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

In [ ]:
```
data= pd.read_csv('Flipkart Sales data.csv')
data= pd.DataFrame(data= data)
data
```

Out[ ]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cos
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	5/28/2010	669165933	6/27/2010	9925	255.28	159.42	2533654.00	1582243.5
1	Central America and the Caribbean	Grenada	Cereal	Online	C	8/22/2012	963881480	9/15/2012	2804	205.70	117.11	576782.80	328376.4
2	Europe	Russia	Office Supplies	Offline	L	05-02-2014	341417157	05-08-2014	1779	651.21	524.96	1158502.59	933903.8
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	56065.8
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	02-01-2013	115456712	02-06-2013	5062	651.21	524.96	3296425.02	2657347.5
...	...	...	...	...	...	...	...	...	...	...	...	...	.
95	Sub-Saharan Africa	Mali	Clothes	Online	M	7/26/2011	512878119	09-03-2011	888	109.28	35.84	97040.64	31825.9
96	Asia	Malaysia	Fruits	Offline	L	11-11-2011	810711038	12/28/2011	6267	9.33	6.92	58471.11	43367.6
97	Sub-Saharan Africa	Sierra Leone	Vegetables	Offline	C	06-01-2016	728815257	6/29/2016	1485	154.06	90.93	228779.10	135031.0
98	North America	Mexico	Personal Care	Offline	M	7/30/2015	559427106	08-08-2015	5767	81.73	56.67	471336.91	326815.8
99	Sub-Saharan Africa	Mozambique	Household	Offline	L	02-10-2012	665095412	2/15/2012	5367	668.27	502.54	3586605.09	2697132.1

100 rows × 14 columns

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

Out[ ]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	To Pr
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	5/28/2010	669165933	6/27/2010	9925	255.28	159.42	2533654.00	1582243.50	95141(
1	Central America and the Caribbean	Grenada	Cereal	Online	C	8/22/2012	963881480	9/15/2012	2804	205.70	117.11	576782.80	328376.44	24840(
2	Europe	Russia	Office Supplies	Offline	L	05-02-2014	341417157	05-08-2014	1779	651.21	524.96	1158502.59	933903.84	224598
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	56065.84	1952:
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	02-01-2013	115456712	02-06-2013	5062	651.21	524.96	3296425.02	2657347.52	639071

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

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

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

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

Out[ ]: 1400

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

Out[ ]:

	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 [ ]:
```
data.isna().sum()
```

Out[ ]:
```
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 [ ]:
```
data.dtypes
```

Out[ ]:
```
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 [ ]:
```
data = data.astype({'Ship Date': 'datetime64[ns]','Order Date':'datetime64[ns]'})
```

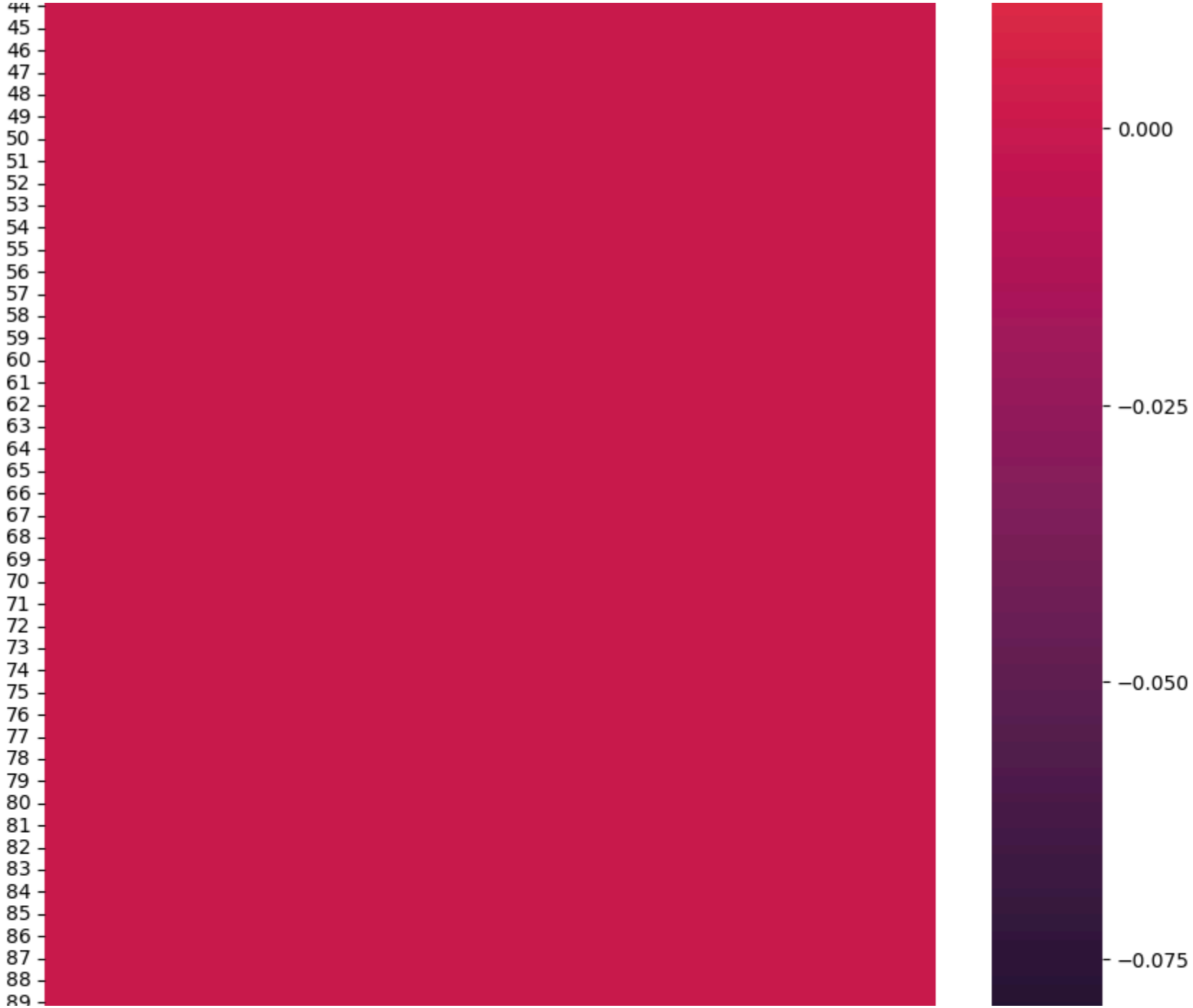
In [ ]:
```
data.dtypes
```

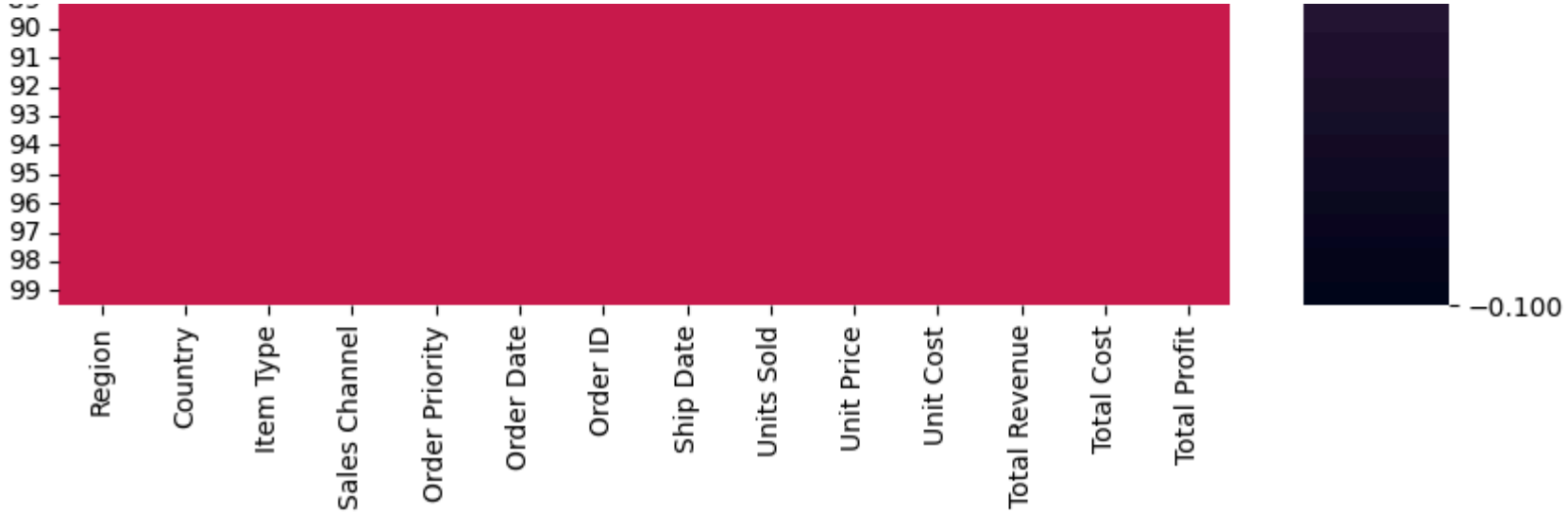
Out[ ]:
```
Region                    object
Country                   object
Item Type                 object
Sales Channel             object
Order Priority            object
Order Date        datetime64[ns]
Order ID                   int64
Ship Date         datetime64[ns]
Units Sold              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

In [ ]:
```
plt.figure(figsize=(10,20))
sns.heatmap(data.isnull()) # NO ANY NULL VALUE PRESENT IN OUR DATASET.
```

Out[ ]:
```
<AxesSubplot: >
```


0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
0.100
0.075
0.050
0.025


44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
0.000
−0.025
−0.050
−0.075

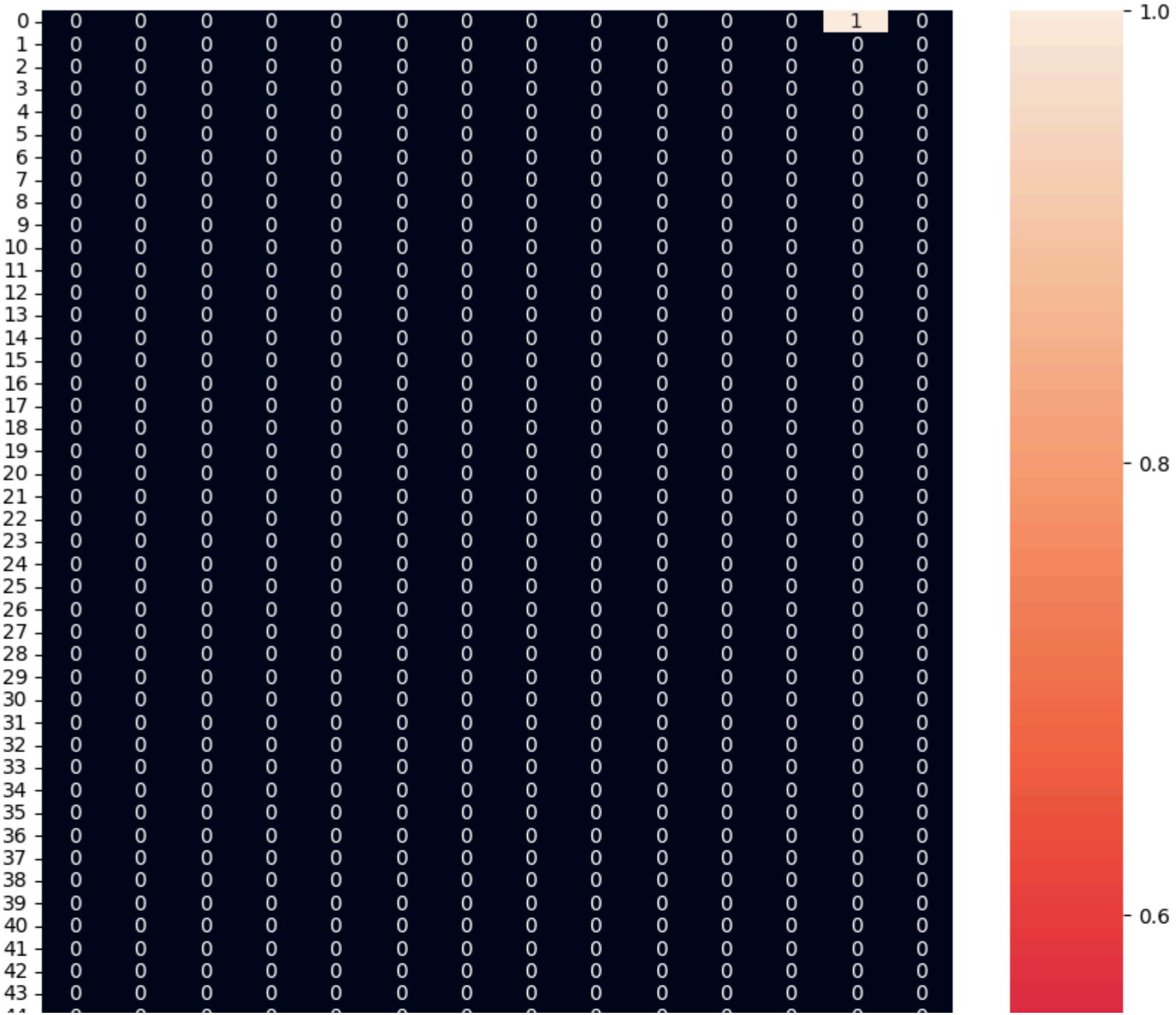


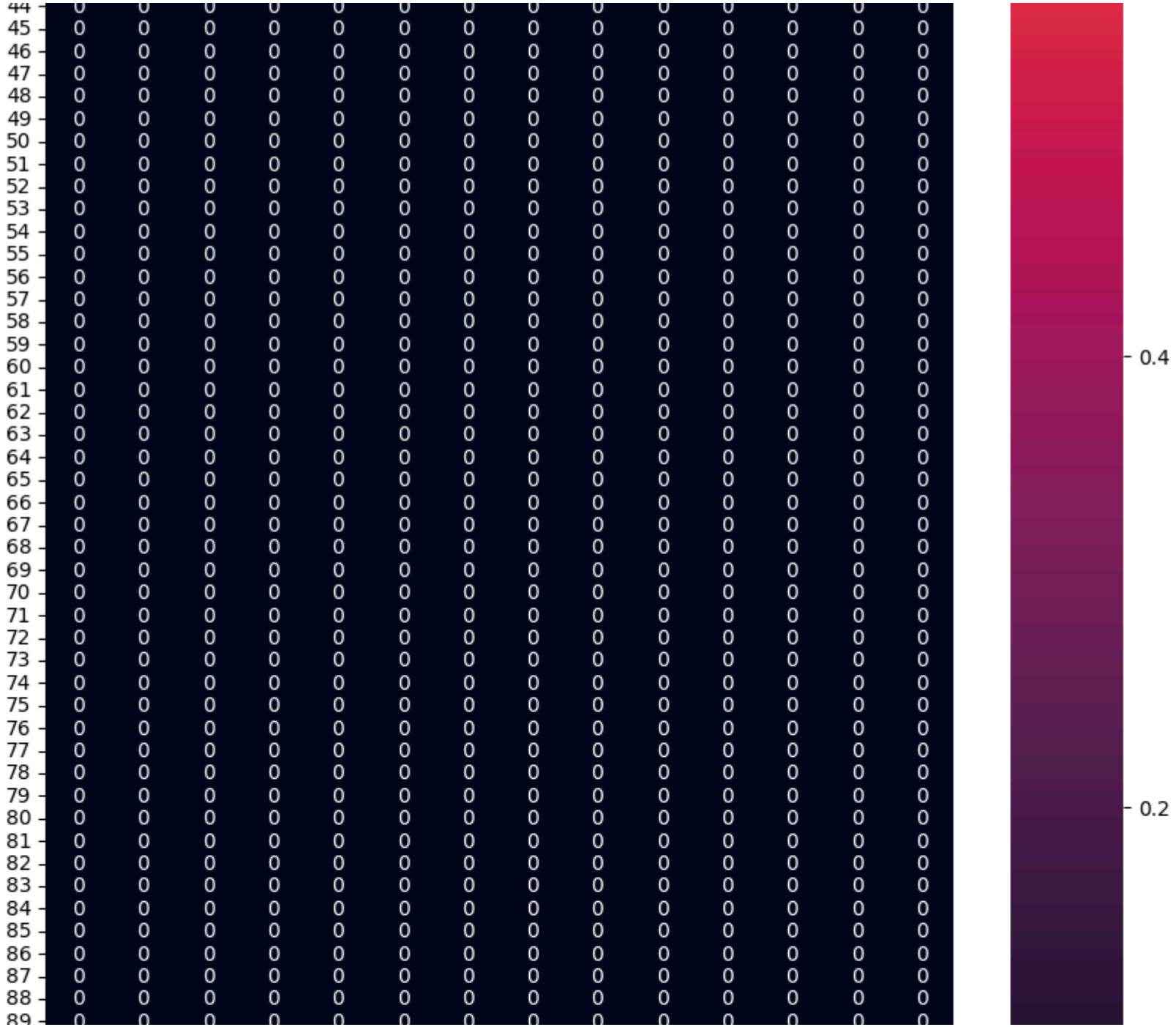
In [ ]:
```
test = data.iloc[0, 12] = np.nan   # ADDING NULL VALUE JUST FOR DEMO
test
```

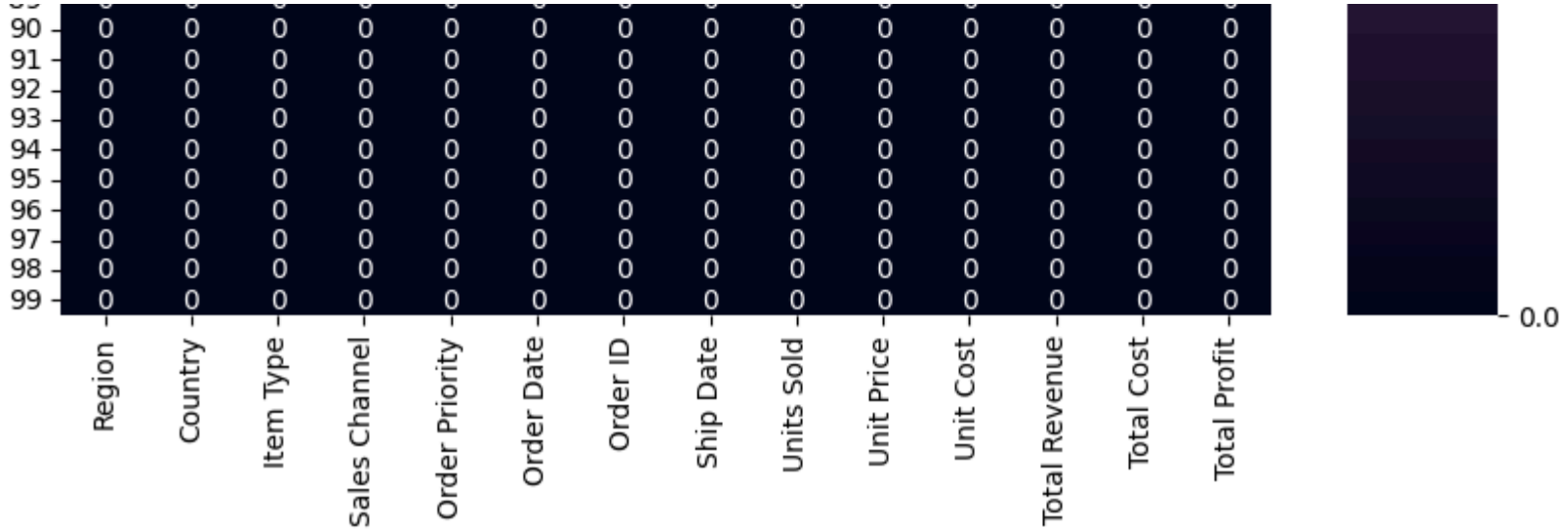
Out[ ]: nan

In [ ]:
```
plt.figure(figsize=(10,20))
sns.heatmap(data.isnull(),annot= True) #NULL VALUE FOUND IN 'TOTAL COST' COLUMN
```

Out[ ]: <AxesSubplot: >


0 0 0 0 0 0 0 0 0 0 0 0 1 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0
2 0 0 0 0 0 0 0 0 0 0 0 0 0
3 0 0 0 0 0 0 0 0 0 0 0 0 0
4 0 0 0 0 0 0 0 0 0 0 0 0 0
5 0 0 0 0 0 0 0 0 0 0 0 0 0
6 0 0 0 0 0 0 0 0 0 0 0 0 0
7 0 0 0 0 0 0 0 0 0 0 0 0 0
8 0 0 0 0 0 0 0 0 0 0 0 0 0
9 0 0 0 0 0 0 0 0 0 0 0 0 0
10 0 0 0 0 0 0 0 0 0 0 0 0 0
11 0 0 0 0 0 0 0 0 0 0 0 0 0
12 0 0 0 0 0 0 0 0 0 0 0 0 0
13 0 0 0 0 0 0 0 0 0 0 0 0 0
14 0 0 0 0 0 0 0 0 0 0 0 0 0
15 0 0 0 0 0 0 0 0 0 0 0 0 0
16 0 0 0 0 0 0 0 0 0 0 0 0 0
17 0 0 0 0 0 0 0 0 0 0 0 0 0
18 0 0 0 0 0 0 0 0 0 0 0 0 0
19 0 0 0 0 0 0 0 0 0 0 0 0 0
20 0 0 0 0 0 0 0 0 0 0 0 0 0
21 0 0 0 0 0 0 0 0 0 0 0 0 0
22 0 0 0 0 0 0 0 0 0 0 0 0 0
23 0 0 0 0 0 0 0 0 0 0 0 0 0
24 0 0 0 0 0 0 0 0 0 0 0 0 0
25 0 0 0 0 0 0 0 0 0 0 0 0 0
26 0 0 0 0 0 0 0 0 0 0 0 0 0
27 0 0 0 0 0 0 0 0 0 0 0 0 0
28 0 0 0 0 0 0 0 0 0 0 0 0 0
29 0 0 0 0 0 0 0 0 0 0 0 0 0
30 0 0 0 0 0 0 0 0 0 0 0 0 0
31 0 0 0 0 0 0 0 0 0 0 0 0 0
32 0 0 0 0 0 0 0 0 0 0 0 0 0
33 0 0 0 0 0 0 0 0 0 0 0 0 0
34 0 0 0 0 0 0 0 0 0 0 0 0 0
35 0 0 0 0 0 0 0 0 0 0 0 0 0
36 0 0 0 0 0 0 0 0 0 0 0 0 0
37 0 0 0 0 0 0 0 0 0 0 0 0 0
38 0 0 0 0 0 0 0 0 0 0 0 0 0
39 0 0 0 0 0 0 0 0 0 0 0 0 0
40 0 0 0 0 0 0 0 0 0 0 0 0 0
41 0 0 0 0 0 0 0 0 0 0 0 0 0
42 0 0 0 0 0 0 0 0 0 0 0 0 0
43 0 0 0 0 0 0 0 0 0 0 0 0 0
1.0
0.8
0.6


44 0 0 0 0 0 0 0 0 0 0 0 0 0
45 0 0 0 0 0 0 0 0 0 0 0 0 0
46 0 0 0 0 0 0 0 0 0 0 0 0 0
47 0 0 0 0 0 0 0 0 0 0 0 0 0
48 0 0 0 0 0 0 0 0 0 0 0 0 0
48 0 0 0 0 0 0 0 0 0 0 0 0 0
49 0 0 0 0 0 0 0 0 0 0 0 0 0
50 0 0 0 0 0 0 0 0 0 0 0 0 0
51 0 0 0 0 0 0 0 0 0 0 0 0 0
52 0 0 0 0 0 0 0 0 0 0 0 0 0
53 0 0 0 0 0 0 0 0 0 0 0 0 0
54 0 0 0 0 0 0 0 0 0 0 0 0 0
55 0 0 0 0 0 0 0 0 0 0 0 0 0
56 0 0 0 0 0 0 0 0 0 0 0 0 0
57 0 0 0 0 0 0 0 0 0 0 0 0 0
58 0 0 0 0 0 0 0 0 0 0 0 0 0
59 0 0 0 0 0 0 0 0 0 0 0 0 0
60 0 0 0 0 0 0 0 0 0 0 0 0 0
61 0 0 0 0 0 0 0 0 0 0 0 0 0
62 0 0 0 0 0 0 0 0 0 0 0 0 0
63 0 0 0 0 0 0 0 0 0 0 0 0 0
64 0 0 0 0 0 0 0 0 0 0 0 0 0
65 0 0 0 0 0 0 0 0 0 0 0 0 0
66 0 0 0 0 0 0 0 0 0 0 0 0 0
67 0 0 0 0 0 0 0 0 0 0 0 0 0
68 0 0 0 0 0 0 0 0 0 0 0 0 0
69 0 0 0 0 0 0 0 0 0 0 0 0 0
70 0 0 0 0 0 0 0 0 0 0 0 0 0
71 0 0 0 0 0 0 0 0 0 0 0 0 0
72 0 0 0 0 0 0 0 0 0 0 0 0 0
73 0 0 0 0 0 0 0 0 0 0 0 0 0
74 0 0 0 0 0 0 0 0 0 0 0 0 0
75 0 0 0 0 0 0 0 0 0 0 0 0 0
76 0 0 0 0 0 0 0 0 0 0 0 0 0
77 0 0 0 0 0 0 0 0 0 0 0 0 0
78 0 0 0 0 0 0 0 0 0 0 0 0 0
79 0 0 0 0 0 0 0 0 0 0 0 0 0
80 0 0 0 0 0 0 0 0 0 0 0 0 0
81 0 0 0 0 0 0 0 0 0 0 0 0 0
82 0 0 0 0 0 0 0 0 0 0 0 0 0
83 0 0 0 0 0 0 0 0 0 0 0 0 0
84 0 0 0 0 0 0 0 0 0 0 0 0 0
85 0 0 0 0 0 0 0 0 0 0 0 0 0
86 0 0 0 0 0 0 0 0 0 0 0 0 0
87 0 0 0 0 0 0 0 0 0 0 0 0 0
88 0 0 0 0 0 0 0 0 0 0 0 0 0
89 0 0 0 0 0 0 0 0 0 0 0 0 0
0.4
0.2



In [ ]:
```
data = data.fillna(data.mean())   #FILL MEAN WHERE NULL VALUE PRESENT
```

```
C:\Users\shory\AppData\Local\Temp\ipykernel_3168\1872784004.py:1: FutureWarning: DataFrame.mean and DataFrame.median with numer
ic_only=None will include datetime64 and datetime64tz columns in a future version.
  data = data.fillna(data.mean())   #FILL MEAN WHERE NULL VALUE PRESENT
C:\Users\shory\AppData\Local\Temp\ipykernel_3168\1872784004.py:1: FutureWarning: The default value of numeric_only in DataFram
e.mean is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this warning.
  data = data.fillna(data.mean())   #FILL MEAN WHERE NULL VALUE PRESENT
```

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

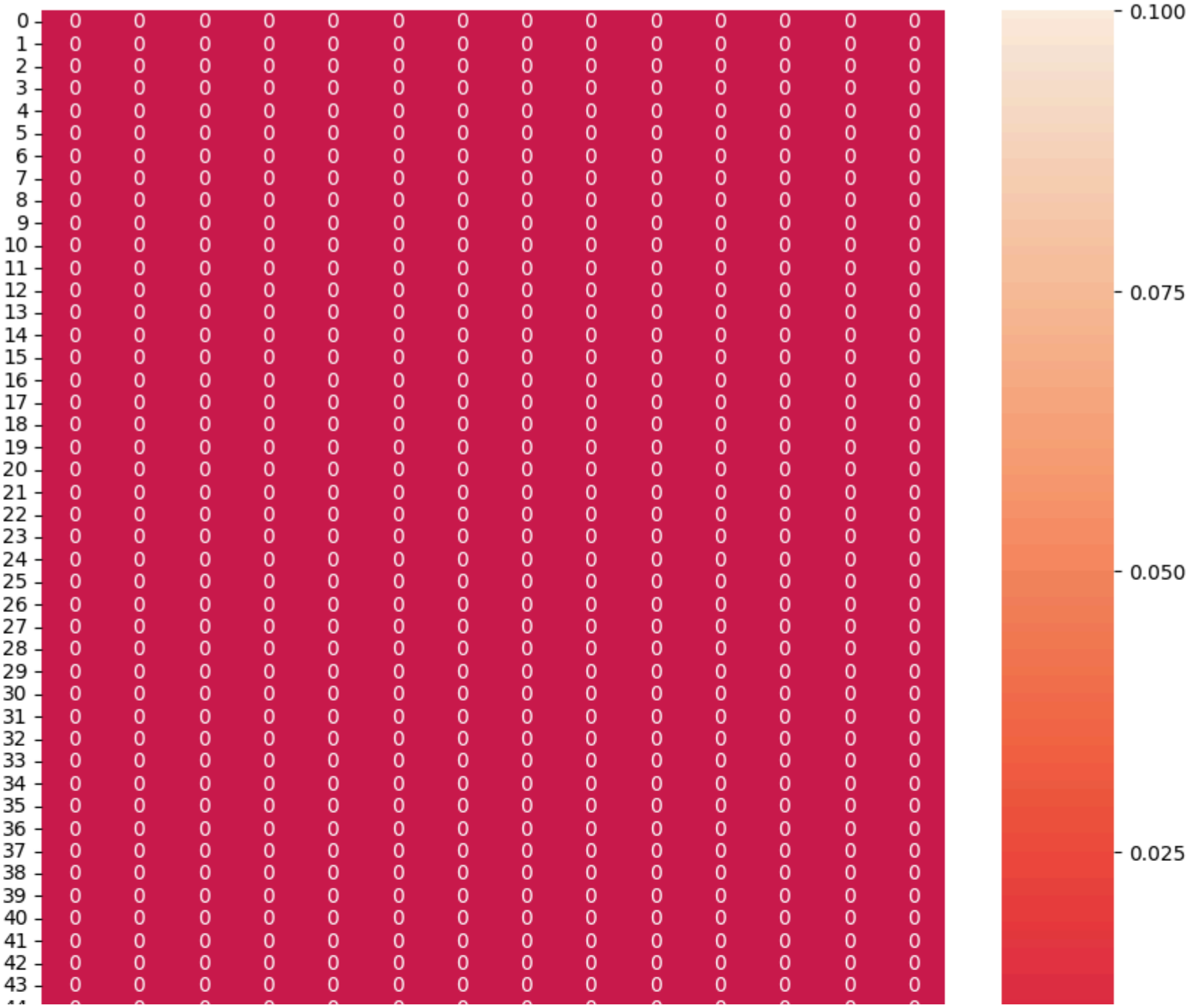
Out[ ]:

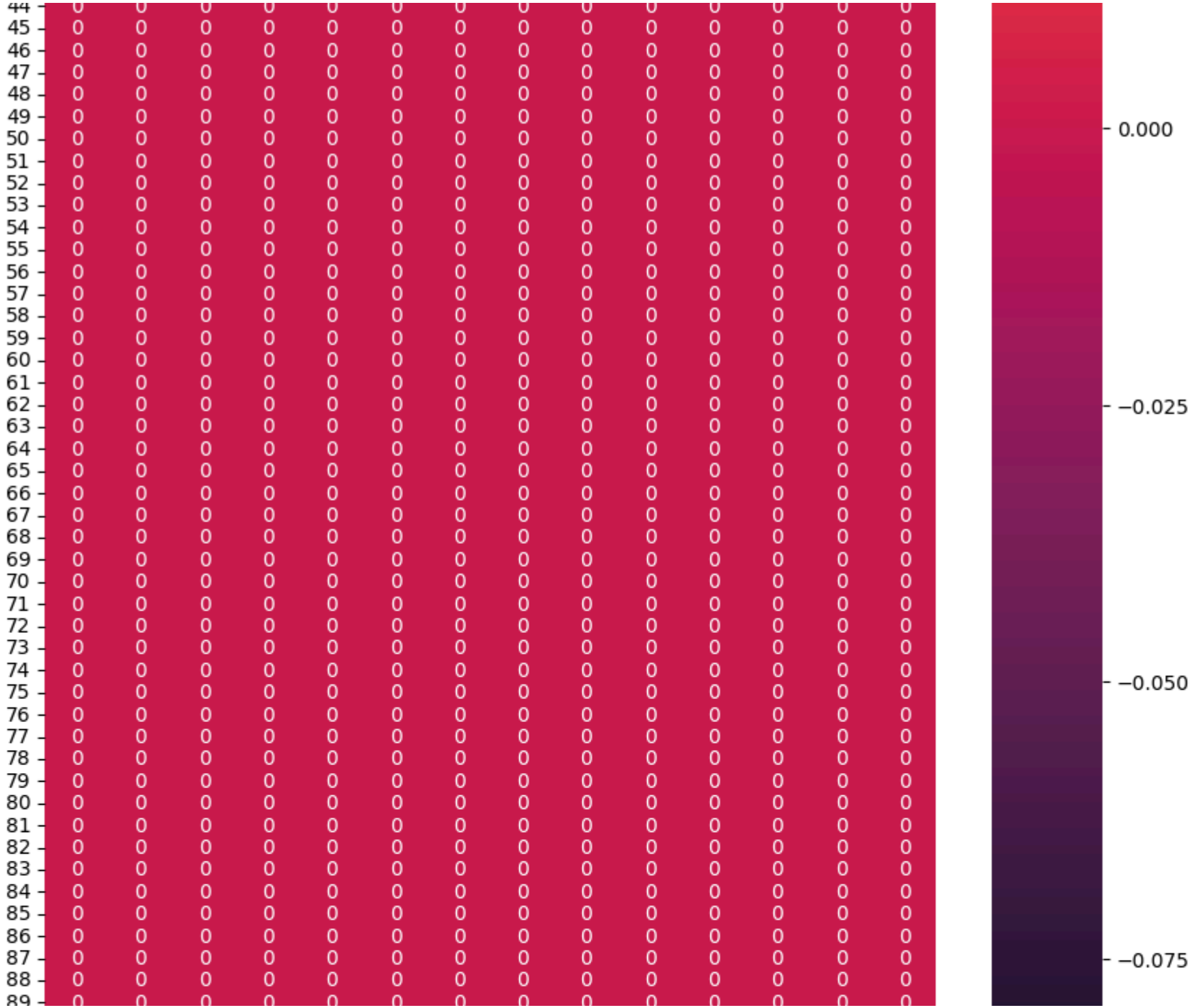
	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	2010-05-28	669165933	2010-06-27	9925	255.28	159.42	2533654.00	925235.620303	951
1	Central America and the Caribbean	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
95	Sub-Saharan Africa	Mali	Clothes	Online	M	2011-07-26	512878119	2011-09-03	888	109.28	35.84	97040.64	31825.92	65
96	Asia	Malaysia	Fruits	Offline	L	2011-11-11	810711038	2011-12-28	6267	9.33	6.92	58471.11	43367.64	15
97	Sub-Saharan Africa	Sierra Leone	Vegetables	Offline	C	2016-06-01	728815257	2016-06-29	1485	154.06	90.93	228779.10	135031.05	93
98	North America	Mexico	Personal Care	Offline	M	2015-07-30	559427106	2015-08-08	5767	81.73	56.67	471336.91	326815.89	144
99	Sub-Saharan Africa	Mozambique	Household	Offline	L	2012-02-10	665095412	2012-02-15	5367	668.27	502.54	3586605.09	2697132.18	889

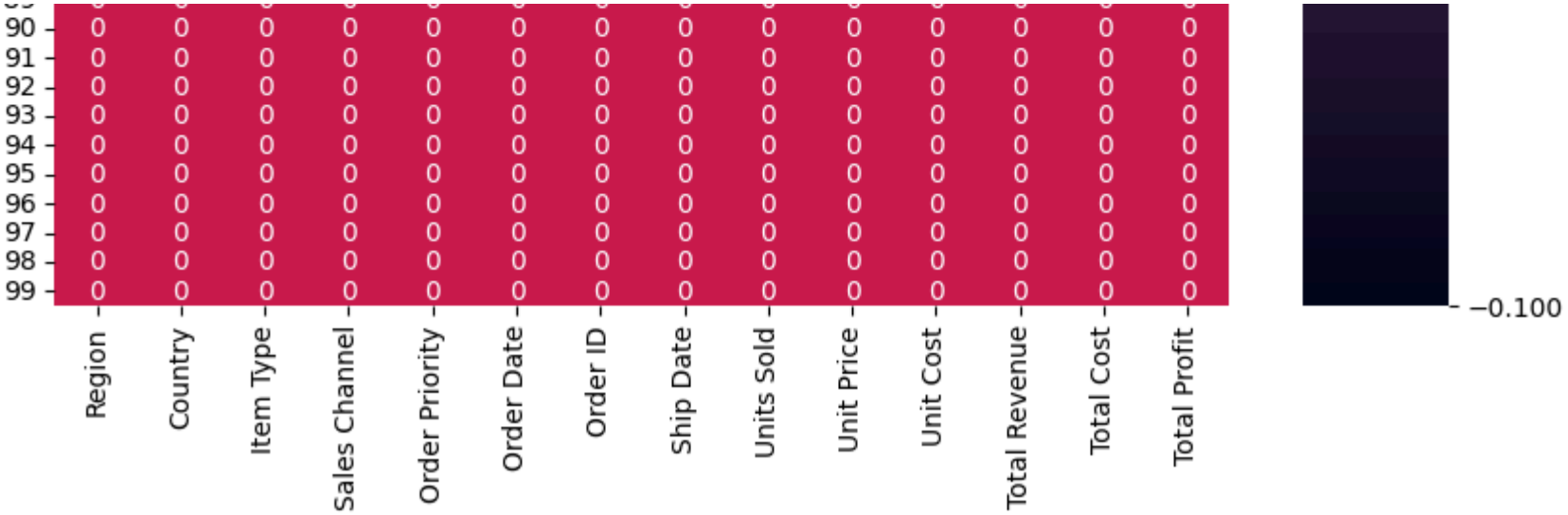
100 rows × 14 columns

In [ ]:
```
plt.figure(figsize=(10,20))
sns.heatmap(data.isnull(),annot= True)    # NO NULL VALUES
```

Out[ ]: <AxesSubplot: >


0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0
2 0 0 0 0 0 0 0 0 0 0 0 0 0
3 0 0 0 0 0 0 0 0 0 0 0 0 0
4 0 0 0 0 0 0 0 0 0 0 0 0 0
5 0 0 0 0 0 0 0 0 0 0 0 0 0
6 0 0 0 0 0 0 0 0 0 0 0 0 0
7 0 0 0 0 0 0 0 0 0 0 0 0 0
8 0 0 0 0 0 0 0 0 0 0 0 0 0
9 0 0 0 0 0 0 0 0 0 0 0 0 0
10 0 0 0 0 0 0 0 0 0 0 0 0 0
11 0 0 0 0 0 0 0 0 0 0 0 0 0
12 0 0 0 0 0 0 0 0 0 0 0 0 0
13 0 0 0 0 0 0 0 0 0 0 0 0 0
14 0 0 0 0 0 0 0 0 0 0 0 0 0
15 0 0 0 0 0 0 0 0 0 0 0 0 0
16 0 0 0 0 0 0 0 0 0 0 0 0 0
17 0 0 0 0 0 0 0 0 0 0 0 0 0
18 0 0 0 0 0 0 0 0 0 0 0 0 0
19 0 0 0 0 0 0 0 0 0 0 0 0 0
20 0 0 0 0 0 0 0 0 0 0 0 0 0
21 0 0 0 0 0 0 0 0 0 0 0 0 0
22 0 0 0 0 0 0 0 0 0 0 0 0 0
23 0 0 0 0 0 0 0 0 0 0 0 0 0
24 0 0 0 0 0 0 0 0 0 0 0 0 0
25 0 0 0 0 0 0 0 0 0 0 0 0 0
26 0 0 0 0 0 0 0 0 0 0 0 0 0
27 0 0 0 0 0 0 0 0 0 0 0 0 0
28 0 0 0 0 0 0 0 0 0 0 0 0 0
29 0 0 0 0 0 0 0 0 0 0 0 0 0
30 0 0 0 0 0 0 0 0 0 0 0 0 0
31 0 0 0 0 0 0 0 0 0 0 0 0 0
32 0 0 0 0 0 0 0 0 0 0 0 0 0
33 0 0 0 0 0 0 0 0 0 0 0 0 0
34 0 0 0 0 0 0 0 0 0 0 0 0 0
35 0 0 0 0 0 0 0 0 0 0 0 0 0
36 0 0 0 0 0 0 0 0 0 0 0 0 0
37 0 0 0 0 0 0 0 0 0 0 0 0 0
38 0 0 0 0 0 0 0 0 0 0 0 0 0
39 0 0 0 0 0 0 0 0 0 0 0 0 0
40 0 0 0 0 0 0 0 0 0 0 0 0 0
41 0 0 0 0 0 0 0 0 0 0 0 0 0
42 0 0 0 0 0 0 0 0 0 0 0 0 0
43 0 0 0 0 0 0 0 0 0 0 0 0 0
0.100
0.075
0.050
0.025


0.000
−0.025
−0.050
−0.075



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

Out[ ]:

	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
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	2010-05-28	669165933	2010-06-27	9925	255.28	159.42	2533654.00	925235.620303	951410.50
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	248406.36
2	Europe	Russia	Office Supplies	Offline	L	2014-05-02	341417157	2014-05-08	1779	651.21	524.96	1158502.59	933903.84	224598.75

Data Analysis:

Queries:

Which regions have the highest total sales revenue?

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

Which country has the highest total profit?

How does the sales channel affect the order priority distribution?

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

Which item types have the highest and lowest total sales?

How does the order priority vary across different regions?

What is the correlation between unit price and total profit?

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

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

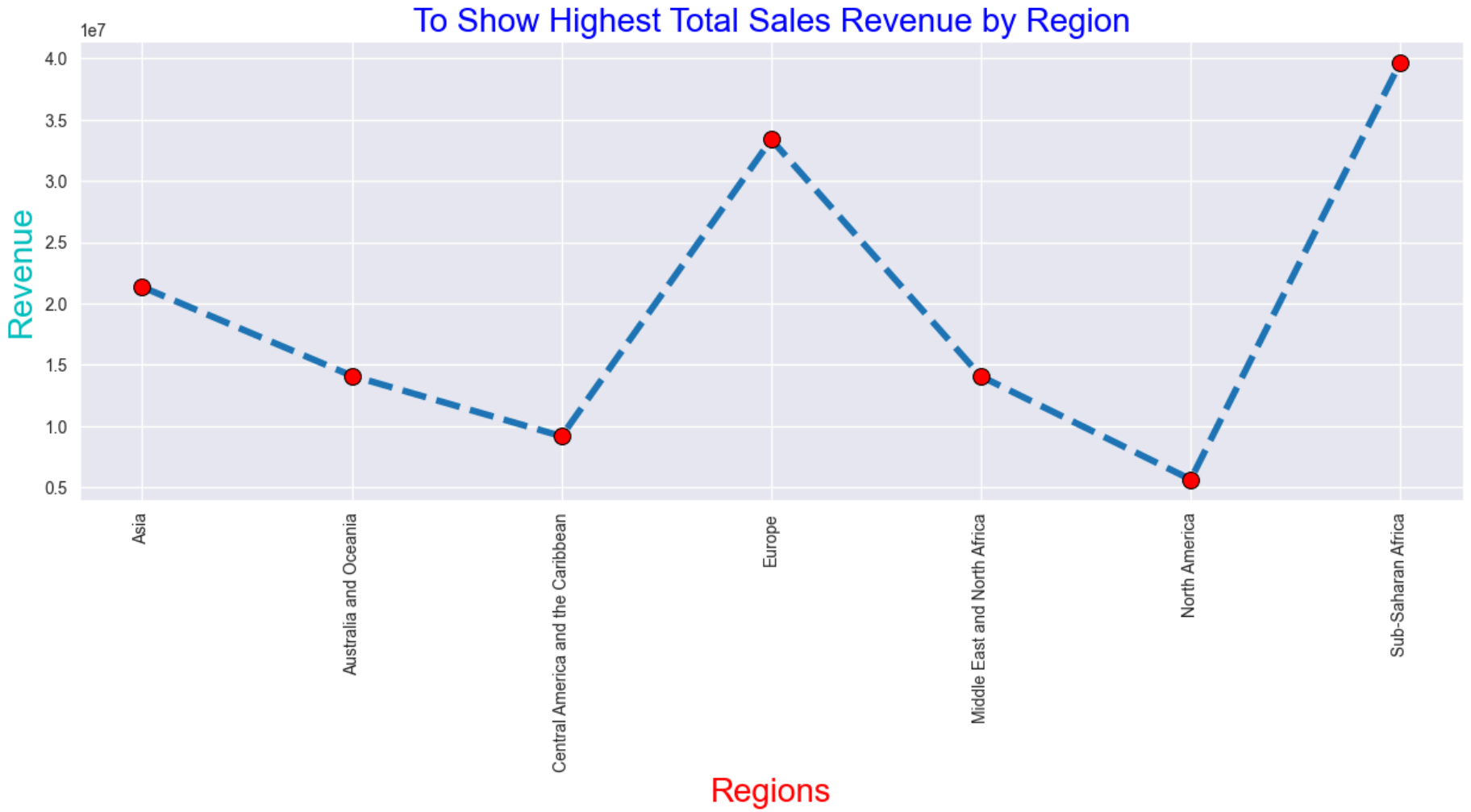
1- Which regions have the highest total sales revenue?

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

Out[ ]:
```
'Sub-Saharan Africa'
```

In [ ]:
```
group_data= data.groupby(data['Region'])['Total Revenue'].sum()
sns.set_style('darkgrid')
plt.figure(figsize=(15,5))
sns.lineplot(data= group_data, linestyle= '--' ,linewidth= 4 , marker= 'o', markersize= 10,
             markerfacecolor='red', markeredgecolor='black')

plt.xticks(rotation= 90)
plt.title('To Show Highest Total Sales Revenue by Region', fontsize= 20, color= 'Blue')
plt.xlabel('Regions', fontsize= 20, color= 'red')
plt.ylabel('Revenue', fontsize= 20, color= 'c')
plt.show()
```

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



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

In [ ]:
```
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[ ]:

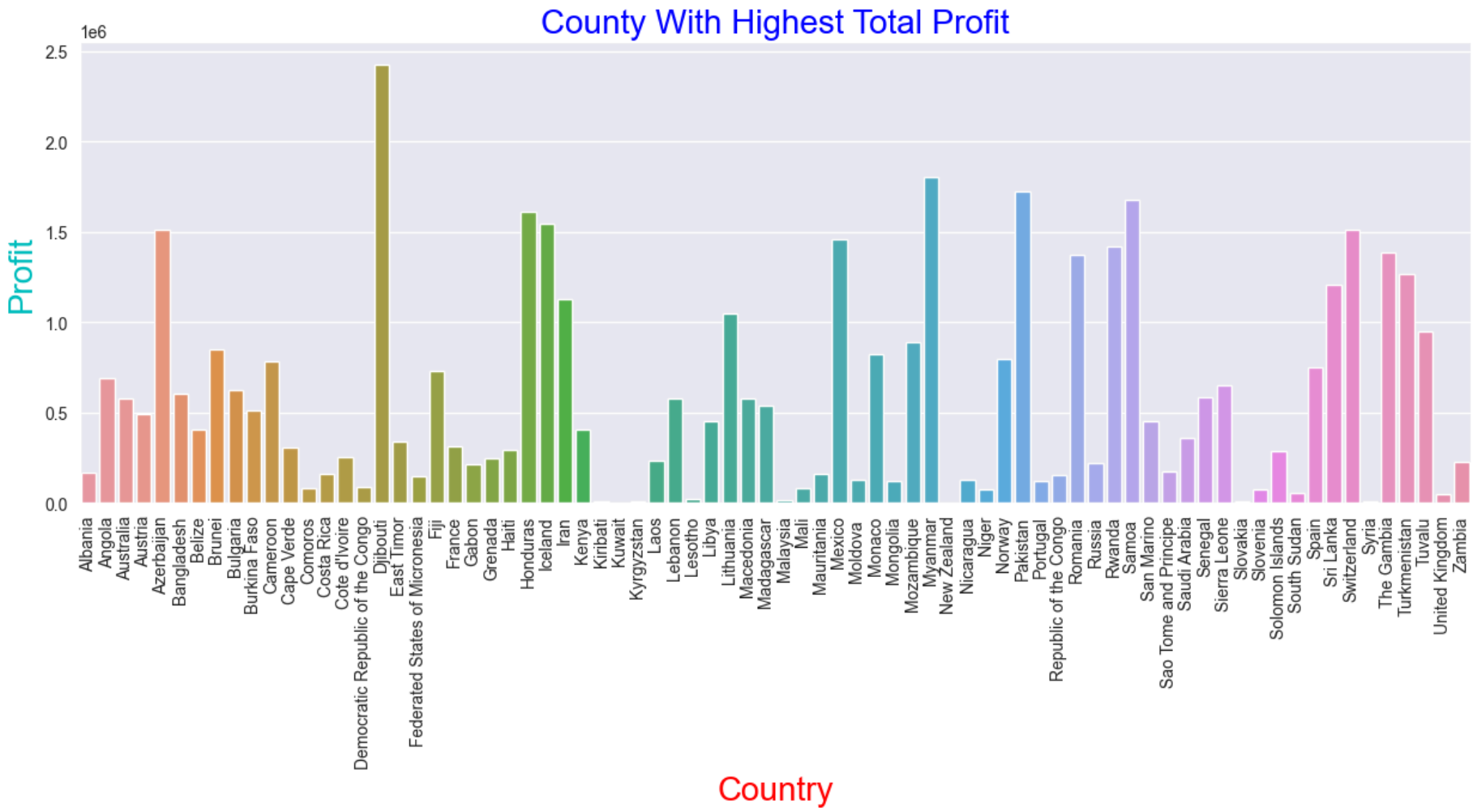
Item Type	Average Unit Price	Average Unit Cost
Baby Food	255.28	159.42
Beverages	47.45	31.79
Cereal	205.70	117.11
Clothes	109.28	35.84
Cosmetics	437.20	263.33
Fruits	9.33	6.92
Household	668.27	502.54
Meat	421.89	364.69
Office Supplies	651.21	524.96
Personal Care	81.73	56.67
Snacks	152.58	97.44
Vegetables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

3- Which country has the highest total profit?

In [ ]:
```
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 [ ]:
```
group_data= data.groupby(data['Country']) ['Total Profit'].sum()
sns.set_style('darkgrid')
plt.figure(figsize=(15,5))
sns.barplot(x= group_data.index, y= group_data )

plt.xticks(rotation= 90)
plt.title('County With Highest Total Profit', fontsize= 20, color= 'Blue')
plt.xlabel('Country', fontsize= 20, color= 'red')
plt.ylabel('Profit', fontsize= 20, color= 'c')
plt.show()
```



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

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

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

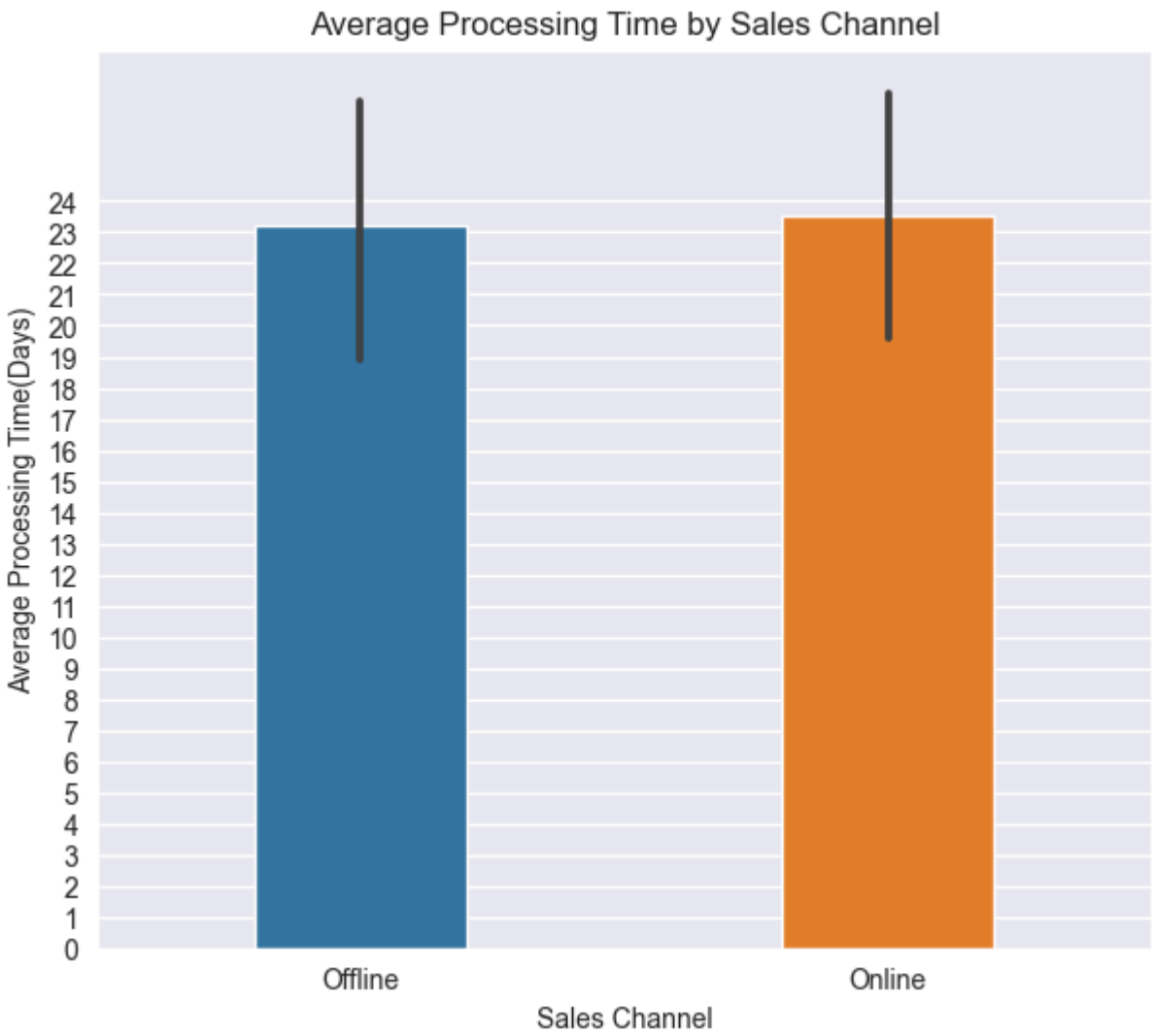
In [ ]:
```
Sales_Channel_Order_Priority_Distribution = data.groupby(['Sales Channel', 'Order Priority'])['Order Priority'].count()

# Reset the index to convert the grouped data into a DataFrame
Sales_Channel_Order_Priority_Distribution = Sales_Channel_Order_Priority_Distribution.reset_index(name='Count')

# Set the style
sns.set_style('darkgrid')

# Create the bar plot
plt.figure(figsize=(10, 6))
sns.barplot(x='Sales Channel', y='Count', hue='Order Priority', data=Sales_Channel_Order_Priority_Distribution)

# Add labels and title
plt.xlabel('Sales Channel')
plt.ylabel('Count')
plt.title('Sales Channel Order Priority Distribution')

# Display the plot
plt.show()
```

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

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

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

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

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

sns.barplot(data= data, x= data['Sales Channel'], y=data['Processing Time'].dt.days, width= 0.4 )

plt.title('Average Processing Time by Sales Channel')
plt.xlabel('Sales Channel')
plt.yticks(np.arange(0,25,1))
plt.ylabel('Average Processing Time(Days)')

plt.show()
```



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

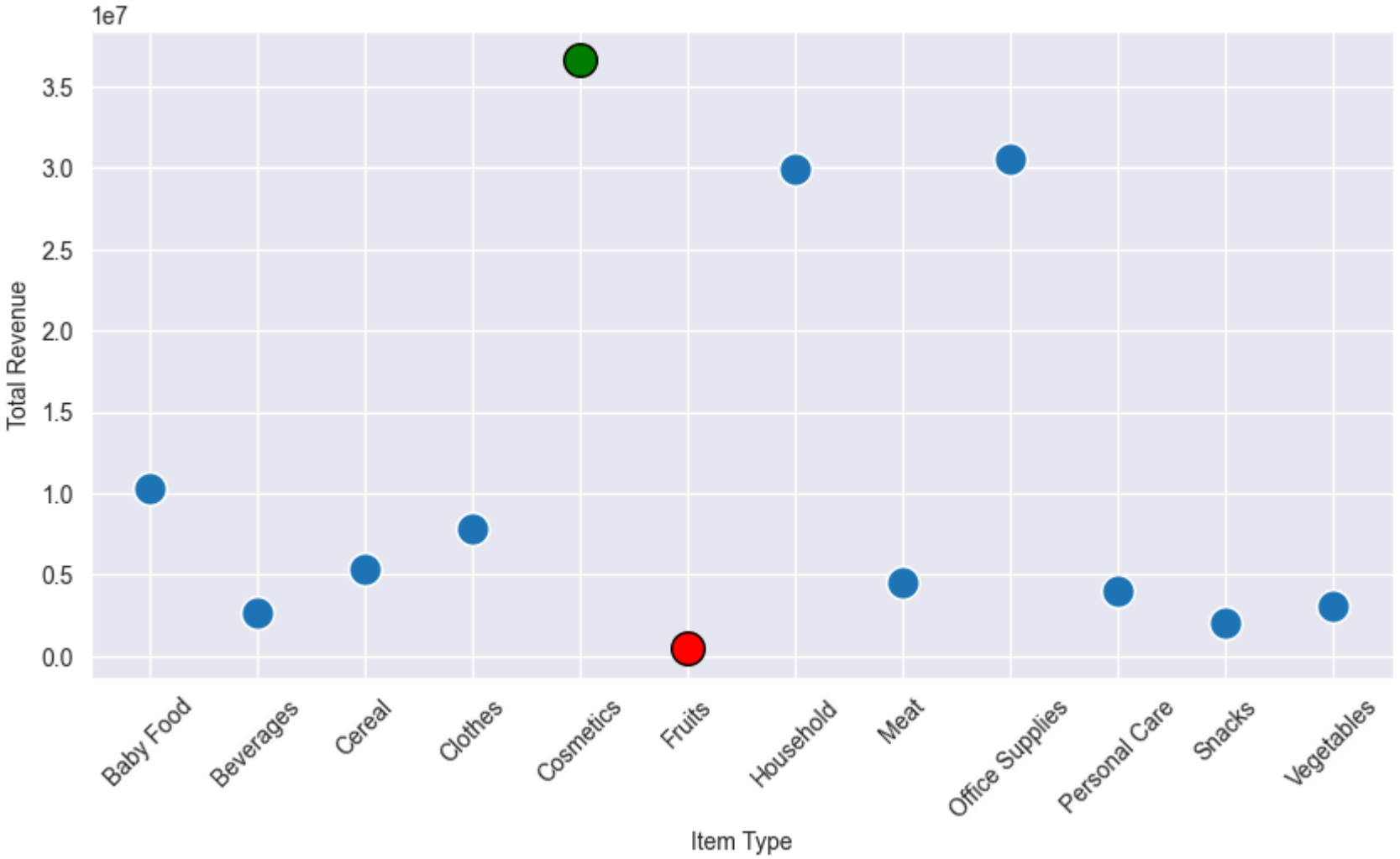
In [ ]:
```
group_item_type= data.groupby(data['Item Type'])['Total Revenue'].sum()

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':", low
```

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

In [ ]:

```
plt.figure(figsize=(10,5))

# Highlight Max Value
sns.scatterplot(x=group_item_type.index, y=group_item_type, s=200)
max_index = group_item_type.idxmax()
plt.scatter(x=max_index, y=group_item_type[max_index], s=200, color='Green', edgecolor='black')

# Highlight the minimum value
min_index = group_item_type.idxmin()
plt.scatter(x=min_index, y=group_item_type[min_index], s=200, color='RED', edgecolor='black')

plt.yticks(rotation= 0)
plt.xticks(rotation= 45)
plt.show()
```



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

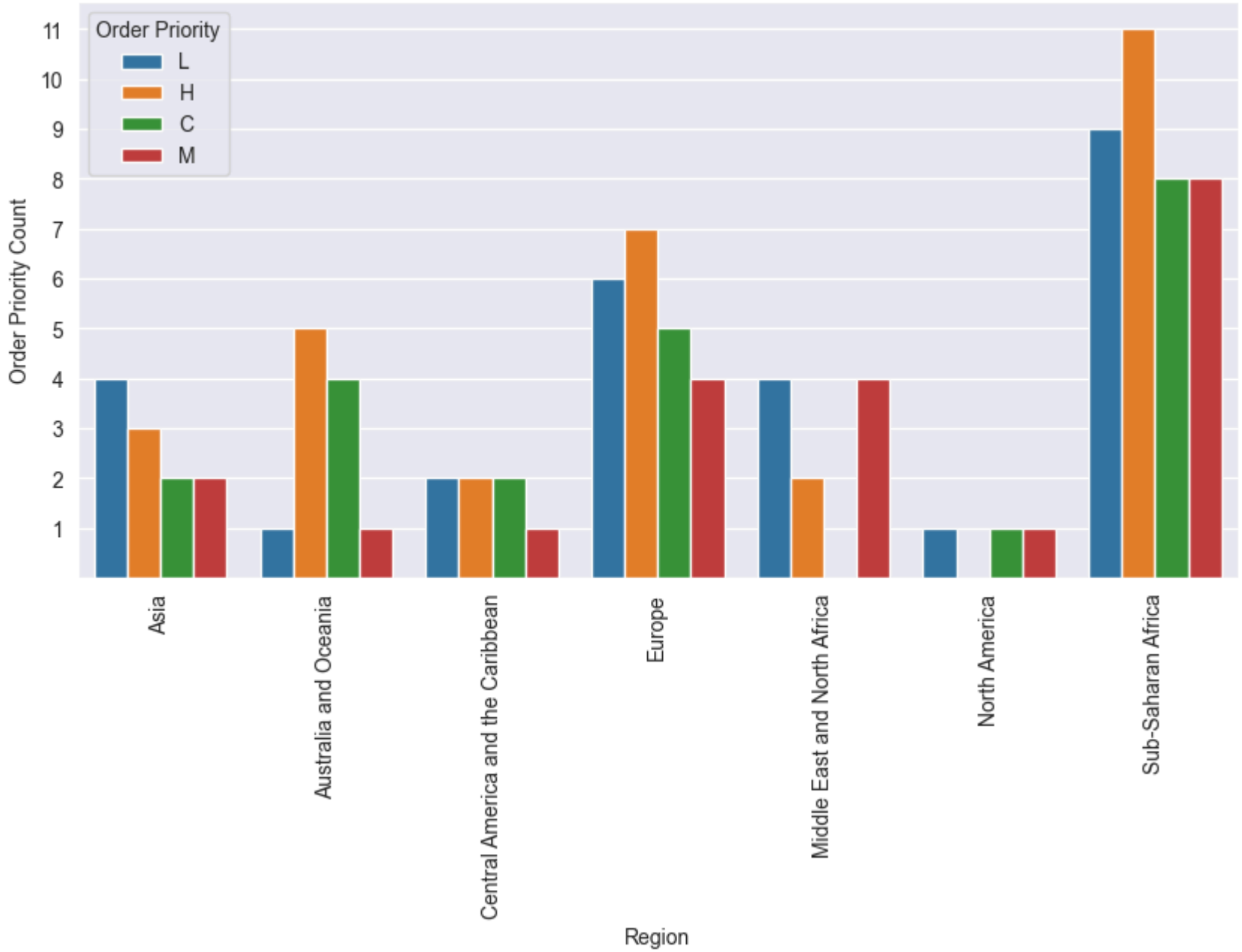
In [ ]:
```
Diff_regions_by_order_priority= data.groupby(data['Region'])['Order Priority'].value_counts()
Diff_regions_by_order_priority
```

Out[ ]:

Region	Order Priority	
Asia	L	4
	H	3
	C	2
	M	2
Australia and Oceania	H	5
	C	4
	L	1
	M	1
Central America and the Caribbean	C	2
	H	2
	L	2
	M	1
Europe	H	7
	L	6
	C	5
	M	4
Middle East and North Africa	L	4
	M	4
	H	2
North America	C	1
	L	1
	M	1
Sub-Saharan Africa	H	11
	L	9
	C	8
	M	8

Name: Order Priority, dtype: int64

In [ ]:
```
Diff_regions_by_order_priority= data.groupby(data['Region'])['Order Priority'].value_counts().reset_index(name='Order Priority
plt.figure(figsize= (10,5))
sns.barplot(data= Diff_regions_by_order_priority, x= 'Region', y= 'Order Priority Count', hue= 'Order Priority')
plt.xticks(rotation= 90)
plt.yticks(np.arange(1,12,1))

plt.show()
```


Order Priority
L
H
C
M
Order Priority Count
11
10
9
8
7
6
5
4
3
2
1
Asia
Australia and Oceania
Central America and the Caribbean
Europe
Middle East and North Africa
North America
Sub-Saharan Africa
Region

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

In [ ]:
```
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: 0.5573652488121267
```

In [ ]:
```
plt.figure(figsize=(4,2))
plt.scatter(x= Correlation_Unit_Price_Total_Profit, y= Correlation_Unit_Price_Total_Profit, s= 200, color= 'RED' )
plt.xticks(np.arange(-1,2,0.5))
plt.yticks(np.arange(-1,2,0.5))
plt.title('Correlation_Unit_Price_Total_Profit')

plt.show
```

Out[ ]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```

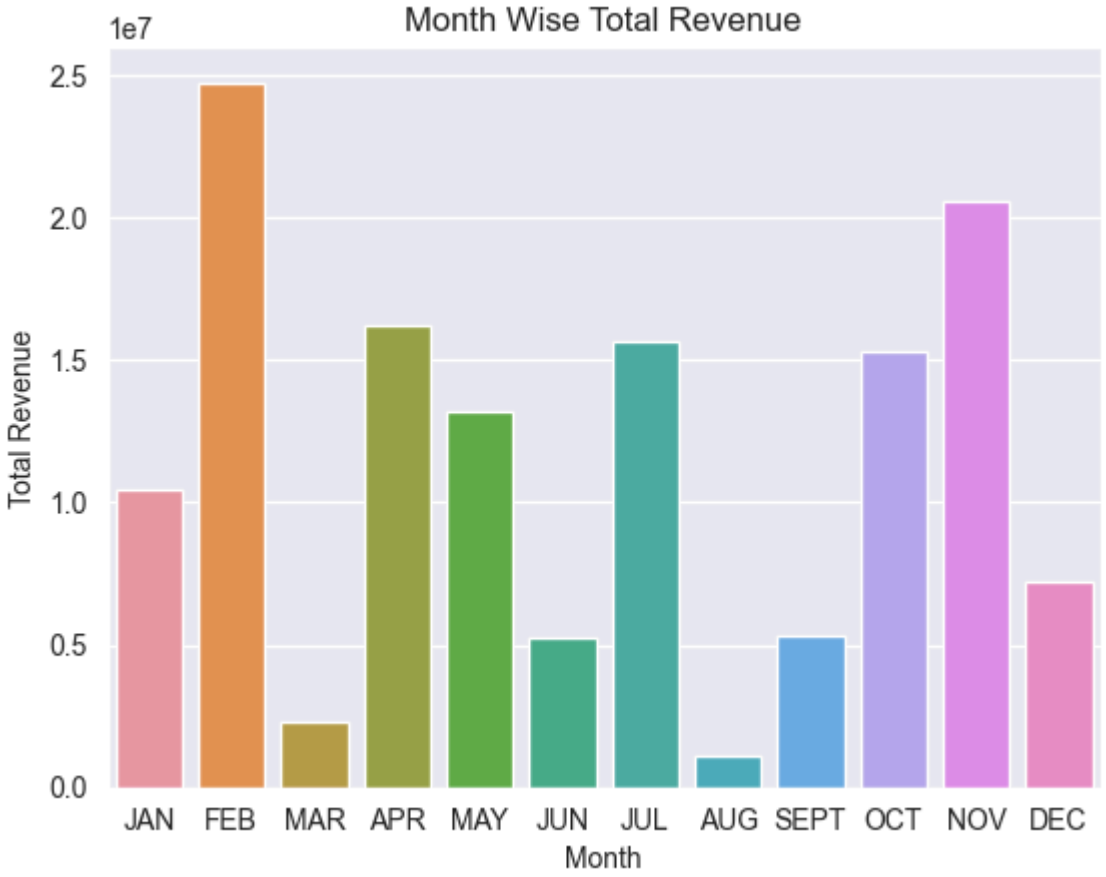
9- Are there any seasonal trends or patterns in the sales data?

In [ ]:
```
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[ ]:
```
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 [ ]:
```
sns.barplot(x= monthly_sales.index, y= monthly_sales)
plt.title('Month Wise Total Revenue')
plt.xlabel('Month')
plt.ylabel('Total Revenue')
plt.show()
```



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

In [ ]:
```
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[ ]:

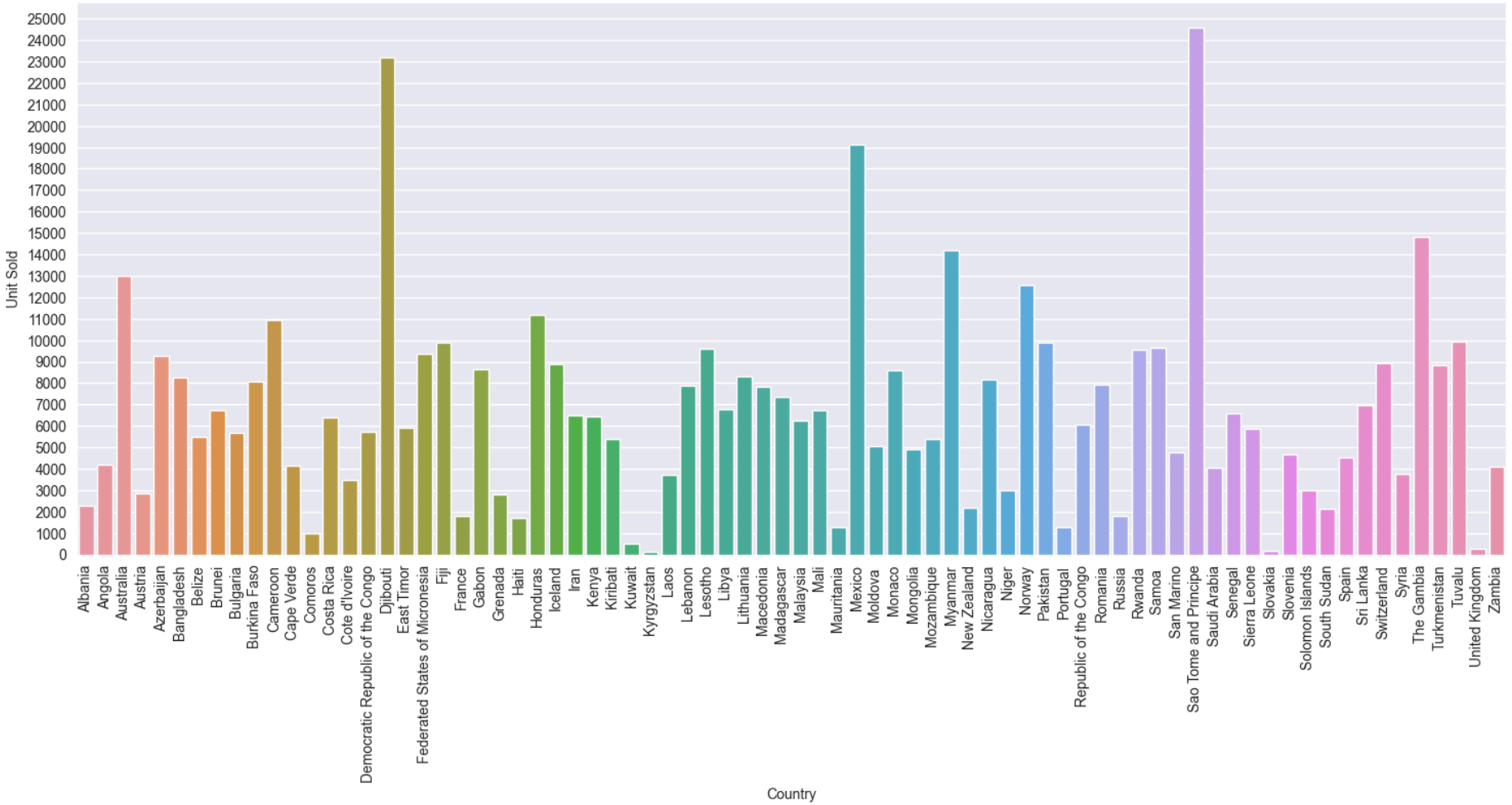
	Country	Unit Sold
0	Albania	2269
1	Angola	4187
2	Australia	12995
3	Austria	2847
4	Azerbaijan	9255
5	Bangladesh	8263
6	Belize	5498
7	Brunei	6708
8	Bulgaria	5660
9	Burkina Faso	8082
10	Cameroon	10948
11	Cape Verde	4168
12	Comoros	962
13	Costa Rica	6409
14	Cote d'Ivoire	3482
15	Democratic Republic of the 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 [ ]:
```
plt.figure(figsize= (18,7))
sns.barplot( data= Diff_countries_by_unit_sold, x= 'Country', y= 'Unit Sold')
plt.xticks(rotation= 90)
plt.yticks(np.arange(0,26000,1000))
plt.show()
```



Other Queries:

How does the total sales revenue vary across different countries?

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

Which sales channel has the highest average unit price?

Are there any outliers in the total cost distribution?

How does the total profit vary across different item types?

What is the average order processing time for each country?

Which region has the highest average total revenue per order?

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

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

Are there any trends or patterns in the order dates?

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

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

Out[ ]:

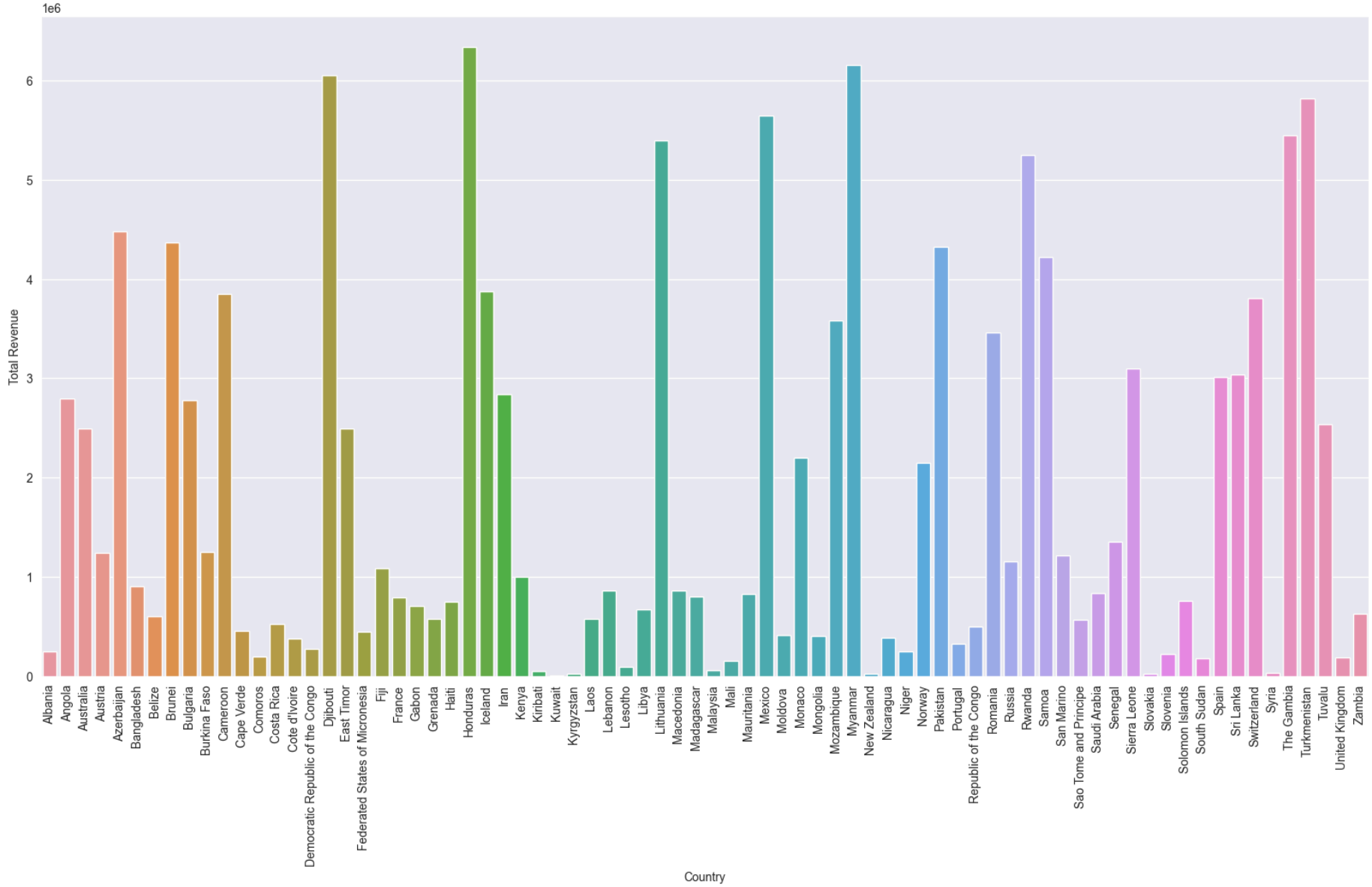
	Country	Total Revenue
0	Albania	247956.32
1	Angola	2798046.49
2	Australia	2489933.49
3	Austria	1244708.40
4	Azerbaijan	4478800.21
5	Bangladesh	902980.64
6	Belize	600821.44
7	Brunei	4368316.68
8	Bulgaria	2779199.71
9	Burkina Faso	1245112.92
10	Cameroon	3851030.28
11	Cape Verde	455479.04
12	Comoros	197883.40
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 [ ]:
```
plt.figure(figsize=(20,10))
sns.barplot(x= sales_revenue_by_countries['Country'], y= sales_revenue_by_countries['Total Revenue'])
plt.xticks(rotation= 90)
plt.show()
```



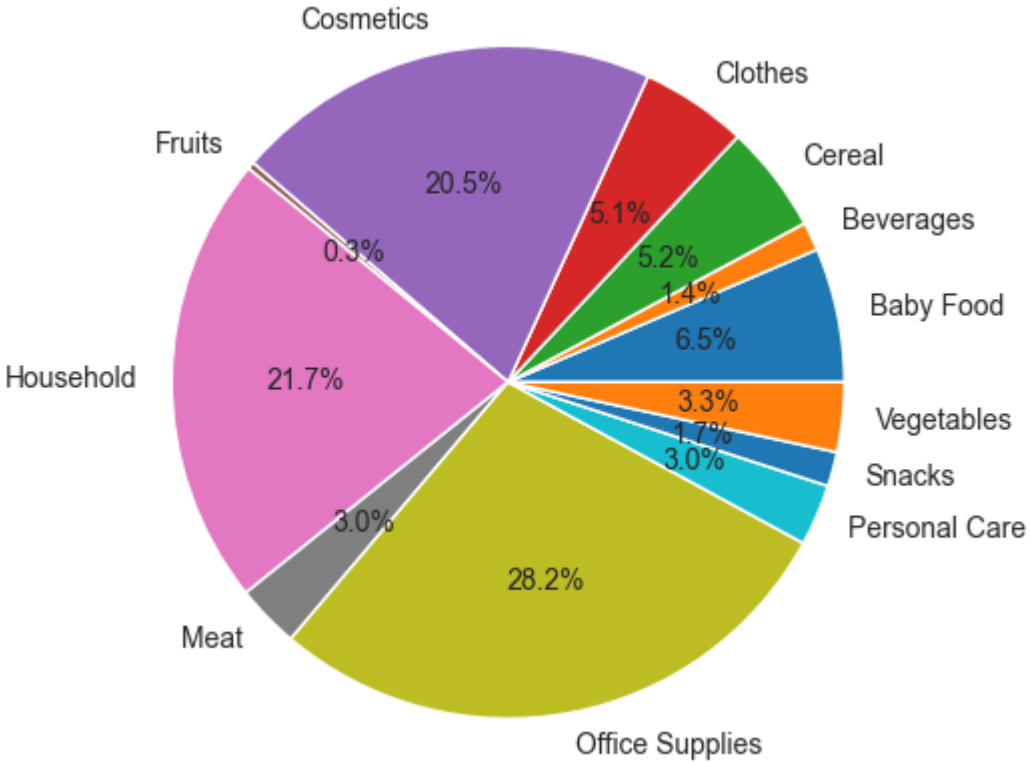
12- What is the distribution of unit prices for each item type?

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

Out[ ]:

	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 [ ]:
```
plt.pie(x= unit_price_and_item_type_distribution['Unit Price'], labels= unit_price_and_item_type_distribution['Item Type'],aut
plt.axis('equal')
plt.show()
```



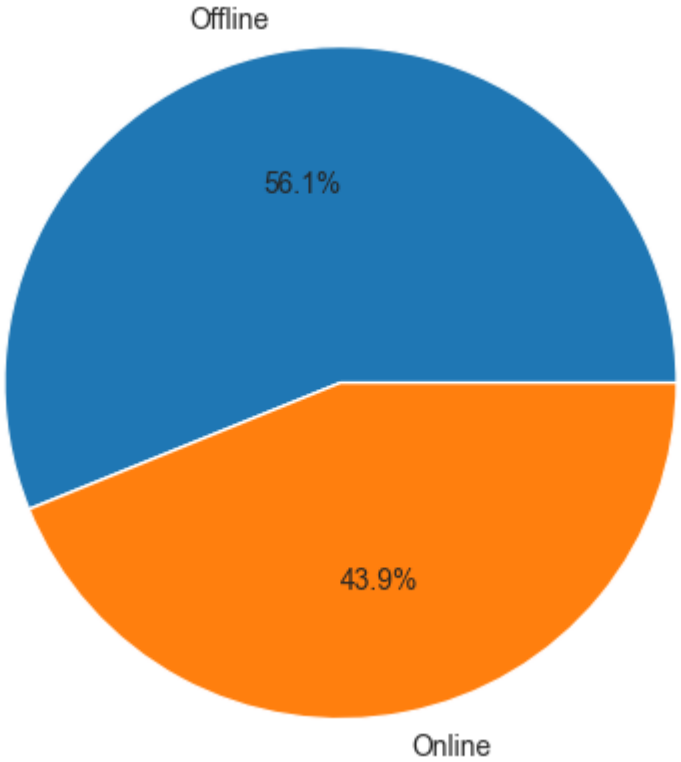
13- Which sales channel has the highest average unit price?

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

Out[ ]:

	Sales Channel	new
0	Offline	310.7206
1	Online	242.8020

In [ ]:
```
plt.pie(x= Highest_avg_unit_price_for_sales_channel['new'],labels=Highest_avg_unit_price_for_sales_channel['Sales Channel'], a
plt.axis('equal')
plt.show()
```



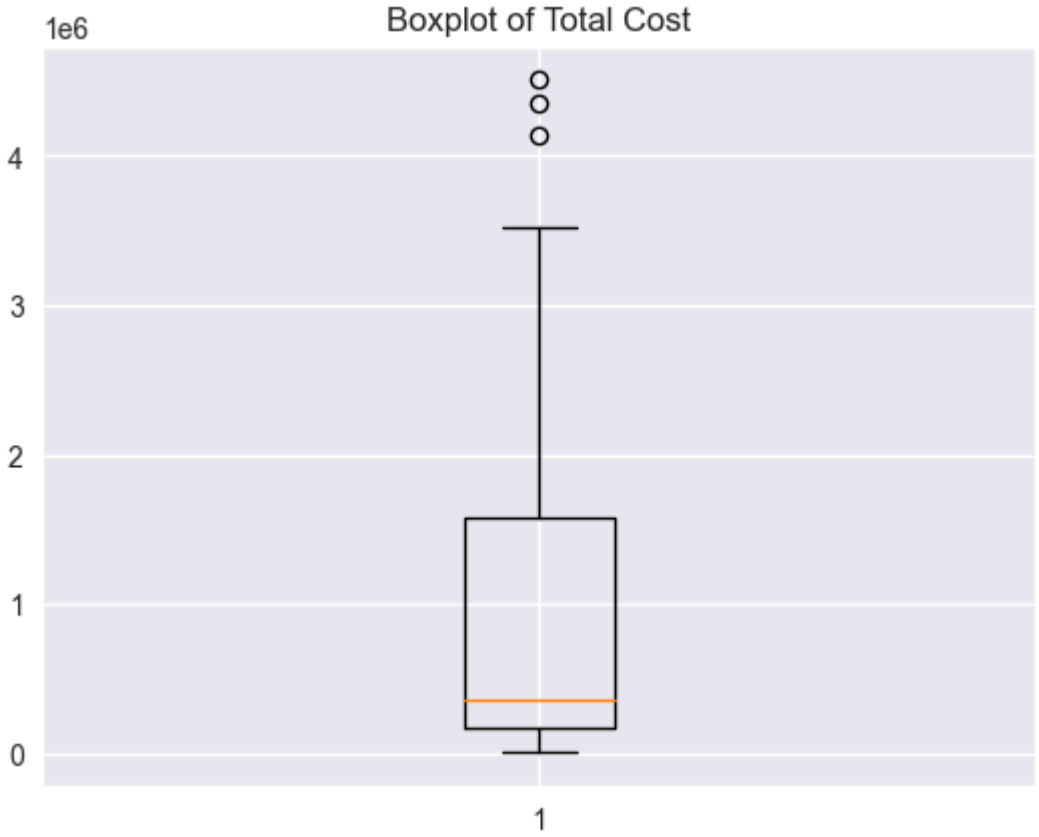
14- Are there any outliers in the total cost distribution?

In [ ]:
```
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[ ]:

	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
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	1487261.02
1	Asia	Myanmar	Household	Offline	H	2015-01-16	177713572	2015-03-01	8250	668.27	502.54	5513227.50	4145955.0	1367272.50
2	Europe	Lithuania	Office Supplies	Offline	H	2010-10-24	166460740	2010-11-17	8287	651.21	524.96	5396577.27	4350343.52	1046233.75

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

Out[ ]:

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

Out[ ]:

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

19- Is there a relationship between the number of units sold and the total profit?

In [ ]:
```
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: 0.5645504620845976
```

20- How does the order priority vary based on the item type?

In [ ]:
```
Order_priority_vary_on_item_type= data.groupby(data['Order Priority'])['Item Type'].value_counts().reset_index(name= 'No. Of I
Order_priority_vary_on_item_type
```

Out[ ]:

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

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

In [ ]: