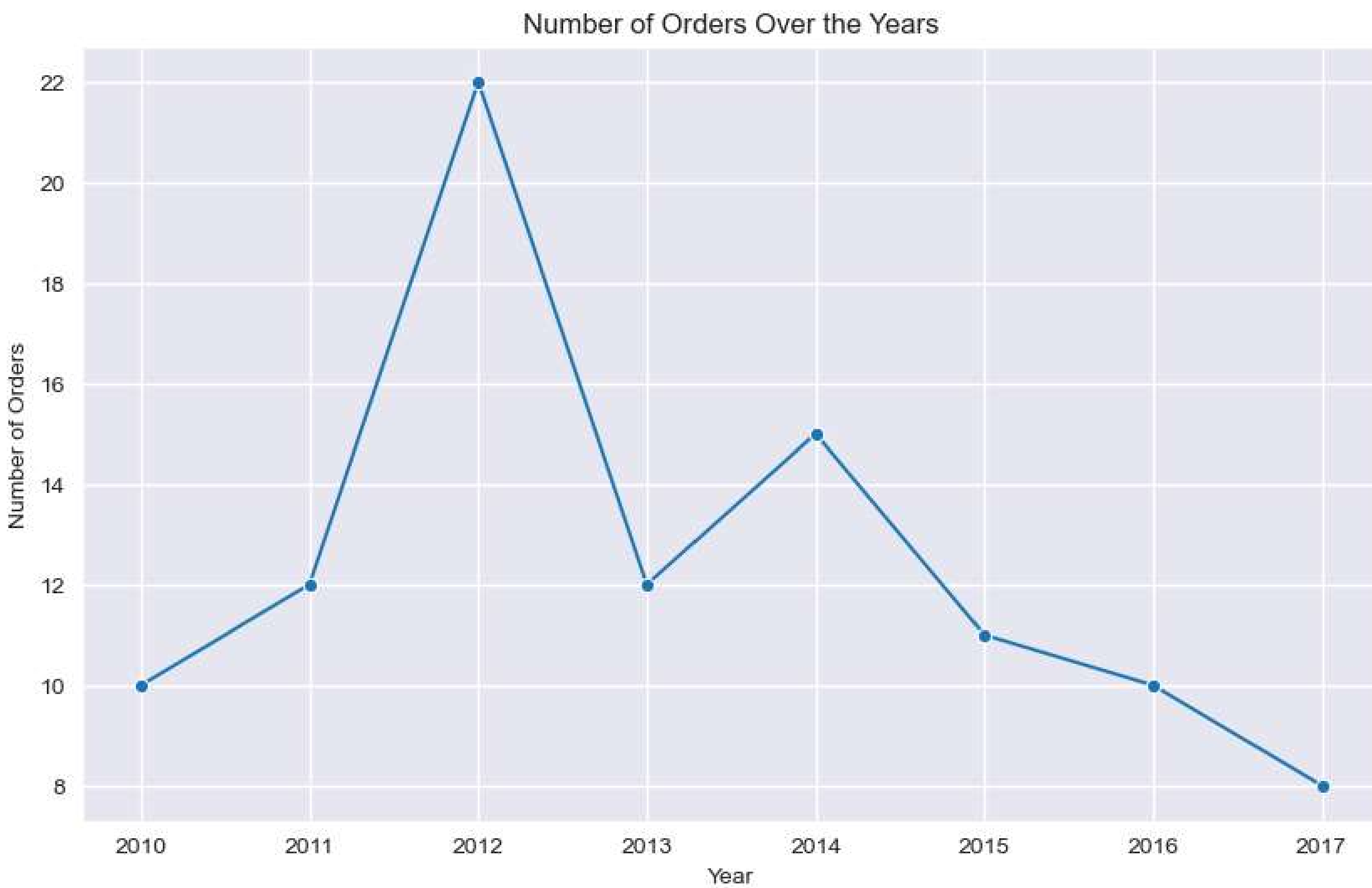
```python
data['Day'] = data['Order Date'].dt.day
data['Weekday'] = data['Order Date'].dt.weekday
```

In [44]:
```python
# Group by year and count the number of orders
orders_by_year = data.groupby('Year').size()

# Group by month to see monthly trends
orders_by_month = data.groupby('Month').size()

print(orders_by_year)
print(orders_by_month)
```

```
Year
2010    10
2011    12
2012    22
2013    12
2014    15
2015    11
2016    10
2017     8
dtype: int64
Month
1      7
2     13
3      4
4      9
5     11
6     10
7     12
8      4
9      5
10    11
11     9
12     5
dtype: int64
```

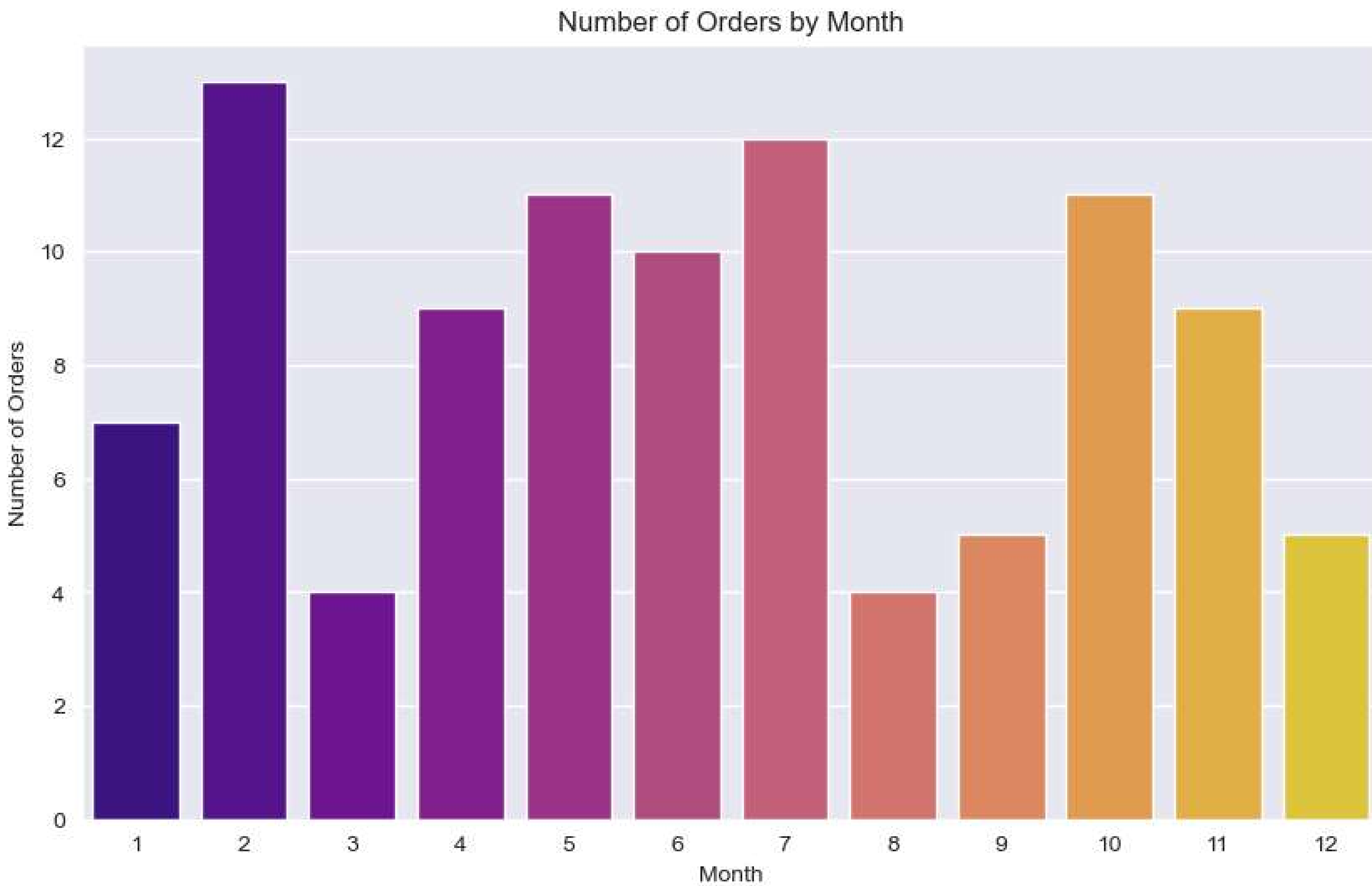
In [45]:
```python
# Group orders by year
orders_by_year = data.groupby(data['Order Date'].dt.year)['Order ID'].count()

# Plot the yearly trend
plt.figure(figsize=(10,6))
```

```
sns.set_style('darkgrid')
sns.lineplot(x=orders_by_year.index, y=orders_by_year.values, marker='o')
plt.title('Number of Orders Over the Years')
plt.xlabel('Year')
plt.ylabel('Number of Orders')
plt.show()
```



In [46]:
```
# Group by month
orders_by_month = data.groupby(data['Order Date'].dt.month)['Order ID'].count()
```

```
# Plot the monthly trend
plt.figure(figsize=(10,6))
sns.set_style('darkgrid')
sns.barplot(x=orders_by_month.index, y=orders_by_month.values, palette='plasma')
plt.title('Number of Orders by Month')
plt.xlabel('Month')
plt.ylabel('Number of Orders')
plt.show()
```



In [ ]: