Analyzing amazon sales data (Project-1)

In [2]:
```
import pandas as pd
```

In [3]:
```
#Loading the data set
data=pd.read_csv(r"C:\Users\cindr\Downloads\Amazon Sales data.csv")
```

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]:
```
# Converting the Order Date and Ship Date columns to datetime format
data['Order Date'] = pd.to_datetime(data['Order Date'], errors='coerce')
data['Ship Date'] = pd.to_datetime(data['Ship Date'], errors='coerce')
```

In [7]:
```
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    datetime64[ns]
 6   Order ID        100 non-null    int64
 7   Ship Date       100 non-null    datetime64[ns]
 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: datetime64[ns](2), float64(5), int64(2), object(5)
memory usage: 11.1+ KB
```

In [8]:
```
# Removing rows with missing values
data.dropna(inplace=True)
```

In [9]:
```
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    datetime64[ns]
 6   Order ID        100 non-null    int64
 7   Ship Date       100 non-null    datetime64[ns]
 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: datetime64[ns](2), float64(5), int64(2), object(5)
memory usage: 11.1+ KB
```

In [10]:
```
# Extracting year and month from the Order Date
data['Year'] = data['Order Date'].dt.year
data['Month'] = data['Order Date'].dt.month
```

In [11]:
```
#cleaned data
data.head()
```

Out[11]:

	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	Year	Month
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	2010-05-28	669165933	2010-06-27	9925	255.28	159.42	2533654.00	1582243.50	951410.50	2010	5
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	2012	8
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	2014	5
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	19525.82	2014	6
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	639077.50	2013	2

In [12]:
```
data.tail()
```

Out[12]:

	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	Year	Month
95	Sub-Saharan Africa	Mali	Clothes	Online	M	2011-07-26	512878119	2011-09-03	888	109.28	35.84	97040.64	31825.92	65214.72	2011	7
96	Asia	Malaysia	Fruits	Offline	L	2011-11-11	810711038	2011-12-28	6267	9.33	6.92	58471.11	43367.64	15103.47	2011	11
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	93748.05	2016	6
98	North America	Mexico	Personal Care	Offline	M	2015-07-30	559427106	2015-08-08	5767	81.73	56.67	471336.91	326815.89	144521.02	2015	7
99	Sub-Saharan Africa	Mozambique	Household	Offline	L	2012-02-10	665095412	2012-02-15	5367	668.27	502.54	3586605.09	2697132.18	889472.91	2012	2

In [13]:
```
#Monthly sale trend
monthly_sales = data.groupby(['Year', 'Month'])['Total Revenue'].sum().reset_index()
print("Monthly Sales Trend:\n", monthly_sales)
```

```
Monthly Sales Trend:
     Year  Month  Total Revenue
0   2010      2     3410661.12
1   2010      5     2587973.26
2   2010      6     1082418.40
3   2010     10     6064933.75
4   2010     11     3458252.00
5   2010     12     2581786.39
6   2011      1     1042225.35
7   2011      2      387002.20
8   2011      4     2798046.49
9   2011      5      272410.45
10  2011      6       19103.44
11  2011      7       97040.64
12  2011      9      574951.92
13  2011     11     5938385.58
14  2012      1     1012884.00
15  2012      2     6707849.42
16  2012      3      994765.42
17  2012      4     4556012.38
18  2012      5     3782781.82
19  2012      6     2132075.27
20  2012      7     4445093.92
21  2012      8      576782.80
22  2012      9     4648152.72
23  2012     10     3042246.77
24  2013      2     3296425.02
25  2013      3      835759.10
26  2013      4     3262562.10
27  2013      6     1352867.40
28  2013      7     8545511.20
29  2013      8       89623.98
30  2013      9       71253.21
31  2013     10     2702770.40
32  2013     12      173676.25
33  2014      2     1819660.25
34  2014      4     4510578.10
35  2014      5     3060338.59
36  2014      6       75591.66
37  2014      7      688641.85
38  2014      8      455479.04
39  2014      9       20404.71
40  2014     10     1352370.65
41  2014     11     4647149.58
42  2015      1     5513227.50
43  2015      2     2003911.12
44  2015      4     1059987.26
45  2015      7     1292409.45
46  2015      8        6279.09
47  2015     10     1904138.04
48  2015     11      648030.40
49  2016      3      197883.40
50  2016      5      414371.10
51  2016      6      568269.60
52  2016      7      600821.44
53  2016     10      221117.00
54  2016     11     5876405.20
55  2016     12     4493999.48
56  2017      1     2914130.27
57  2017      2     7115008.64
58  2017      3      246415.95
59  2017      5     3097864.77
```

In [14]:
```
# Yearly Sales Trend
yearly_sales = data.groupby('Year')['Total Revenue'].sum().reset_index()
print("Yearly Sales Trend:\n", yearly_sales)
```

```
Yearly Sales Trend:
    Year  Total Revenue
0  2010    19186024.92
1  2011    11129166.07
2  2012    31898644.52
3  2013    20330448.66
4  2014    16630214.43
5  2015    12427982.86
6  2016    12372867.22
7  2017    13373419.63
```

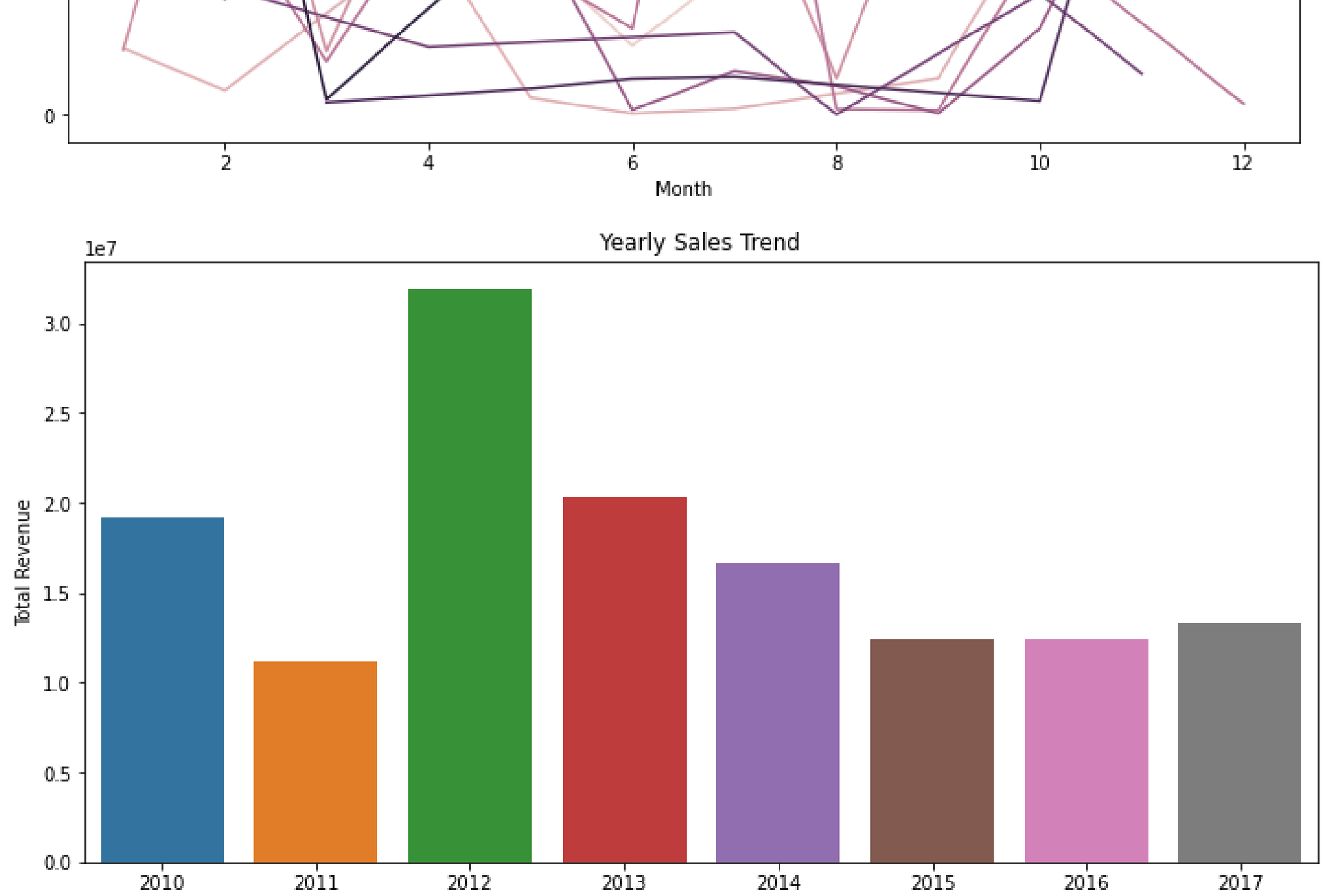
In [15]:
```
# Yearly Month-wise Sales Trend
yearly_monthly_sales = data.groupby(['Year', 'Month'])['Total Revenue'].sum().unstack().fillna(0)
print("Yearly Month-wise Sales Trend:\n", yearly_monthly_sales)
```

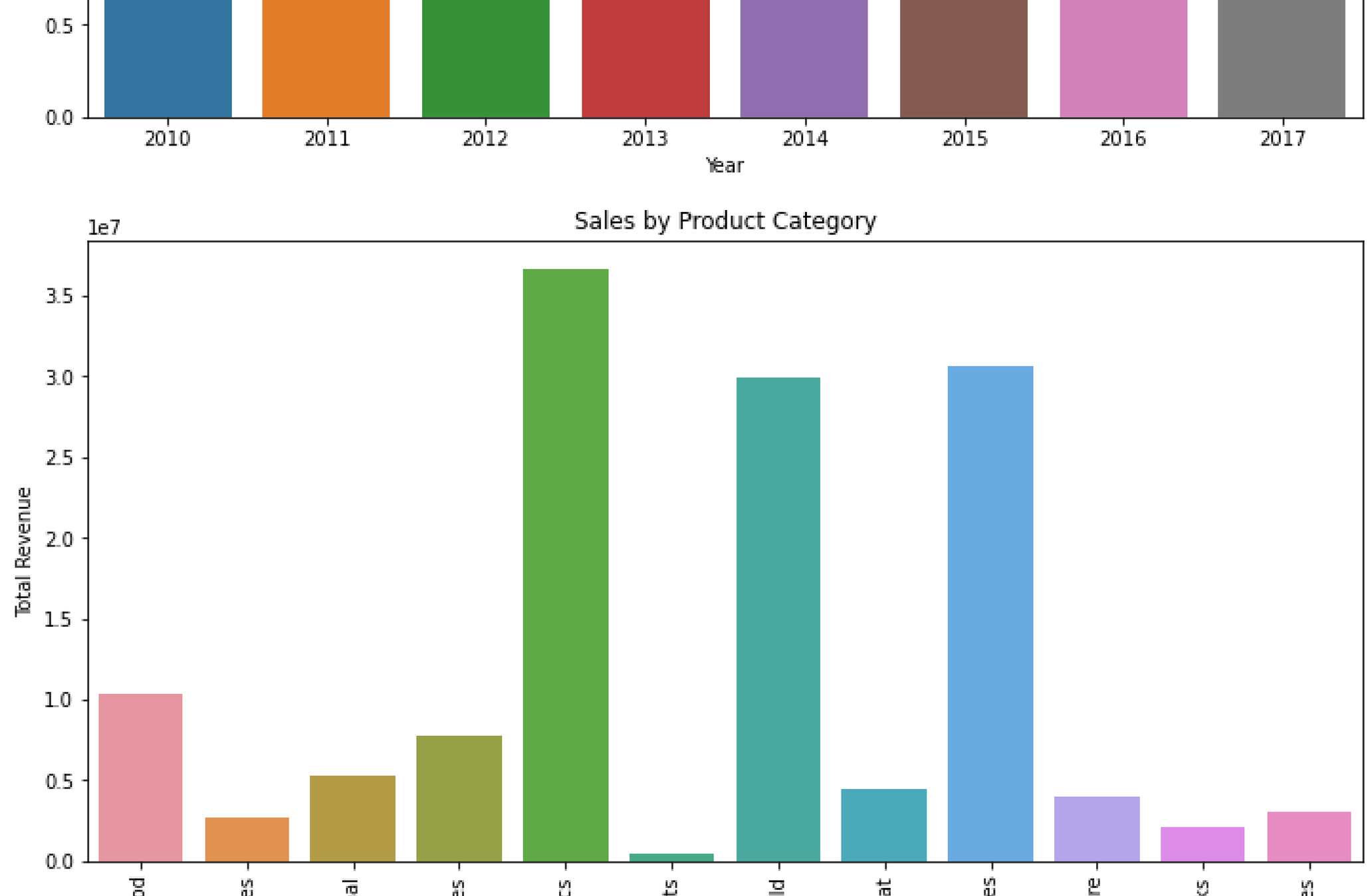
```
Yearly Month-wise Sales Trend:
 Month          1           2          3           4           5           6  \
Year
2010         0.00  3410661.12       0.00        0.00  2587973.26  1082418.40
2011   1042225.35   387002.20       0.00  2798046.49   272410.45    19103.44
2012   1012884.00  6707849.42  994765.42  4556012.38  3782781.82  2132075.27
2013         0.00  3296425.02  835759.10  3262562.10        0.00  1352867.40
2014         0.00  1819660.25       0.00  4510578.10  3060338.59    75591.66
2015   5513227.50  2003911.12       0.00  1059987.26        0.00        0.00
2016         0.00        0.00  197883.40        0.00   414371.10   568269.60
2017   2914130.27  7115008.64  246415.95        0.00  3097864.77        0.00

Month          7          8           9          10          11          12
Year
2010         0.00       0.00        0.00  6064933.75  3458252.00  2581786.39
2011     97040.64       0.00   574951.92        0.00  5938385.58        0.00
2012   4445093.92  576782.80  4648152.72  3042246.77        0.00        0.00
2013   8545511.20   89623.98    71253.21  2702770.40        0.00   173676.25
2014    688641.85  455479.04    20404.71  1352370.65  4647149.58        0.00
2015   1292409.45    6279.09        0.00  1904138.04   648030.40        0.00
2016    600821.44       0.00        0.00   221117.00  5876405.20  4493999.48
2017         0.00       0.00        0.00        0.00        0.00        0.00
```

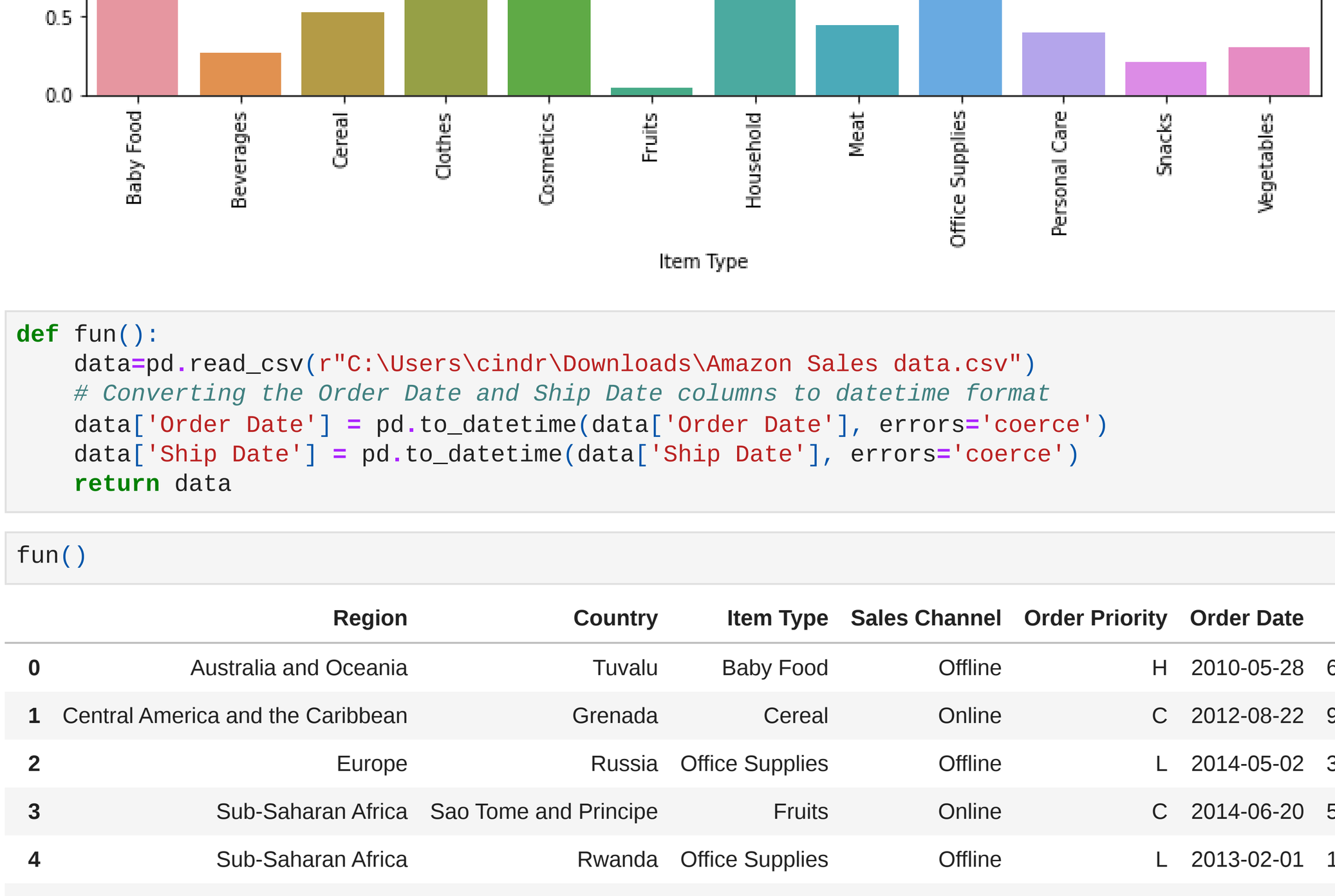
In [16]:
```
#Finding Key Metrics and Relationships
#Calculating total sales and average sales
total_sales = data['Total Revenue'].sum()
average_sales = data['Total Revenue'].mean()

# Group by product category and calculate sales
category_sales = data.groupby('Item Type')['Total Revenue'].sum().reset_index()

# Display key metrics
print(f"Total Sales: {total_sales}")
print(f"Average Sales: {average_sales}")
print("Sales by Product Category:\n", category_sales)
```

```
Total Sales: 137348768.31
Average Sales: 1373487.6830999998
Sales by Product Category:
           Item Type  Total Revenue
0         Baby Food    10350327.60
1         Beverages     2690794.60
2            Cereal     5322898.90
3           Clothes     7787292.80
4         Cosmetics    36601509.60
5            Fruits      466481.34
6         Household    29889712.29
7              Meat     4503675.75
8   Office Supplies    30585380.07
9     Personal Care     3980904.84
10           Snacks     2080733.46
11       Vegetables     3089057.06
```

In [22]:
```
#Visualizations
import matplotlib.pyplot as plt
import seaborn as sns

# Plot monthly sales trend
plt.figure(figsize=(12, 6))
sns.lineplot(data=monthly_sales, x='Month', y='Total Revenue', hue='Year')
plt.title('Monthly Sales Trend')
plt.show()

# Plot yearly sales trend
plt.figure(figsize=(12, 6))
sns.barplot(data=yearly_sales, x='Year', y='Total Revenue')
plt.title('Yearly Sales Trend')
plt.show()

# Plot sales by product category
plt.figure(figsize=(12, 6))
sns.barplot(data=category_sales, x='Item Type', y='Total Revenue')
plt.title('Sales by Product Category')
plt.xticks(rotation=90)
plt.show()
```







In [19]:
```
def fun():
    data=pd.read_csv(r"C:\Users\cindr\Downloads\Amazon Sales data.csv")
    # Converting the Order Date and Ship Date columns to datetime format
    data['Order Date'] = pd.to_datetime(data['Order Date'], errors='coerce')
    data['Ship Date'] = pd.to_datetime(data['Ship Date'], errors='coerce')
    return data
```

In [20]:
```
fun()
```

Out[20]:

	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	1582243.50	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
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	19525.82
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	639077.50
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
95	Sub-Saharan Africa	Mali	Clothes	Online	M	2011-07-26	512878119	2011-09-03	888	109.28	35.84	97040.64	31825.92	65214.72
96	Asia	Malaysia	Fruits	Offline	L	2011-11-11	810711038	2011-12-28	6267	9.33	6.92	58471.11	43367.64	15103.47
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	93748.05
98	North America	Mexico	Personal Care	Offline	M	2015-07-30	559427106	2015-08-08	5767	81.73	56.67	471336.91	326815.89	144521.02
99	Sub-Saharan Africa	Mozambique	Household	Offline	L	2012-02-10	665095412	2012-02-15	5367	668.27	502.54	3586605.09	2697132.18	889472.91

100 rows × 14 columns

THE END

In [ ]: