In [1]:
```
import pandas as pd
```

In [3]:
```
import numpy as np
```

In [4]:
```
import seaborn as sns
```

In [5]:
```
import matplotlib.pyplot as plt
```

In [8]:
```
df=pd.read_csv("C:\\Users\\wwwsa\\Downloads\\sel. Projects\\Amazon Sales data
df
```

Out[8]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Uni So
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	5/28/2010	669165933	6/27/2010	99
1	Central America and the Caribbean	Grenada	Cereal	Online	C	8/22/2012	963881480	9/15/2012	28
2	Europe	Russia	Office Supplies	Offline	L	5/2/2014	341417157	5/8/2014	17
3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	6/20/2014	514321792	7/5/2014	81
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2/1/2013	115456712	2/6/2013	50
...	...	...	...	...	...	...	...	...	
95	Sub-Saharan Africa	Mali	Clothes	Online	M	7/26/2011	512878119	9/3/2011	8
96	Asia	Malaysia	Fruits	Offline	L	11/11/2011	810711038	12/28/2011	62
97	Sub-Saharan Africa	Sierra Leone	Vegetables	Offline	C	6/1/2016	728815257	6/29/2016	14
98	North America	Mexico	Personal Care	Offline	M	7/30/2015	559427106	8/8/2015	57
99	Sub-Saharan Africa	Mozambique	Household	Offline	L	2/10/2012	665095412	2/15/2012	53

100 rows × 14 columns

In [10]:
```
df.shape
```

Out[10]:
```
(100, 14)
```

In [11]:
```
df.describe()
```

Out[11]:

	Order ID	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Total
count	1.000000e+02	100.000000	100.000000	100.000000	1.000000e+02	1.000000e+02	1.00000
mean	5.550204e+08	5128.710000	276.761300	191.048000	1.373488e+06	9.318057e+05	4.41682
std	2.606153e+08	2794.484562	235.592241	188.208181	1.460029e+06	1.083938e+06	4.38537
min	1.146066e+08	124.000000	9.330000	6.920000	4.870260e+03	3.612240e+03	1.25802
25%	3.389225e+08	2836.250000	81.730000	35.840000	2.687212e+05	1.688680e+05	1.21443
50%	5.577086e+08	5382.500000	179.880000	107.275000	7.523144e+05	3.635664e+05	2.90768
75%	7.907551e+08	7369.000000	437.200000	263.330000	2.212045e+06	1.613870e+06	6.35828
max	9.940222e+08	9925.000000	668.270000	524.960000	5.997055e+06	4.509794e+06	1.71992

In [13]:
```
df.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 [14]:
```
df.isnull()
```

Out[14]:

	Region	Country	Item	Sales	Order	Order	Order	Ship	Units	Unit	Unit	Tot
			Type	Channel	Priority	Date	ID	Date	Sold	Price	Cost	Revenu
0	False	False	False	False	False	False	False	False	False	False	False	Fals
1	False	False	False	False	False	False	False	False	False	False	False	Fals
2	False	False	False	False	False	False	False	False	False	False	False	Fals
3	False	False	False	False	False	False	False	False	False	False	False	Fals
4	False	False	False	False	False	False	False	False	False	False	False	Fals
...	...	...	...	...	...	...	...	...	...	...	...	.
95	False	False	False	False	False	False	False	False	False	False	False	Fals
96	False	False	False	False	False	False	False	False	False	False	False	Fals
97	False	False	False	False	False	False	False	False	False	False	False	Fals
98	False	False	False	False	False	False	False	False	False	False	False	Fals
99	False	False	False	False	False	False	False	False	False	False	False	Fals

100 rows × 14 columns

In [15]:
```
df.isnull().sum()
```

Out[15]:
```
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
```

sales trend over time

```
In [18]: df['Order Date'] = pd.to_datetime(df['Order Date'])
         df['Order Date']
```

```
Out[18]: 0     2010-05-28
         1     2012-08-22
         2     2014-05-02
         3     2014-06-20
         4     2013-02-01
                  ...
         95    2011-07-26
         96    2011-11-11
         97    2016-06-01
         98    2015-07-30
         99    2012-02-10
         Name: Order Date, Length: 100, dtype: datetime64[ns]
```

calculate total sales

```
In [19]: total_sales = df['Total Revenue'].sum()
```

```
In [20]: total_sales
```

```
Out[20]: 137348768.31
```

```
In [21]: total_units = df['Units Sold'].sum()
```

```
In [22]: total_units
```

```
Out[22]: 512871
```

```
In [23]: total_cost = df['Total Cost'].sum()
```

```
In [24]: total_cost
```

```
Out[24]: 93180569.91000001
```

```
In [25]: total_profit = df['Total Profit'].sum()
```

```
In [26]: total_profit
```

```
Out[26]: 44168198.39999999
```

```
In [27]: total_Profit1 = -total_cost
In [28]: total_Profit1

Out[28]: 44168198.39999999

In [31]: total_profit_percentage = total_profit/total_cost*100

In [32]: total_profit_percentage

Out[32]: 47.400652778428565
```

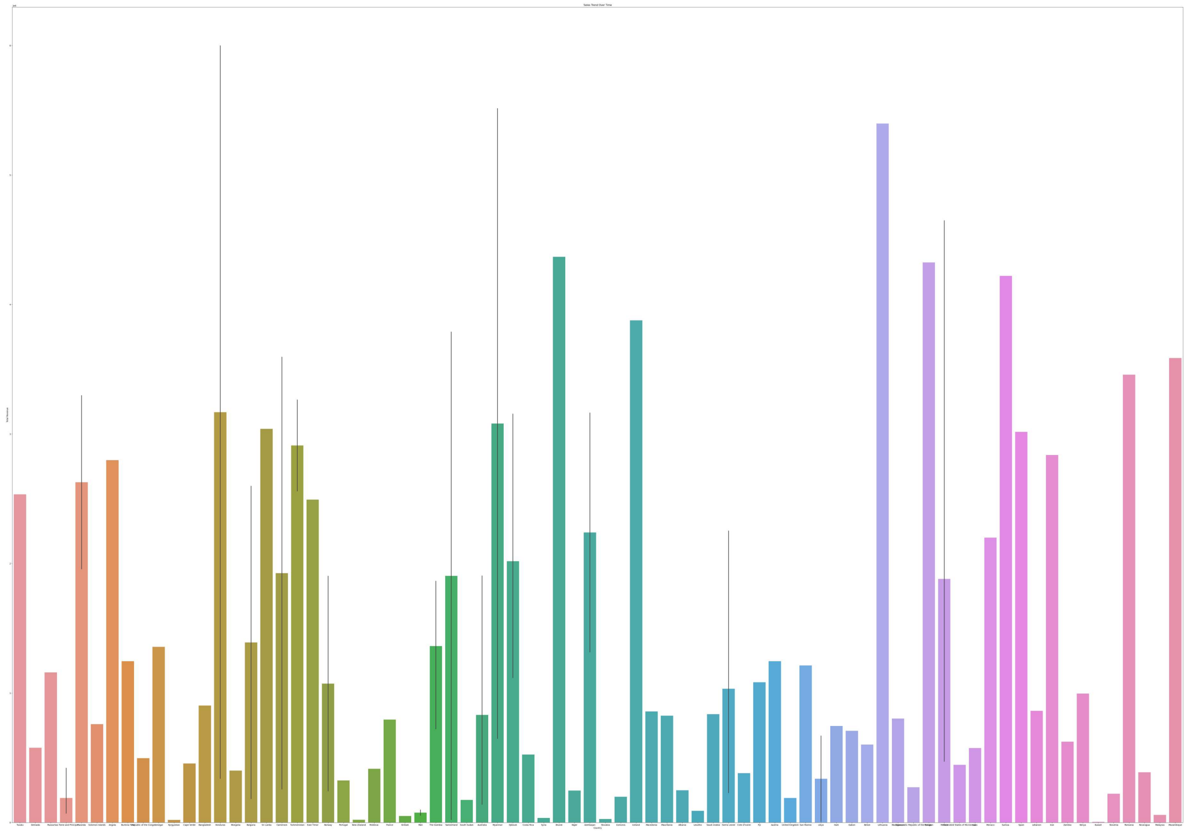
identifying best selling products

```
In [33]: bsp = df.groupby('Item Type')['Units Sold'].sum().sort_values(ascending=False

In [34]: bsp

Out[34]: Item Type
         Cosmetics          83718
         Clothes            71260
         Beverages          56708
         Fruits             49998
         Personal Care      48708
         Office Supplies    46967
         Household          44727
         Baby Food          40545
         Cereal             25877
         Vegetables         20051
         Snacks             13637
         Meat               10675
         Name: Units Sold, dtype: int64
```

visualization of sales trend

```
In [ ]:
```

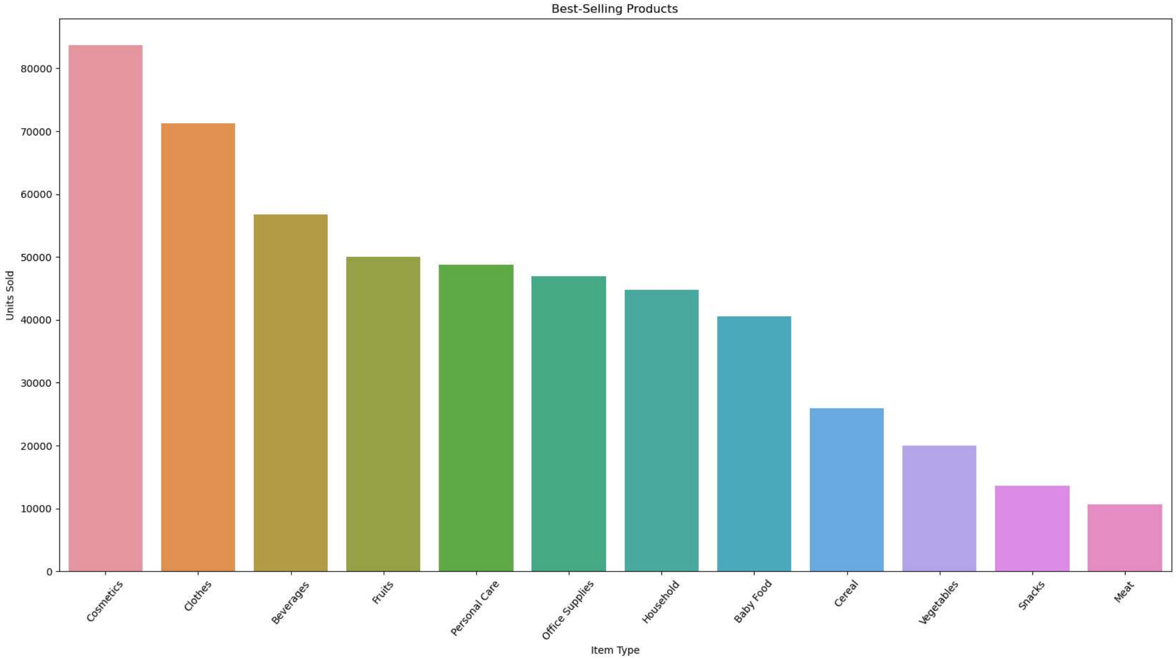
In [54]:
```
plt.figure(figsize=(100, 70))
sns.barplot(x='Country', y='Total Revenue', data = df)
plt.title('Sales Trend Over Time')
plt.show()
```



BSP Visualization

In [ ]:

In [56]:
```
plt.figure(figsize=(20, 10))
sns.barplot(x=bsp.index, y=bsp.values)
plt.title('Best-Selling Products')
plt.xlabel('Item Type')
plt.ylabel('Units Sold')
plt.xticks(rotation=50)
plt.show()
```



In [ ]:

Maximum Revenue countrywise

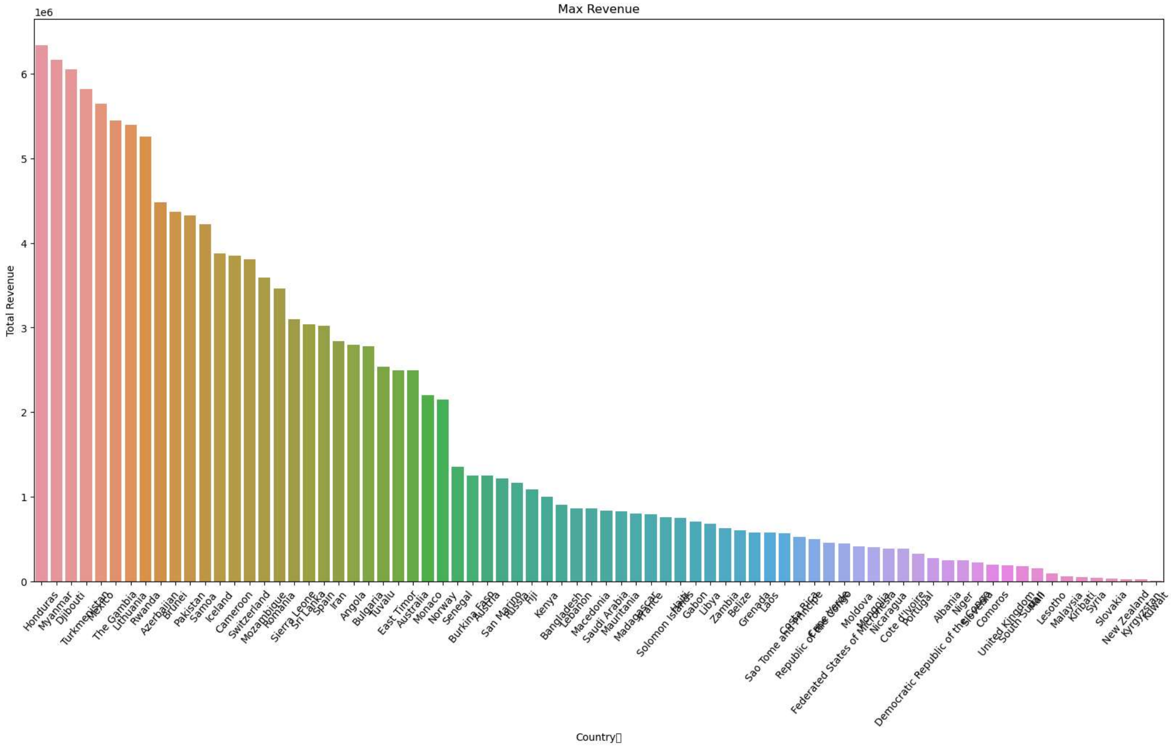
In [58]:
```
Max_revenue = df.groupby('Country')['Total Revenue'].sum().sort_values(ascend
```

In [59]:
```
Max_revenue
```

Out[59]:
```
Country
Honduras        6336545.48
Myanmar         6161257.90
Djibouti        6052890.86
Turkmenistan    5822036.20
Mexico          5643356.55
                   ...
Syria             35304.72
Slovakia          26344.26
New Zealand       20404.71
Kyrgyzstan        19103.44
Kuwait             4870.26
Name: Total Revenue, Length: 76, dtype: float64
```

In [60]:
```
plt.figure(figsize=(20, 10))
sns.barplot(x=Max_revenue.index, y=Max_revenue.values)
plt.title('Max Revenue')
plt.xlabel('Country ')
plt.ylabel('Total Revenue')
plt.xticks(rotation=50)
plt.show()
```

```
C:\ProgramData\anaconda3\Lib\site-packages\IPython\core\pylabtools.py:152: U
serWarning: Glyph 9 (	) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
```



In [ ]: