```
In [51]: import pandas as pd
         import numpy as np
         from matplotlib import pyplot as plt
         dataframe = pd.read_csv('P1 Amazon Sales data.csv')

         dataframe['month']= pd.DatetimeIndex(dataframe['Order Date']).month
         month=dataframe.groupby('month')['Total Revenue'].mean().reset_index()
         print(month)
         monthvalue=month['month']
         revenue=month['Total Revenue']
         plt.subplots(figsize=(20,8))
         plt.plot(monthvalue,revenue,color="red",linewidth=3,label="Month-wise Sales Trend")
         plt.xlabel("Month")
         plt.ylabel("Total Revenue")
         plt.legend()
         plt.show()
```

```
     month  Total Revenue
0        1   1.497495e+06
1        2   1.903117e+06
2        3   5.687060e+05
3        4   1.798576e+06
4        5   1.201431e+06
5        6   5.230326e+05
6        7   1.305793e+06
7        8   2.820412e+05
8        9   1.062953e+06
9       10   1.389780e+06
10      11   2.285358e+06
11      12   1.449892e+06
```
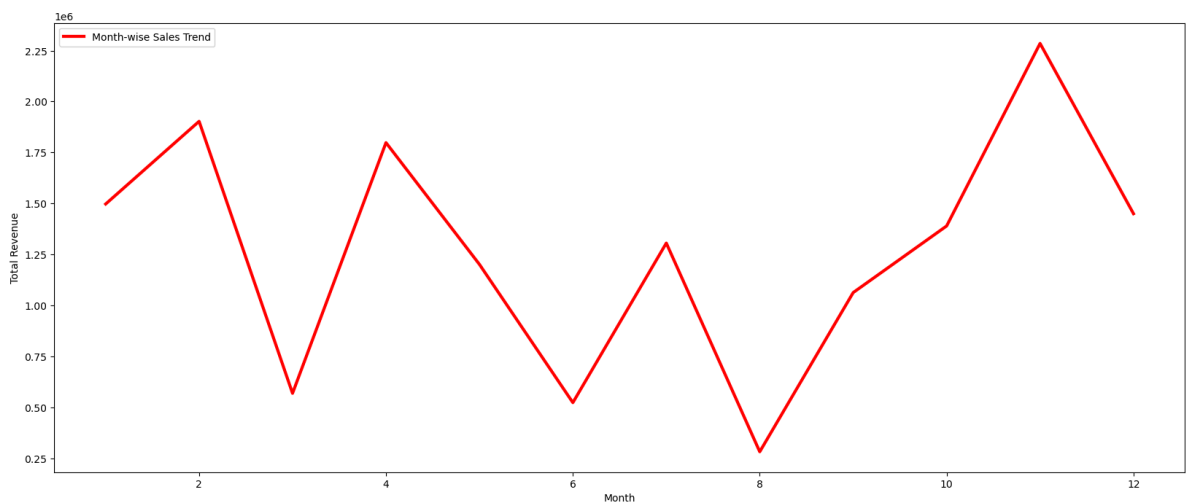


```
In [ ]: Summary:The highest revenue was generated in month of November 2.285358e+06
```

```
In [37]: dataframe['year']= pd.DatetimeIndex(dataframe['Order Date']).year

         year=dataframe.groupby('year')['Total Revenue'].mean().reset_index()
         print(year)
         yearvalue=year['year']
         revenue=year['Total Revenue']
         plt.plot(yearvalue,revenue,linewidth=2.0,label="Year-wise Sales Trend")
         plt.xlabel("Year")
         plt.ylabel("Total Revenue")
         plt.legend()
         plt.show()
```

```
      year   Total Revenue
0     2010   1.918602e+06
1     2011   9.274305e+05
2     2012   1.449938e+06
3     2013   1.694204e+06
4     2014   1.108681e+06
5     2015   1.129817e+06
6     2016   1.237287e+06
7     2017   1.671677e+06
```


In [ ]: Summary:The highest revenue was generated **in** the year 2010

In [41]:
```python
dataframe['Order Date'] = pd.to_datetime(dataframe['Order Date'])
dataframe['YearMonth'] = dataframe['Order Date'].dt.to_period('M')

by_yearmonth = dataframe.groupby('YearMonth')['Total Revenue'].mean().reset_index()
by_yearmonth['YearMonth'] = by_yearmonth['YearMonth'].dt.to_timestamp()
print(by_yearmonth)
year_month =by_yearmonth['YearMonth'].values
revenue =by_yearmonth['Total Revenue'].values

plt.figure(figsize=(15,6))
plt.plot(year_month, revenue, color="green", label="Yearly Month-wise Sales Trend")
plt.xlabel('Year-Month')
plt.ylabel('Total Revenue')
plt.title('Yearly Month-wise Sales Trend')
plt.legend()
plt.show()
```

```
    YearMonth   Total Revenue
0   2010-02-01  1.705331e+06
1   2010-05-01  1.293987e+06
2   2010-06-01  1.082418e+06
3   2010-10-01  3.032467e+06
4   2010-11-01  3.458252e+06
5   2010-12-01  1.290893e+06
6   2011-01-01  5.211127e+05
7   2011-02-01  3.870022e+05
8   2011-04-01  2.798046e+06
9   2011-05-01  2.724105e+05
10  2011-06-01  1.910344e+04
11  2011-07-01  9.704064e+04
12  2011-09-01  5.749519e+05
13  2011-11-01  1.484596e+06
14  2012-01-01  5.064420e+05
15  2012-02-01  2.235950e+06
16  2012-03-01  9.947654e+05
17  2012-04-01  1.518671e+06
18  2012-05-01  1.891391e+06
19  2012-06-01  7.106918e+05
20  2012-07-01  1.481698e+06
21  2012-08-01  5.767828e+05
22  2012-09-01  2.324076e+06
23  2012-10-01  1.521123e+06
24  2013-02-01  3.296425e+06
25  2013-03-01  8.357591e+05
26  2013-04-01  3.262562e+06
27  2013-06-01  6.764337e+05
28  2013-07-01  4.272756e+06
29  2013-08-01  8.962398e+04
30  2013-09-01  7.125321e+04
31  2013-10-01  1.351385e+06
32  2013-12-01  1.736762e+05
33  2014-02-01  9.098301e+05
34  2014-04-01  2.255289e+06
35  2014-05-01  1.530169e+06
36  2014-06-01  7.559166e+04
37  2014-07-01  3.443209e+05
38  2014-08-01  4.554790e+05
39  2014-09-01  2.040471e+04
40  2014-10-01  4.507902e+05
41  2014-11-01  4.647150e+06
42  2015-01-01  5.513228e+06
43  2015-02-01  1.001956e+06
44  2015-04-01  5.299936e+05
45  2015-07-01  4.308031e+05
46  2015-08-01  6.279090e+03
47  2015-10-01  1.904138e+06
48  2015-11-01  6.480304e+05
49  2016-03-01  1.978834e+05
50  2016-05-01  4.143711e+05
51  2016-06-01  2.841348e+05
52  2016-07-01  6.008214e+05
53  2016-10-01  2.211170e+05
54  2016-11-01  2.938203e+06
55  2016-12-01  2.247000e+06
56  2017-01-01  1.457065e+06
57  2017-02-01  3.557504e+06
58  2017-03-01  2.464160e+05
59  2017-05-01  1.032622e+06
```

Yearly Month-wise Sales Trend
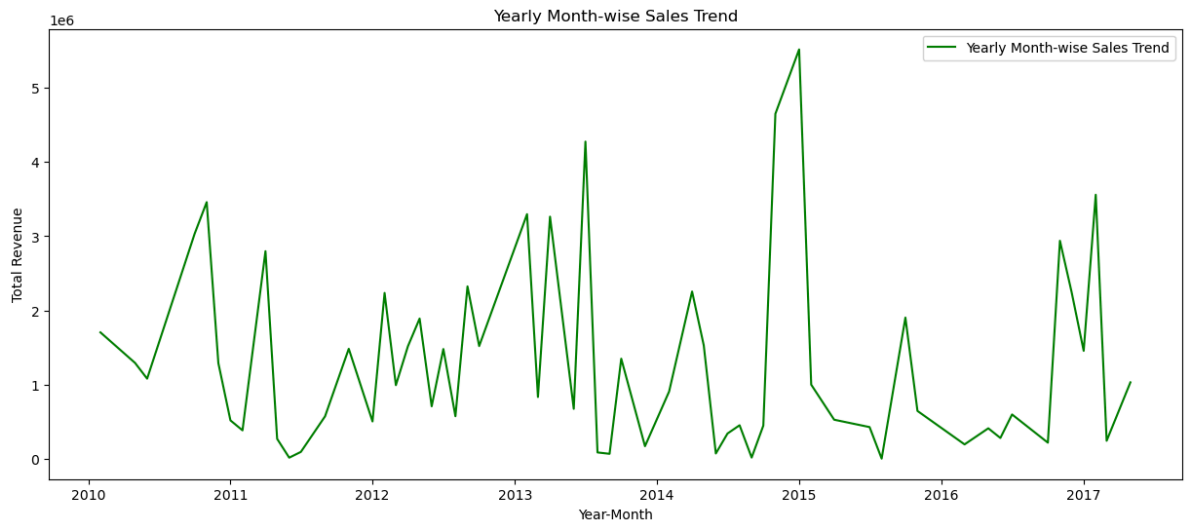
In [ ]: Summary:The average revenue generated was 2M

```
In [43]: revenue_item = dataframe.groupby('Item Type')['Total Revenue'].mean().reset_index()
         cost_item = dataframe.groupby('Item Type')['Total Cost'].mean().reset_index()

         label1= revenue_item['Total Revenue'].values
         label2= cost_item['Total Cost'].values
         item = revenue_item['Item Type']

         plt.figure(figsize=(10, 6))
         plt.plot(item,label1,color="blue", label="Total Revenue vs Item Type")
         plt.plot(item,label2,color="red", label="Total Cost vs Item Type")
         plt.xlabel('Item Type')
         plt.xticks(rotation=30)
         plt.ylabel('Total Revenue')
         plt.title('Item Type vs Total Revenue')
         plt.title('Item Type vs Total Cost')
         plt.legend()
         plt.show()
```
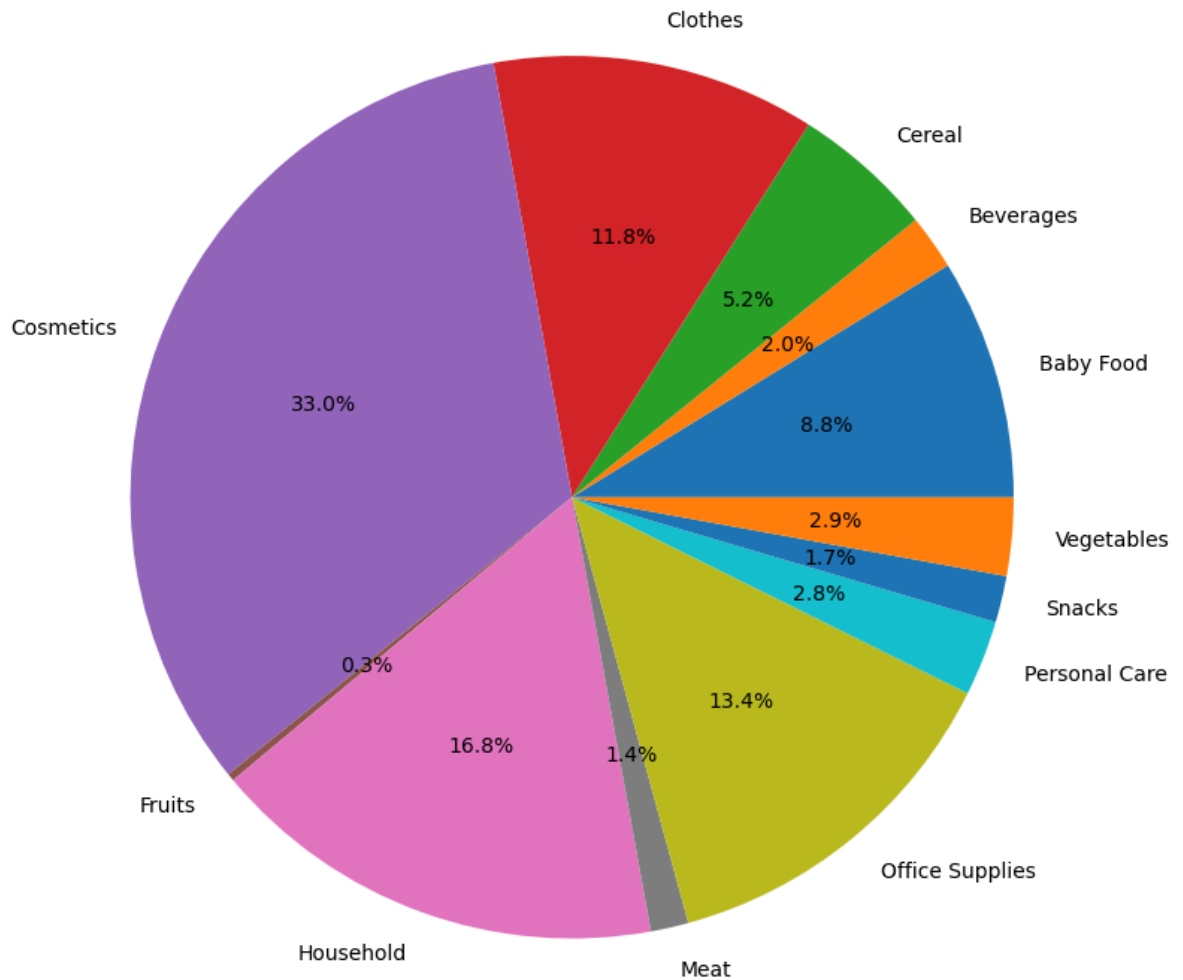
Item Type vs Total Cost
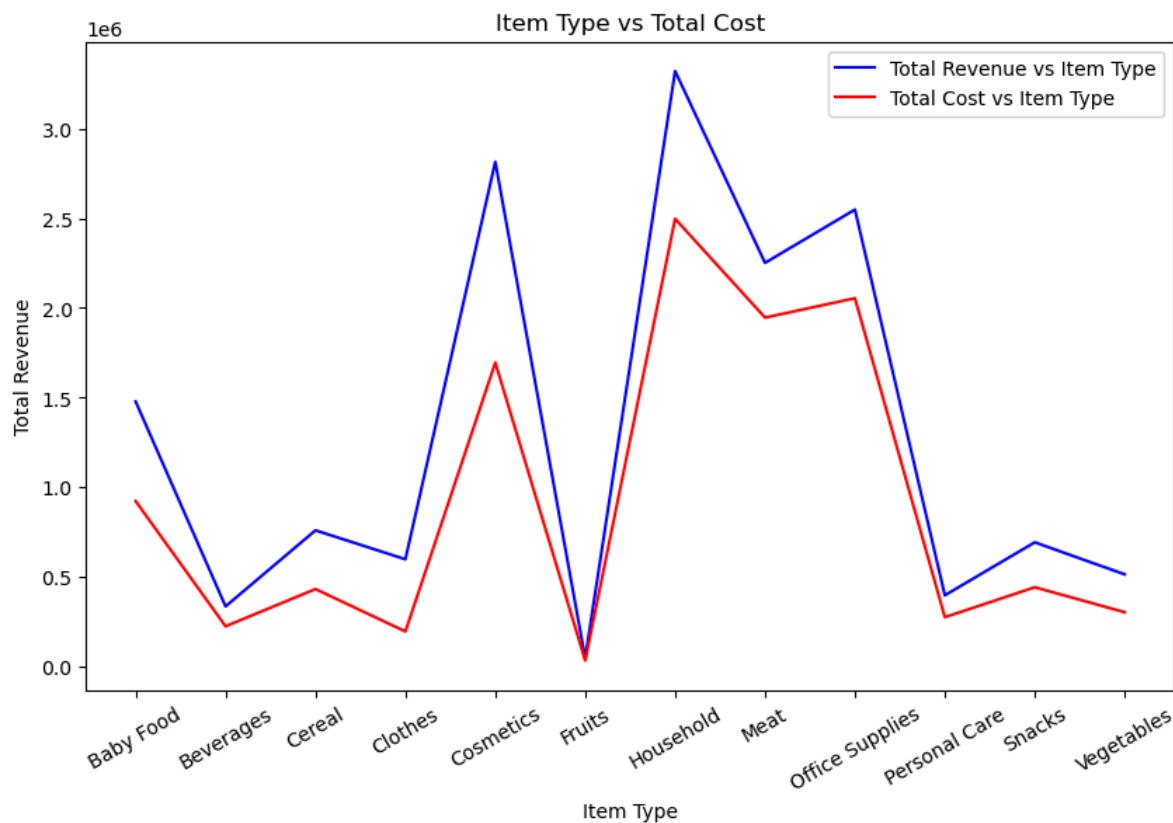
In [ ]: Summary:

```
In [45]: item_profit= dataframe.groupby('Item Type')['Total Profit'].sum().reset_index()
         print(item_profit)
         value= item_profit['Total Profit']
         label= item_profit['Item Type']
         plt.pie(value,labels= label, autopct='%0.1f%%', radius=2)
         plt.show()
```

```
         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
```

Clothes — 11.8%
Cereal — 5.2%
Beverages — 2.0%
Baby Food — 8.8%
Vegetables — 2.9%
Snacks — 1.7%
Personal Care — 2.8%
Office Supplies — 13.4%
Meat — 1.4%
Household — 16.8%
Fruits — 0.3%
Cosmetics — 33.0%

In [ ]:
Summary:Cosmetics generated the highest profit of 33% whereas fruits generated only 0

In [47]:
```python
Region_profit= dataframe.groupby('Region')['Total Profit'].sum().reset_index()
print(Region_profit)
fig=plt.subplots(figsize=(17,8))
plt.title("Total Profit by Region",fontsize=20)
plt.xlabel("Region",fontsize=16)
plt.xticks(rotation=45)
plt.ylabel("Total Profit", fontsize=16)

label=Region_profit['Region']
value=Region_profit['Total Profit']
plt.bar(label, value, width=0.4, color=('red','blue','green','yellow','orange','viole
plt.show()
```

```
                            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
```
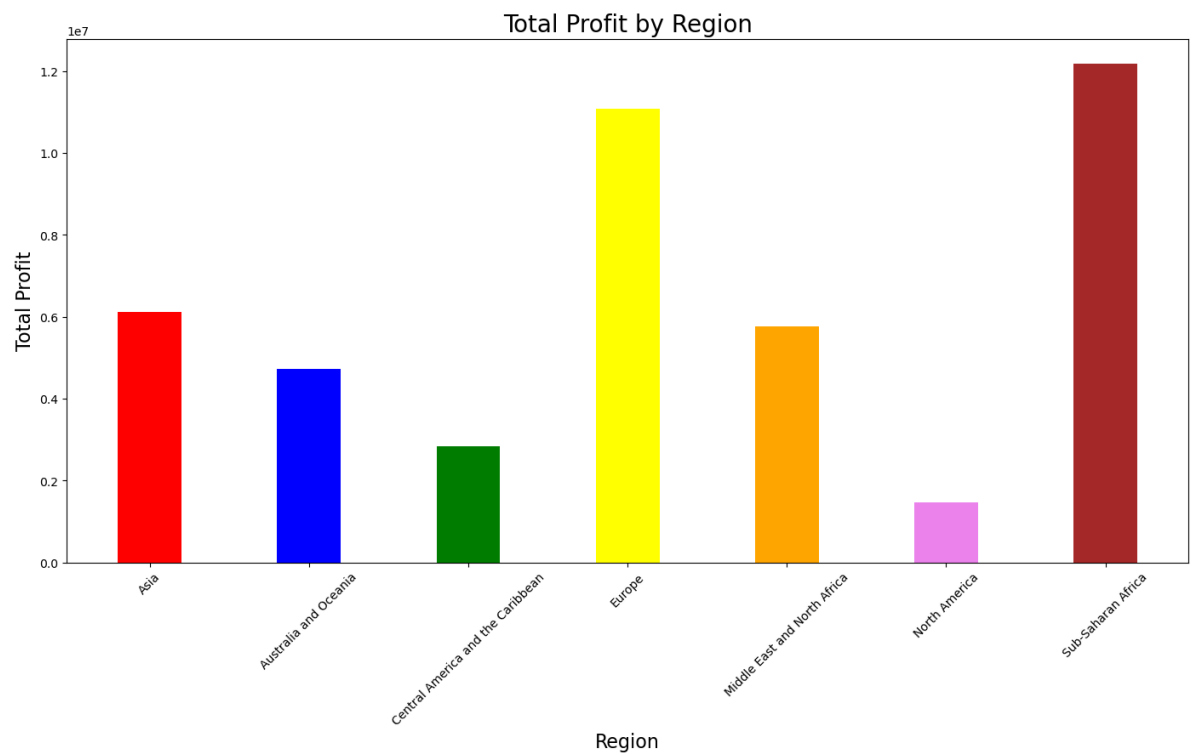
Total Profit by Region

Summary:Sub-Saharan African region was the highest profited region **and** North America

```python
grp_channel = dataframe.groupby('Sales Channel')['Units Sold'].sum().reset_index()
print(grp_channel)
value=grp_channel['Sales Channel']
label = grp_channel['Units Sold']

plt.pie(label, labels=value, autopct='%1.1f%%',startangle=90, wedgeprops={'linewidth'
centre_circle = plt.Circle((0,0),0.70,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.show()
```

```
  Sales Channel  Units Sold
0       Offline      276782
1        Online      236089
```
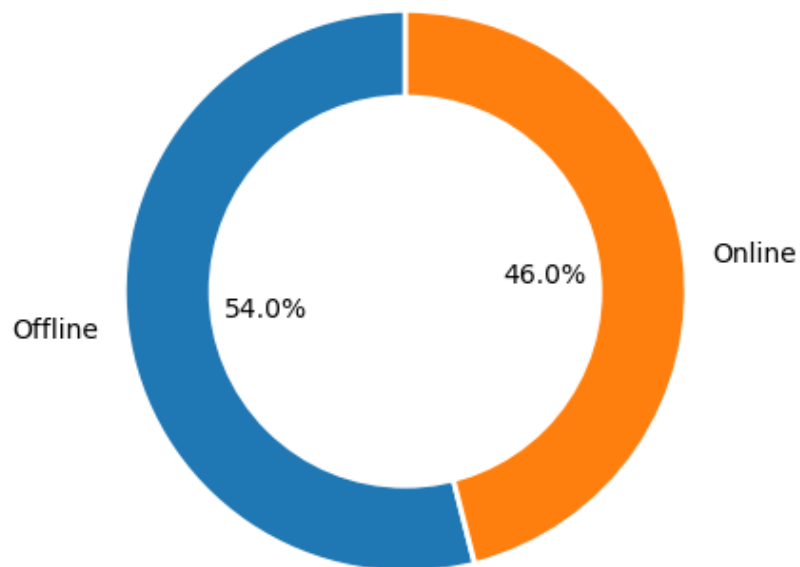
In [ ]: `Summary:Offline sales sold 54% units and online could sale 46% of items.`

In [ ]: