Amazon Sales Data Analysis

With the increasing importance of sales management in commercial and business enterprises, there's a growing need to analyze sales trends and key metrics using data-driven approaches. Extracting, transforming, and loading (ETL) data from Amazon datasets can provide valuable insights into month-wise, year-wise, and yearly-month-wise sales trends. Identifying key metrics and factors and understanding the relationships between attributes can help optimize sales strategies, reduce costs, and increase profits. The objective of this project is to perform ETL processes on Amazon dataset to extract relevant sales data.

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

import warnings
warnings.filterwarnings("ignore")
```

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

In [3]:
```
amazon.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	Total Cost	Total Profit
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	951410.50
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	248406.36
2	Europe	Russia	Office Supplies	Offline	L	5/2/2014	341417157	5/8/2014	1779	651.21	524.96	1158502.59	933903.84	224598.75
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	56065.84	19525.82
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2/1/2013	115456712	2/6/2013	5062	651.21	524.96	3296425.02	2657347.52	639077.50

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

Out[4]:
```
(100, 14)
```

In [5]:
```
## data set contains 100 rows and 14 columns
```

In [6]:
```
## Checking info of data
amazon.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]:
```
##Checking for missing values
amazon.isnull().sum()
```

Out[6]:
```
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 [ ]:
```
## no missing values found
```

In [7]:
```
amazon.describe()
```

Out[7]:

	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 [ ]:
```
## change oder date from object to date
```

In [7]:
```
amazon['Order Date'] = pd.to_datetime(amazon['Order Date'])
```

In [8]:
```
amazon.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    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: datetime64[ns](1), float64(5), int64(2), object(6)
memory usage: 11.1+ KB
```

In [7]:
```
amazon.head()
```

Out[7]:

	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	6/27/2010	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	9/15/2012	2804	205.70	117.11	576782.80	328376.44	248406.36
2	Europe	Russia	Office Supplies	Offline	L	2014-05-02	341417157	5/8/2014	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	7/5/2014	8102	9.33	6.92	75591.66	56065.84	19525.82
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2013-02-01	115456712	2/6/2013	5062	651.21	524.96	3296425.02	2657347.52	639077.50

In [8]:
```
## crearing month column from Order Date
amazon['month'] = amazon['Order Date'].dt.strftime('%b')
```

In [10]:
```
amazon = amazon.sort_values(by = 'month',ascending = True)
```

In [11]:
```
amazon.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	month
17	Sub-Saharan Africa	Cameroon	Beverages	Offline	C	2015-04-01	519820964	4/18/2015	5430	47.45	31.79	257653.50	172619.70	85033.80	Apr
59	Sub-Saharan Africa	Djibouti	Cosmetics	Offline	H	2014-04-07	259353148	4/19/2014	7215	437.20	263.33	3154398.00	1899925.95	1254472.05	Apr
15	Europe	Bulgaria	Clothes	Online	M	2012-04-23	972292029	6/3/2012	1673	109.28	35.84	182825.44	59960.32	122865.12	Apr
70	Asia	Turkmenistan	Office Supplies	Online	M	2013-04-23	462405812	5/20/2013	5010	651.21	524.96	3262562.10	2630049.60	632512.50	Apr
9	Sub-Saharan Africa	Senegal	Cereal	Online	H	2014-04-18	616607081	5/30/2014	6593	205.70	117.11	1356180.10	772106.23	584073.87	Apr

In [9]:
```
## creating year column from Order Date
amazon['year'] = amazon['Order Date'].dt.year
```

In [10]:
```
amazon.head()
```

Out[10]:

	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	month	year
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	2010-05-28	669165933	6/27/2010	9925	255.28	159.42	2533654.00	1582243.50	951410.50	May	2010
1	Central America and the Caribbean	Grenada	Cereal	Online	C	2012-08-22	963881480	9/15/2012	2804	205.70	117.11	576782.80	328376.44	248406.36	Aug	2012
2	Europe	Russia	Office Supplies	Offline	L	2014-05-02	341417157	5/8/2014	1779	651.21	524.96	1158502.59	933903.84	224598.75	May	2014
3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	2014-06-20	514321792	7/5/2014	8102	9.33	6.92	75591.66	56065.84	19525.82	Jun	2014
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2013-02-01	115456712	2/6/2013	5062	651.21	524.96	3296425.02	2657347.52	639077.50	Feb	2013

In [29]:
```
amazon['year'].value_counts()
```

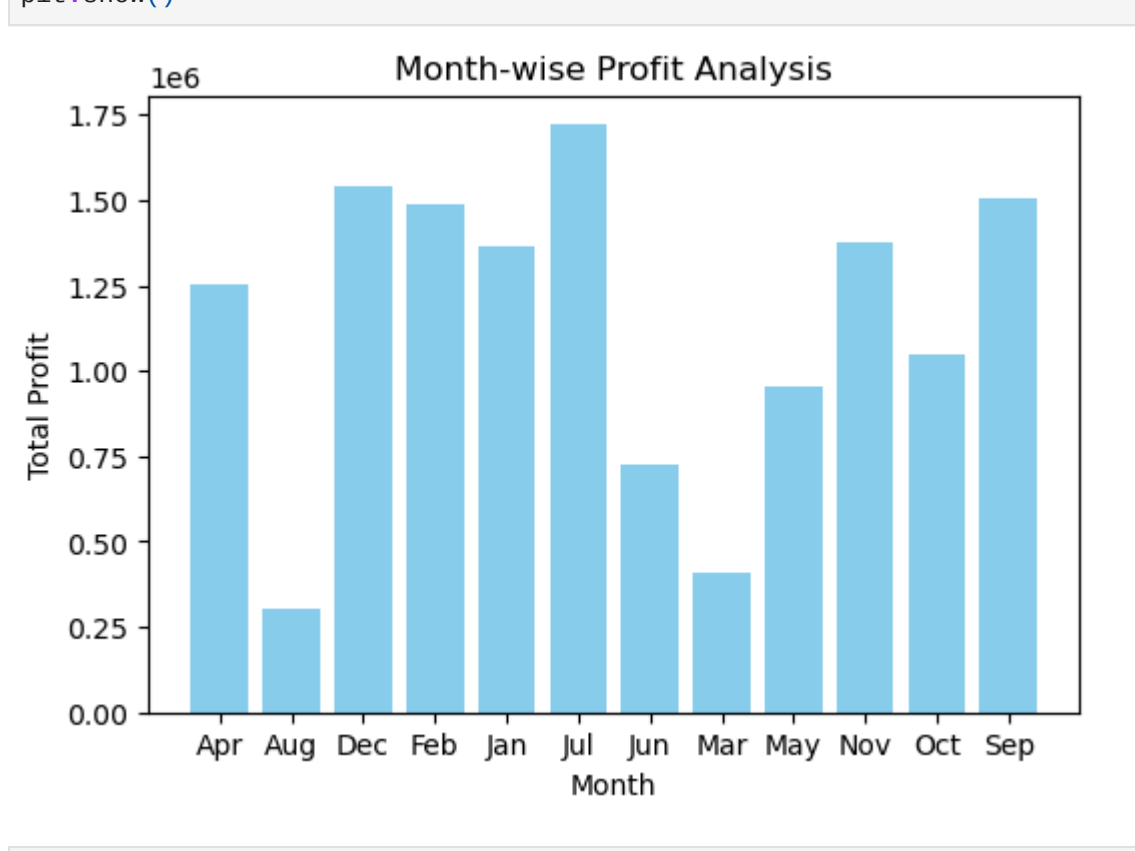
Out[29]:
```
year
2012    22
2014    15
2013    12
2011    12
2015    11
2010    10
2016    10
2017     8
Name: count, dtype: int64
```

In [30]:
```
amazon['month'].value_counts()
```

Out[30]:
```
month
Feb    13
Jul    12
May    11
Oct    11
Jun    10
Apr     9
Nov     9
Jan     7
Dec     5
Sep     5
Aug     4
Mar     4
Name: count, dtype: int64
```

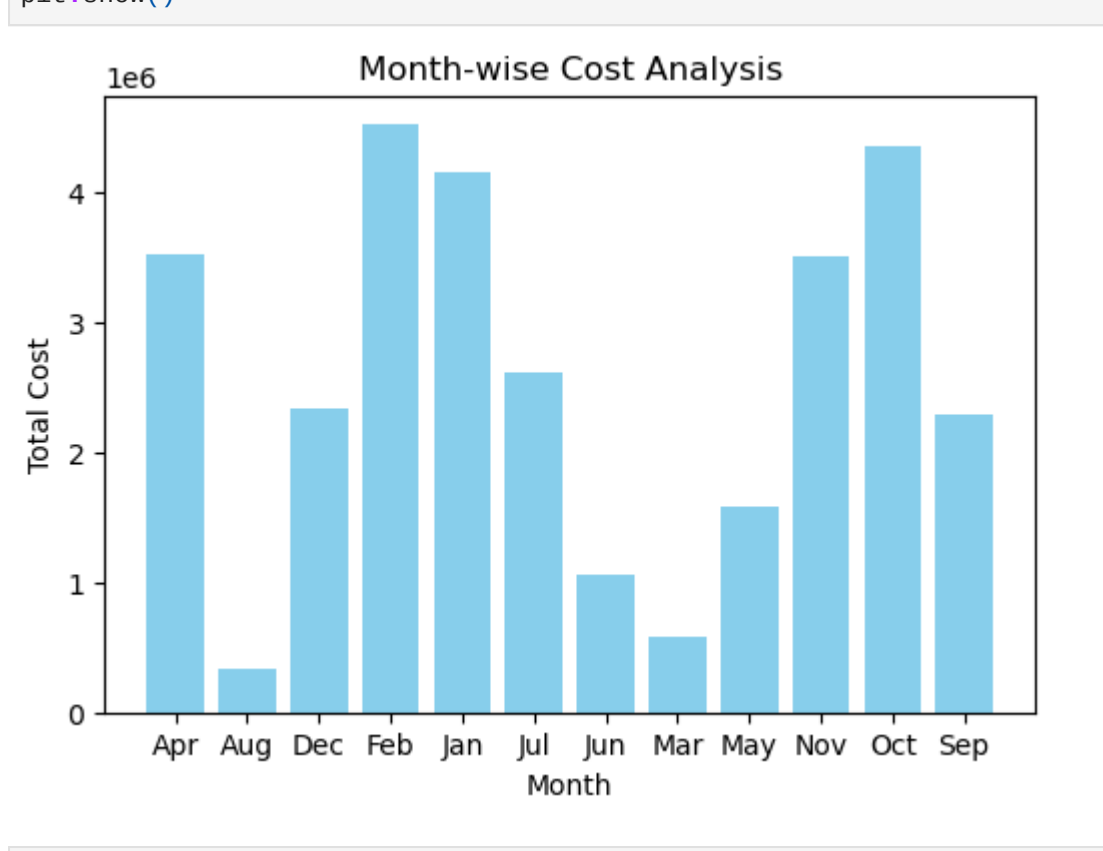
EDA

In [33]:
```
plt.figure(figsize=(6, 4))
plt.bar(amazon['month'], amazon['Total Profit'], color='skyblue')
plt.title('Month-wise Profit Analysis')
plt.xlabel('Month')
plt.ylabel('Total Profit')
plt.show()
```



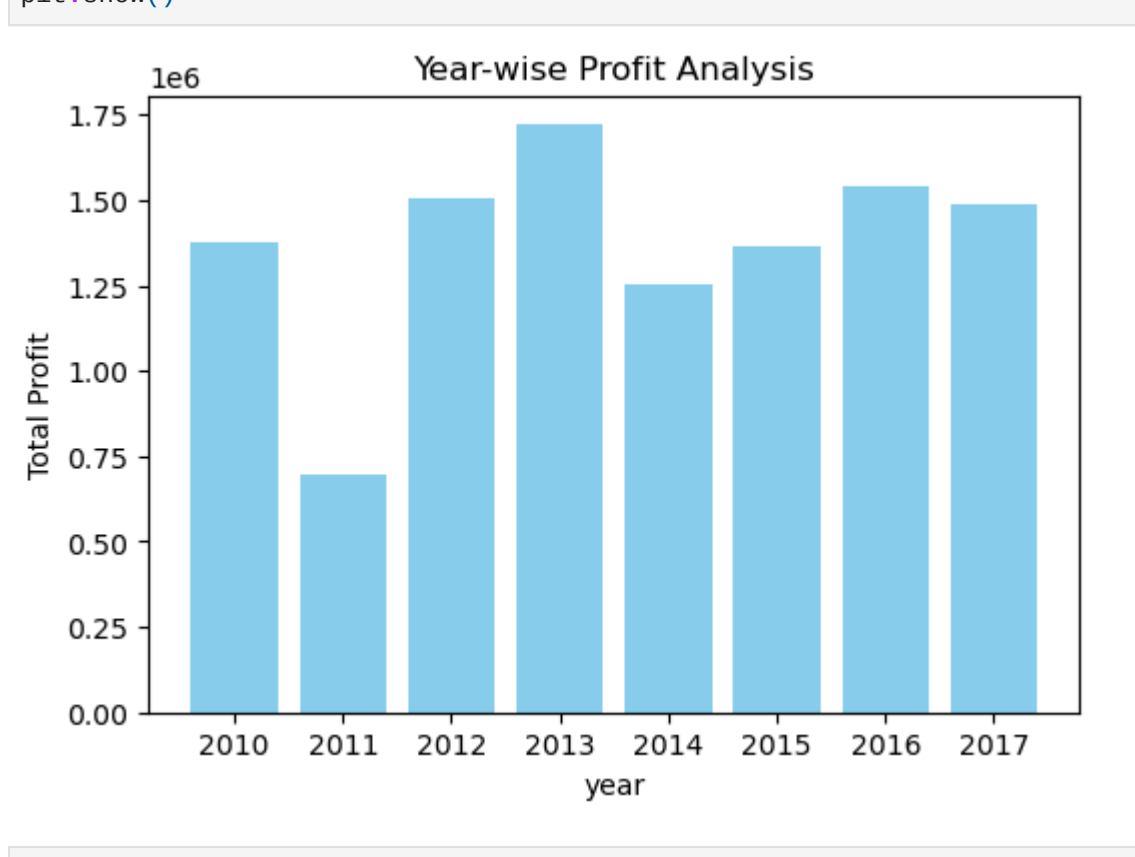
In [ ]:
```
## Highest Profit in the month of July
```

In [31]:
```
plt.figure(figsize=(6, 4))
plt.bar(amazon['month'], amazon['Total Cost'], color='skyblue')
plt.title('Month-wise Cost Analysis')
plt.xlabel('Month')
plt.ylabel('Total Cost')
plt.show()
```



In [ ]:
```
## Highest cost in the month of Feb
```

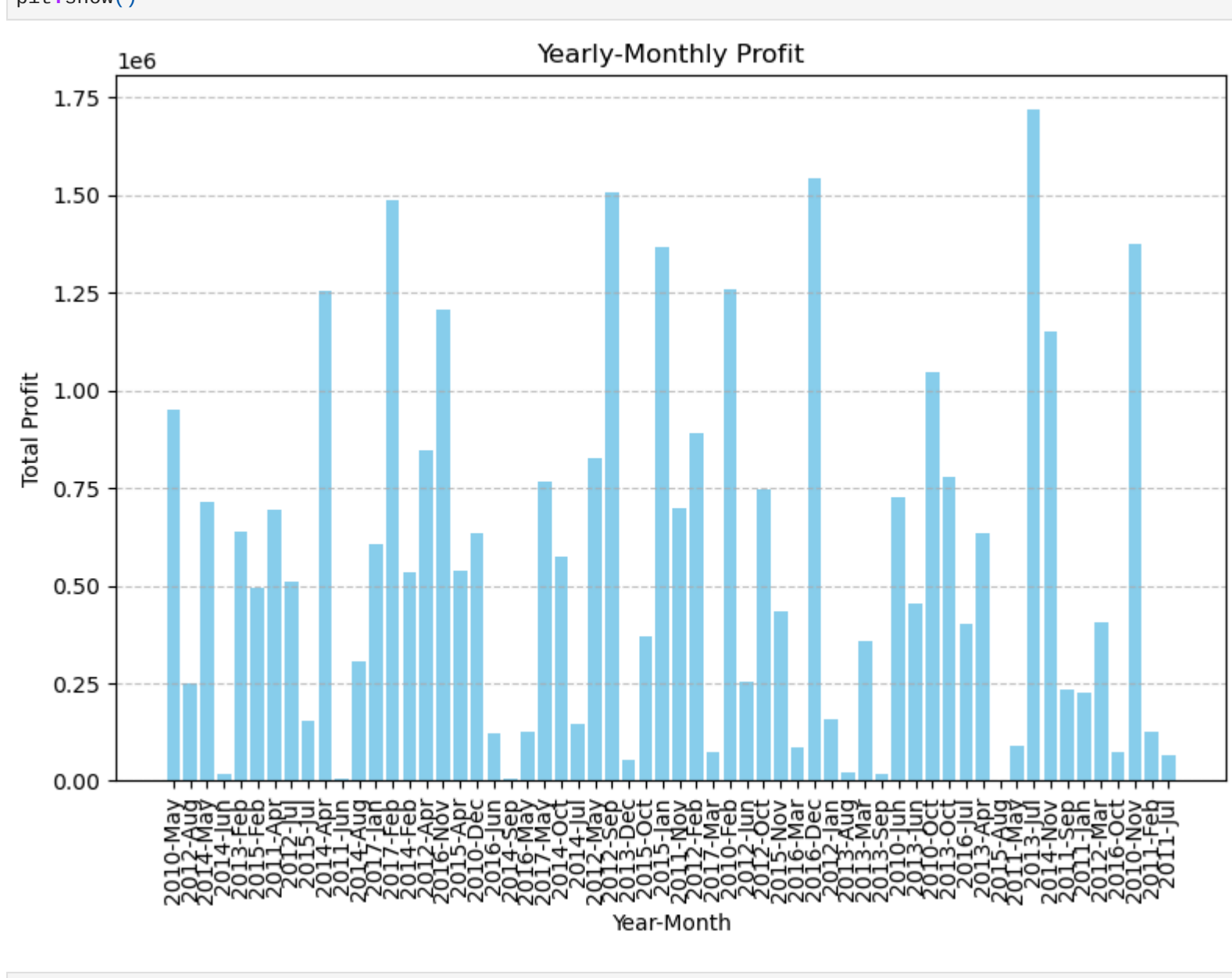
In [34]:
```
plt.figure(figsize=(6, 4))
plt.bar(amazon['year'], amazon['Total Profit'], color='skyblue')
plt.title('Year-wise Profit Analysis')
plt.xlabel('year')
plt.ylabel('Total Profit')
plt.show()
```



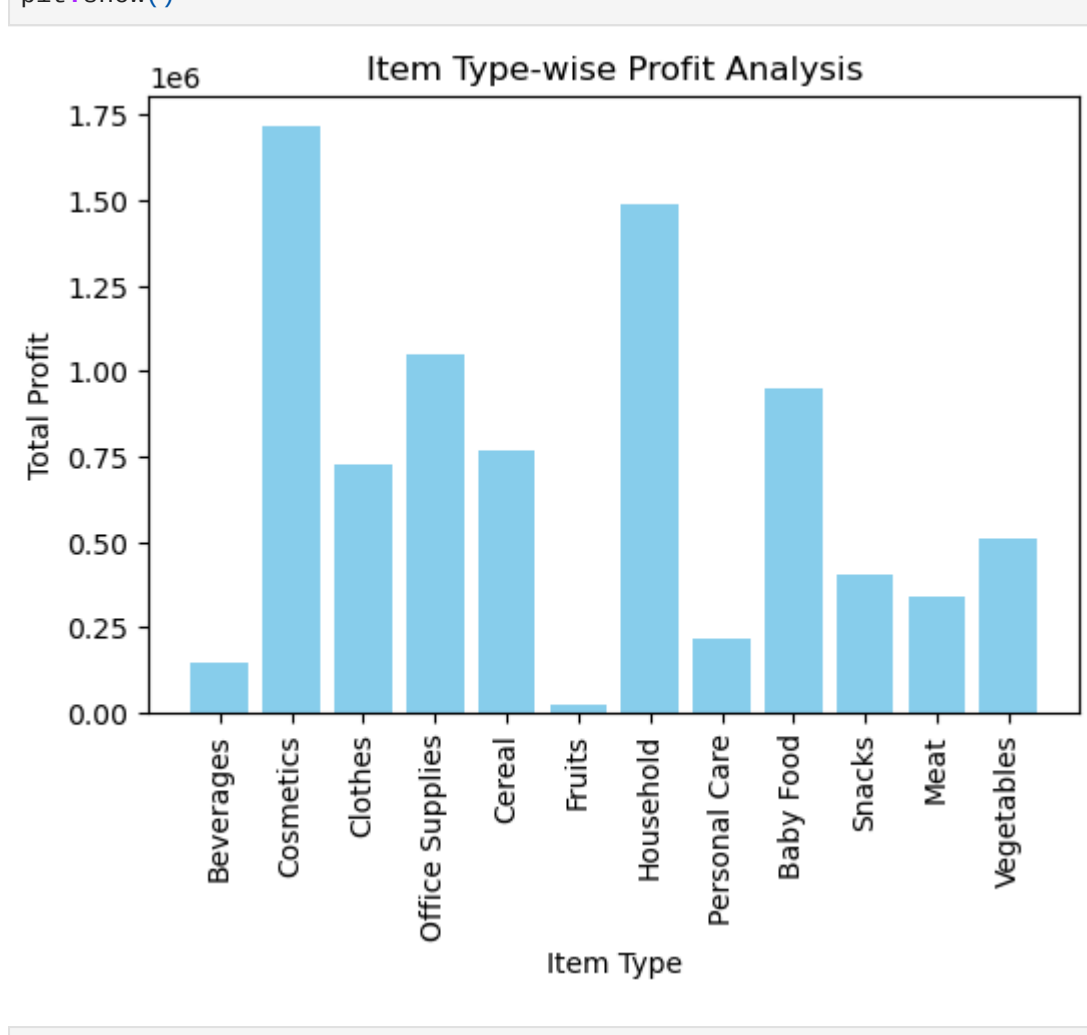
In [35]:
```
## Highest profit in the year of 2013
```

In [14]:
```
amazon['Year_Month'] = amazon['year'].astype(str) + '-' + amazon['month'].astype(str)
```

In [17]:
```
# Plotting
plt.figure(figsize=(8, 6))
plt.bar(amazon['Year_Month'], amazon['Total Profit'], color='skyblue')
plt.title('Yearly-Monthly Profit')
plt.xlabel('Year-Month')
plt.ylabel('Total Profit')
plt.xticks(rotation = 90)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

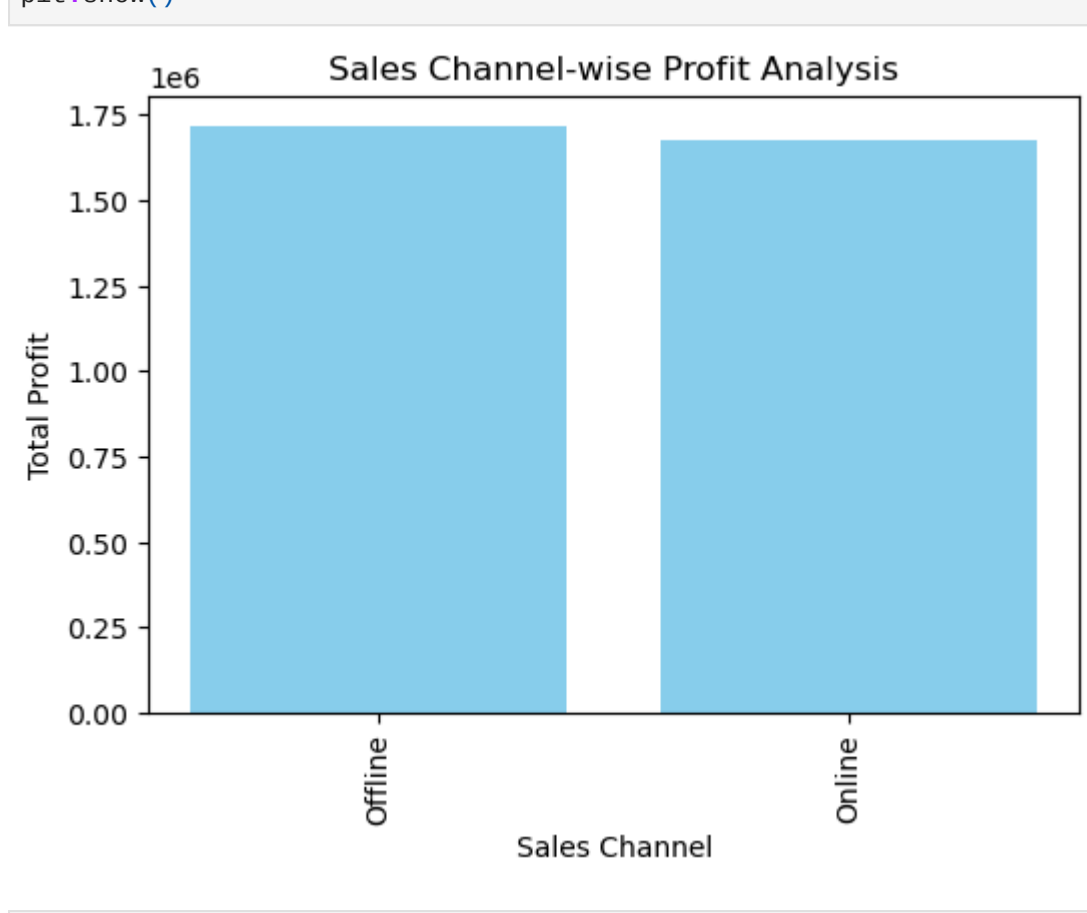


In [29]:
```
plt.figure(figsize=(6, 4))
plt.bar(amazon['Item Type'], amazon['Total Profit'], color='skyblue')
plt.title('Item Type-wise Profit Analysis')
plt.xlabel('Item Type')
plt.xticks(rotation=90)
plt.ylabel('Total Profit')
plt.show()
```

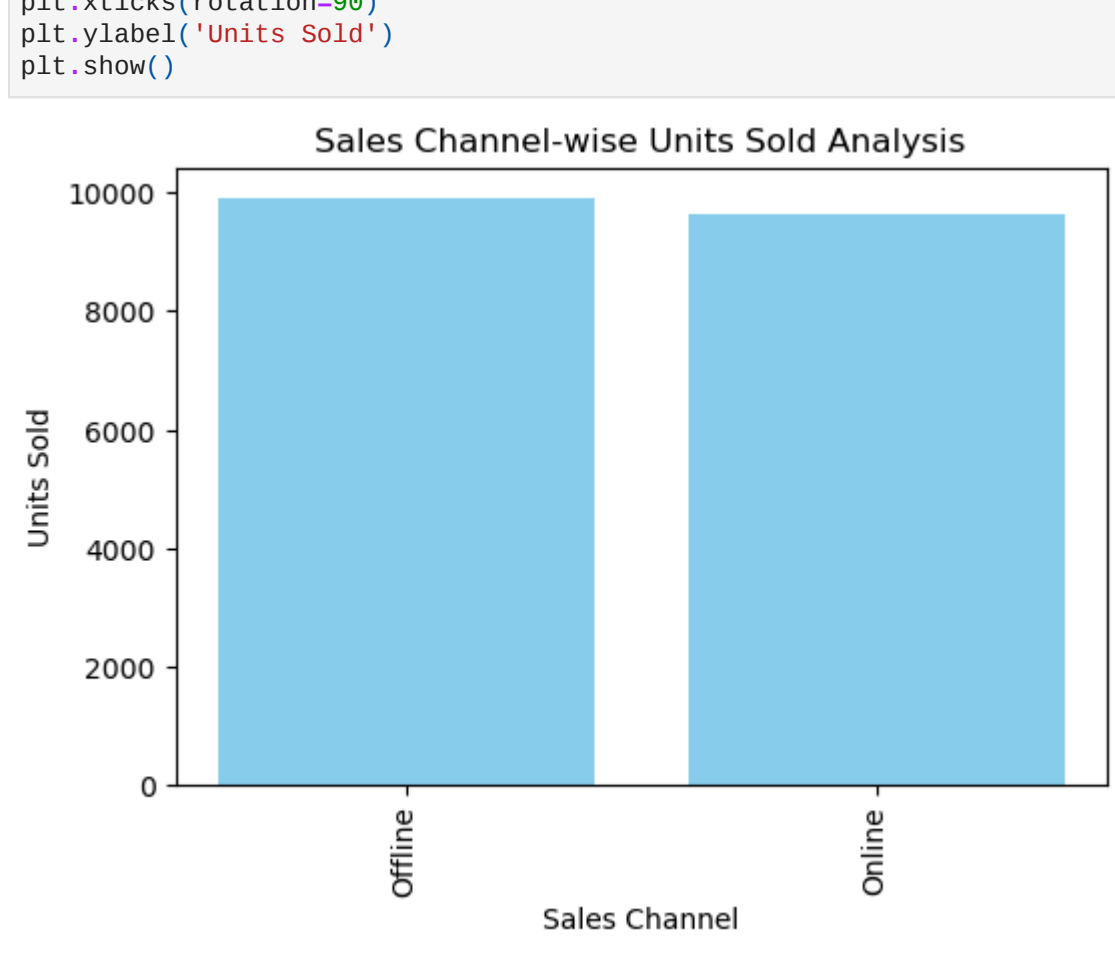


In [37]:
```
## highest profit from Cosmetics
```

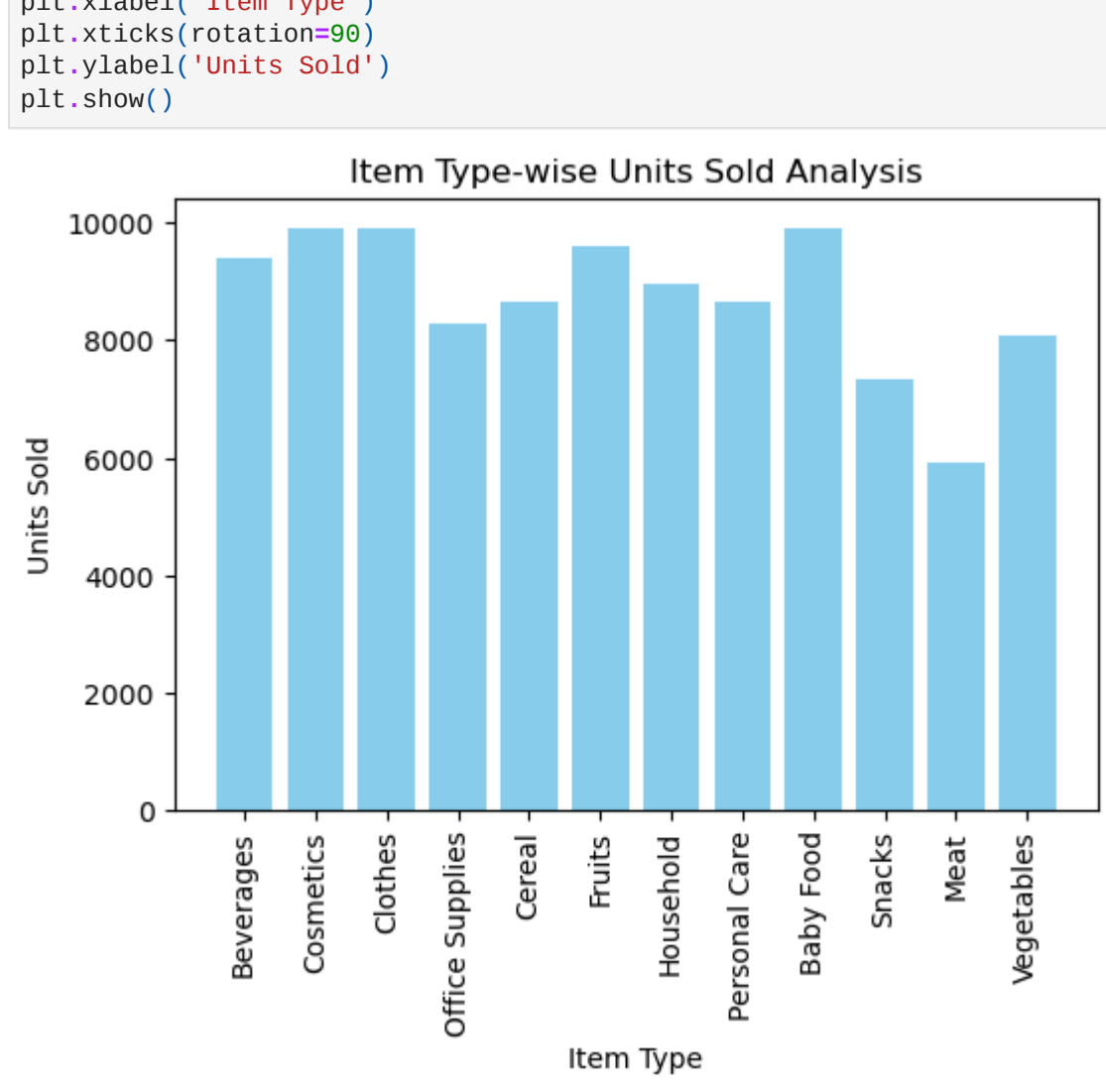
In [5]:
```
plt.figure(figsize=(6, 4))
plt.bar(amazon['Sales Channel'], amazon['Total Profit'], color='skyblue')
plt.title('Sales Channel-wise Profit Analysis')
plt.xlabel('Sales Channel')
plt.xticks(rotation=90)
plt.ylabel('Total Profit')
plt.show()
```



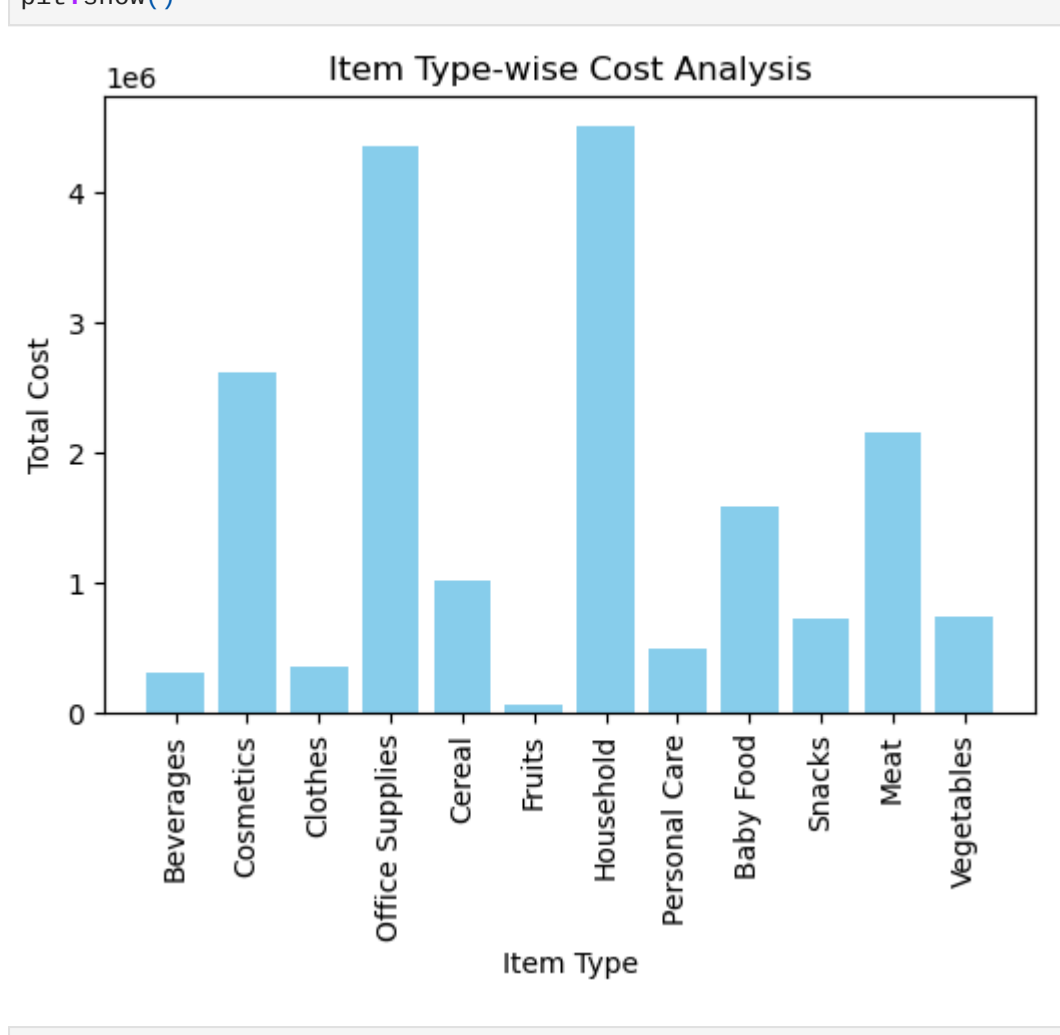
In [7]:
```
plt.figure(figsize=(6, 4))
plt.bar(amazon['Sales Channel'], amazon['Units Sold'], color='skyblue')
plt.title('Sales Channel-wise Units Sold Analysis')
plt.xlabel('Sales Channel')
plt.xticks(rotation=90)
plt.ylabel('Units Sold')
plt.show()
```



In [36]:
```
plt.figure(figsize=(6, 4))
plt.bar(amazon['Item Type'], amazon['Units Sold'], color='skyblue')
plt.title('Item Type-wise Units Sold Analysis')
plt.xlabel('Item Type')
plt.xticks(rotation=90)
plt.ylabel('Units Sold')
plt.show()
```



In [38]:
```
plt.figure(figsize=(6, 4))
plt.bar(amazon['Item Type'], amazon['Total Cost'], color='skyblue')
plt.title('Item Type-wise Cost Analysis')
plt.xlabel('Item Type')
plt.xticks(rotation=90)
plt.ylabel('Total Cost')
plt.show()
```



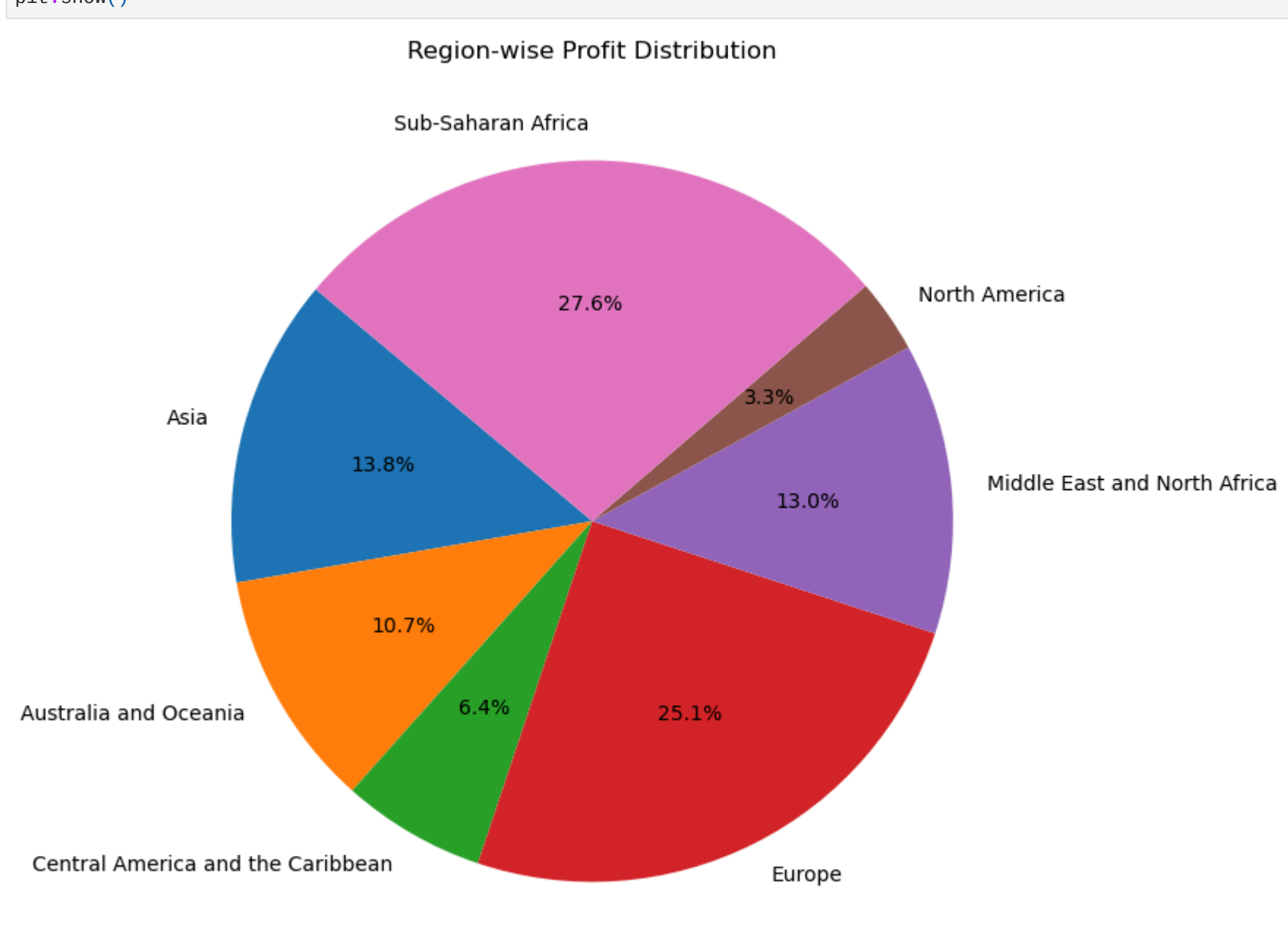
In [15]:
```
Region_Profit = amazon.groupby('Region')['Total Profit'].sum().reset_index()
```

In [16]:
```
Region_Profit
```

Out[16]:

	Region	Total Profit
0	Asia	6113845.87
1	Australia and Oceania	4722160.03
2	Central America and the Caribbean	2846907.85
3	Europe	11082938.63
4	Middle East and North Africa	5761191.86
5	North America	1457942.76
6	Sub-Saharan Africa	12183211.40

In [17]:
```
plt.figure(figsize=(8, 8))
plt.pie(Region_Profit['Total Profit'], labels=Region_Profit['Region'], autopct='%1.1f%%', startangle=140)
plt.title('Region-wise Profit Distribution')
plt.show()
```



In [ ]:
```
## 27.6% of profit from Sub-Saharan Africa
```

In [40]:
```
Country_Profit = amazon.groupby('Country')['Total Profit'].sum().reset_index()
```

In [54]:
```
Country_Profit = Country_Profit.sort_values(by='Total Profit', ascending=False)
```

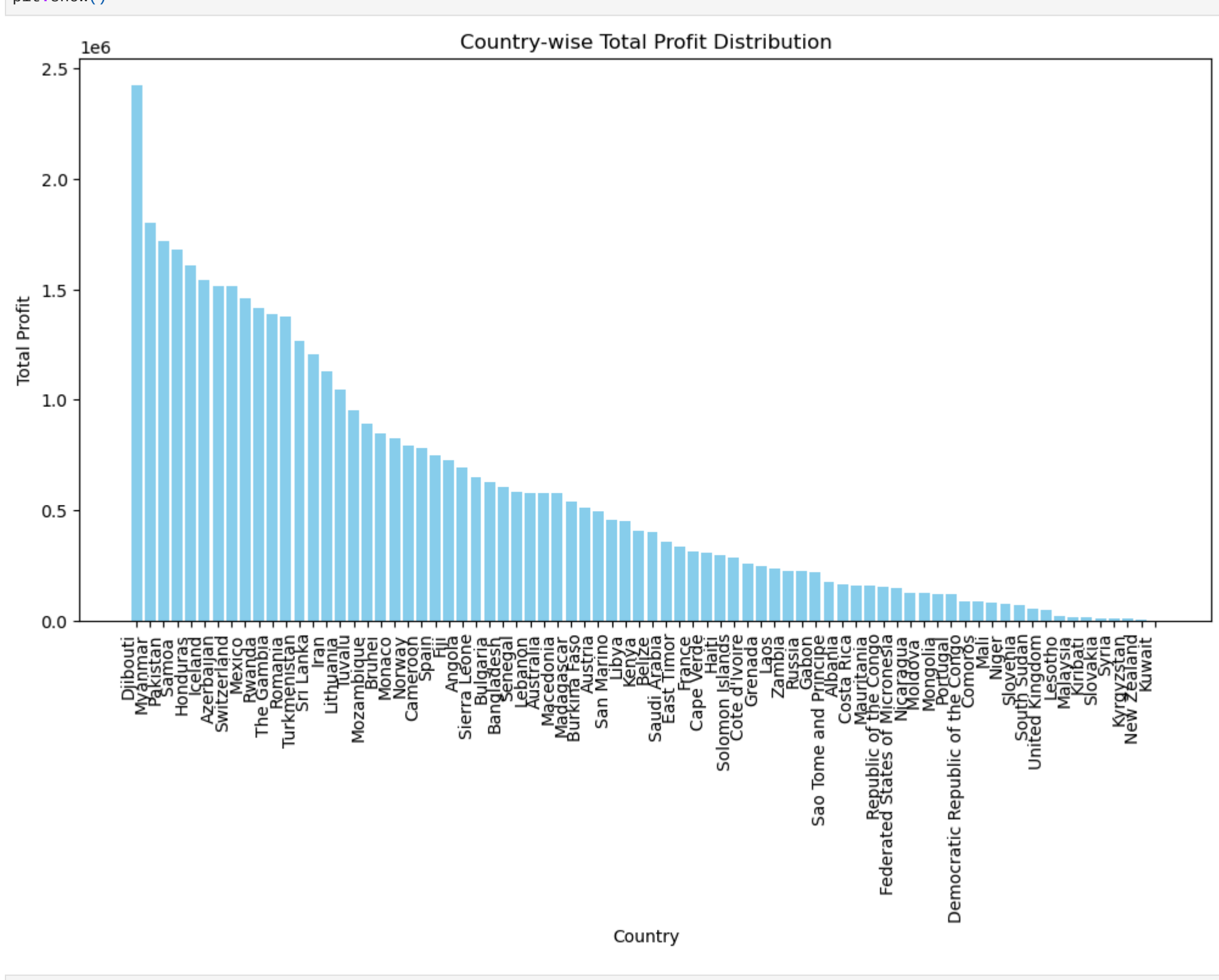
In [55]:
```
pd.set_option("display.max_rows",100)
```

In [56]:
```
Country_Profit
```

Out[56]:

	Country	Total Profit
16	Djibouti	2425317.87
46	Myanmar	1802771.70
51	Pakistan	1719922.04
57	Samoa	1678540.98
24	Honduras	1609947.52
25	Iceland	1541705.29
4	Azerbaijan	1512926.83
69	Switzerland	1512729.45
41	Mexico	1457942.76
56	Rwanda	1417493.49
71	The Gambia	1385883.27
54	Romania	1375311.70
72	Turkmenistan	1267258.40
68	Sri Lanka	1208744.24
26	Iran	1128242.43
35	Lithuania	1046233.75
73	Tuvalu	951410.50
45	Mozambique	889472.91
7	Brunei	846885.00
43	Monaco	825738.04
50	Norway	794398.84
10	Cameroon	781681.30
67	Spain	747939.49
19	Fiji	727423.20
1	Angola	693911.51
62	Sierra Leone	649879.30
8	Bulgaria	626223.87
5	Bangladesh	606834.72
61	Senegal	584073.87
32	Lebanon	579000.96
2	Australia	576605.12
36	Macedonia	575916.48
37	Madagascar	539196.48
9	Burkina Faso	510216.66
3	Austria	495007.89
58	San Marino	455335.00
34	Libya	450780.97
27	Kenya	407630.41
6	Belize	403773.12
60	Saudi Arabia	359941.17
17	East Timor	337937.60
20	France	315574.05
11	Cape Verde	306097.92
23	Haiti	296448.35
65	Solomon Islands	285087.64
14	Cote d'Ivoire	255718.08
22	Grenada	248406.36
31	Laos	235601.16
75	Zambia	225246.90
55	Russia	224598.75
21	Gabon	216919.36
59	Sao Tome and Principe	176193.13
0	Albania	166635.36
13	Costa Rica	160609.54
40	Mauritania	159832.50
53	Republic of the Congo	152114.20
18	Federated States of Micronesia	146875.14
48	Nicaragua	127722.96
42	Moldova	127054.20
44	Mongolia	122819.06
52	Portugal	122029.78
15	Democratic Republic of the Congo	89904.06
12	Comoros	85223.58
39	Mali	79245.74
49	Niger	75555.90
64	Slovenia	72975.60
66	South Sudan	53252.50
74	United Kingdom	46735.86
33	Lesotho	23150.46
38	Malaysia	15103.47
28	Kiribati	13009.18
63	Slovakia	10795.23
70	Syria	9119.44
30	Kyrgyzstan	7828.12
47	New Zealand	5270.67
29	Kuwait	1258.02

In [57]:
```
plt.figure(figsize=(12, 6))
plt.bar(Country_Profit['Country'], Country_Profit['Total Profit'], color='skyblue')
plt.xlabel('Country')
plt.ylabel('Total Profit')
plt.title('Country-wise Total Profit Distribution')
plt.xticks(rotation=90, ha='right')
plt.show()
```



In [58]:
```
## highest profit from Djibouti Country
```

In [ ]: