unified mentor data science internship

Amazon sales data analysis project

NAME:- PANKAJ VARSHNEY

In [29]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [30]:
```
#Load the Dataset
data=pd.read_csv("C:/Users/himan/Downloads/Amazon Sales data.csv")
```

In [31]:
```
data
```

Out[31]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Uni Sol
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	5/28/2010	669165933	6/27/2010	992
1	Central America and the Caribbean	Grenada	Cereal	Online	C	8/22/2012	963881480	9/15/2012	280
2	Europe	Russia	Office Supplies	Offline	L	5/2/2014	341417157	5/8/2014	177
3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	6/20/2014	514321792	7/5/2014	810
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2/1/2013	115456712	2/6/2013	506
...	...	...	...	...	...	...	...	...	
95	Sub-Saharan Africa	Mali	Clothes	Online	M	7/26/2011	512878119	9/3/2011	88
96	Asia	Malaysia	Fruits	Offline	L	11/11/2011	810711038	12/28/2011	626
97	Sub-Saharan Africa	Sierra Leone	Vegetables	Offline	C	6/1/2016	728815257	6/29/2016	148
98	North America	Mexico	Personal Care	Offline	M	7/30/2015	559427106	8/8/2015	576
99	Sub-Saharan Africa	Mozambique	Household	Offline	L	2/10/2012	665095412	2/15/2012	536

100 rows × 14 columns

In [60]:
```
data.columns
```

Out[60]:
```
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', 'Year',
       'Month'],
      dtype='object')
```

In [58]:
```
#Descriptive Statistics
data.describe()
```

Out[58]:

	Order Date	Order ID	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	
count	100	1.000000e+02	100.000000	100.000000	100.000000	1.000000e+02	1.000000e+02	1
mean	2013-09-16 14:09:36	5.550204e+08	5128.710000	276.761300	191.048000	1.373488e+06	9.318057e+05	4
min	2010-02-02 00:00:00	1.146066e+08	124.000000	9.330000	6.920000	4.870260e+03	3.612240e+03	1
25%	2012-02-14 12:00:00	3.389225e+08	2836.250000	81.730000	35.840000	2.687212e+05	1.688680e+05	1
50%	2013-07-12 12:00:00	5.577086e+08	5382.500000	179.880000	107.275000	7.523144e+05	3.635664e+05	2
75%	2015-04-07 00:00:00	7.907551e+08	7369.000000	437.200000	263.330000	2.212045e+06	1.613870e+06	6
max	2017-05-22 00:00:00	9.940222e+08	9925.000000	668.270000	524.960000	5.997055e+06	4.509794e+06	1
std	NaN	2.606153e+08	2794.484562	235.592241	188.208181	1.460029e+06	1.083938e+06	4

In [32]:
```
#Check the missing values
data.isnull().sum()
```

Out[32]:
```
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
```

there is no missing values or null values in the data our data is already clean

In [33]:
```
#convert order date to datetime
data["Order Date"]=pd.to_datetime(data["Order Date"])
```

In [34]:
```
#extract year and month from order date
data["Year"]=data["Order Date"].dt.year
data["Month"]=data["Order Date"].dt.month
data.head()
```

Out[34]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	2010-05-28	669165933	6/27/2010	9925	255.28	15
1	Central America and the Caribbean	Grenada	Cereal	Online	C	2012-08-22	963881480	9/15/2012	2804	205.70	11
2	Europe	Russia	Office Supplies	Offline	L	2014-05-02	341417157	5/8/2014	1779	651.21	52
3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	2014-06-20	514321792	7/5/2014	8102	9.33	
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2013-02-01	115456712	2/6/2013	5062	651.21	52

In [35]:
```
#calculate the number of regions
regions=data["Region"].nunique()
print("Number of Regions:",regions)
```

```
Number of Regions: 7
```

In [36]:
```
#calculate the number of countries
country=data["Country"].nunique()
print("Number of countries:",country)
```

```
Number of countries: 76
```

In [37]:
```
#calculate the item types
item_type=data["Item Type"].nunique()
print("Number of Item Types:",item_type)
```

```
Number of Item Types: 12
```

In [38]:
```
#calculate the unit sold
unit_sold=data["Units Sold"].sum()
print("Total Units Sold:",unit_sold)
```

```
Total Units Sold: 512871
```

In [39]:
```
#calculate the unit cost
unit_cost=data["Unit Cost"].sum()
print("Total Units Cost:",unit_cost)
```

```
Total Units Cost: 19104.8
```

In [40]:
```
#calculate the total revenue
total_revenue=data["Total Revenue"].sum()
```

```
print("Total Revenue:",total_revenue)
```

```
Total Revenue: 137348768.31
```

In [41]:
```
#calculate the total cost
total_cost=data["Total Cost"].sum()
print("Total Cost:",total_cost)
```

```
Total Cost: 93180569.91000001
```

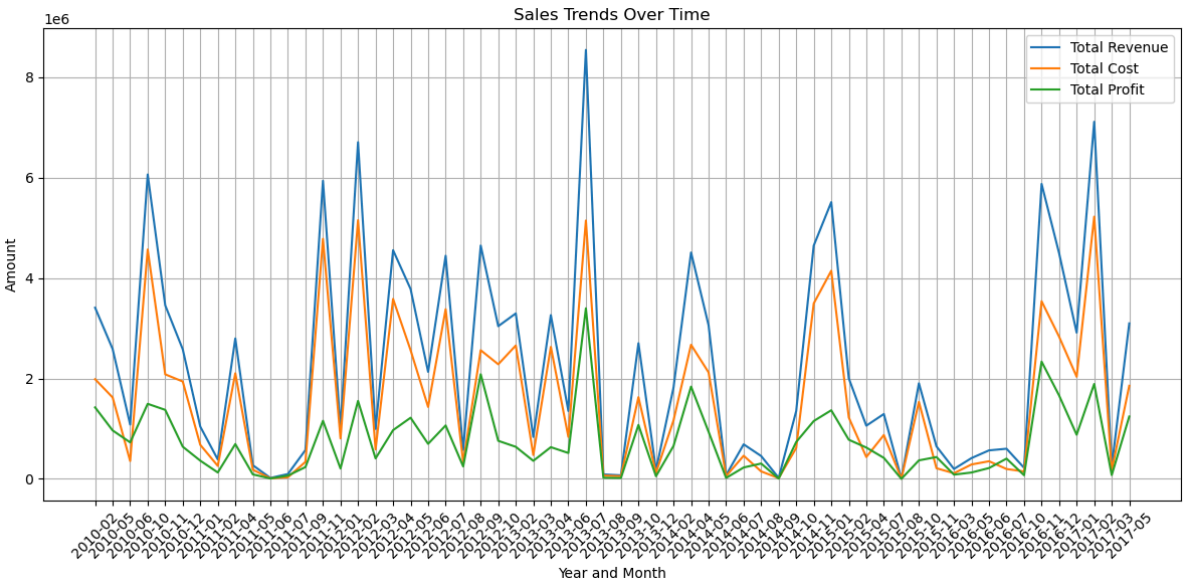
In [42]:
```
#calculate the total profit
total_profit=data["Total Profit"].sum()
print("Total Profit:",total_profit)
```

```
Total Profit: 44168198.39999999
```

In [43]:
```
data.groupby(['Region','Sales Channel'])['Total Profit'].sum()
```

Out[43]:
```
Region                            Sales Channel
Asia                              Offline          3584286.33
                                  Online           2529559.54
Australia and Oceania             Offline          1886283.82
                                  Online           2835876.21
Central America and the Caribbean Offline          2475814.99
                                  Online            371092.86
Europe                            Offline          5574539.91
                                  Online           5508398.72
Middle East and North Africa      Offline          2169081.08
                                  Online           3592110.78
North America                     Offline          1457942.76
Sub-Saharan Africa                Offline          7772777.78
                                  Online           4410433.62
Name: Total Profit, dtype: float64
```

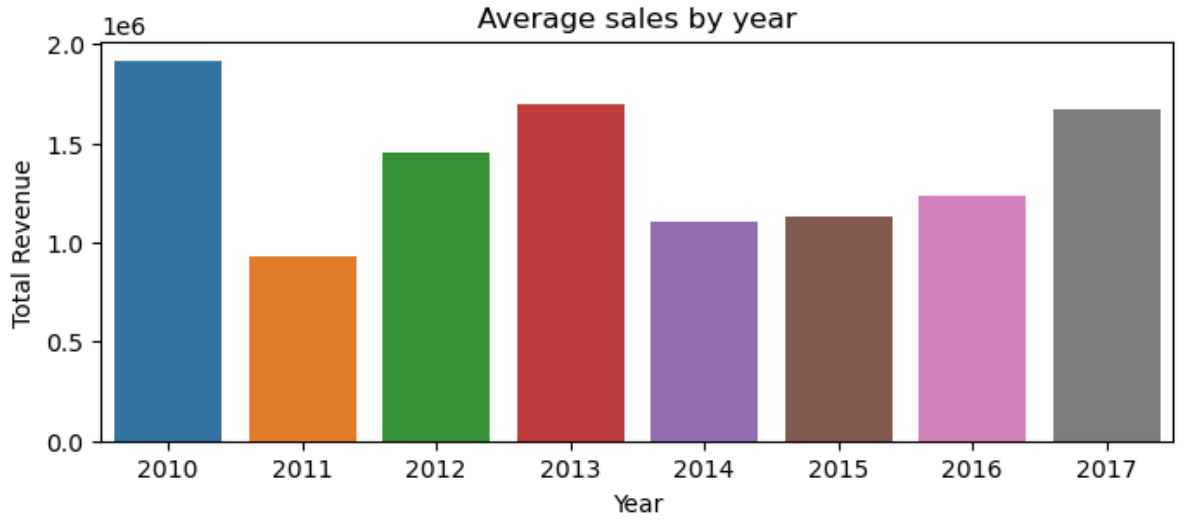
In [61]:
```
# Convert 'Order Date' to datetime format
data['Order Date'] = pd.to_datetime(data['Order Date'])

# Create a new column for Year and Month
data['YearMonth'] = data['Order Date'].dt.to_period('M')

# Aggregate data by Year and Month
YearMonth_Sales =data.groupby('YearMonth').sum(numeric_only=True)[['Total Revenue',

# Plot sales trends over time
plt.figure(figsize=(12, 6))
plt.plot(YearMonth_Sales.index.astype(str), YearMonth_Sales['Total Revenue'], label
plt.plot(YearMonth_Sales.index.astype(str), YearMonth_Sales['Total Cost'], label='T
plt.plot(YearMonth_Sales.index.astype(str), YearMonth_Sales['Total Profit'], label=
plt.xlabel('Year and Month')
plt.ylabel('Amount')
plt.title('Sales Trends Over Time')
plt.legend()
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



The plot above shows the trends for Total Revenue, Total Cost, and Total Profit over time. You can see how these metrics change month by month

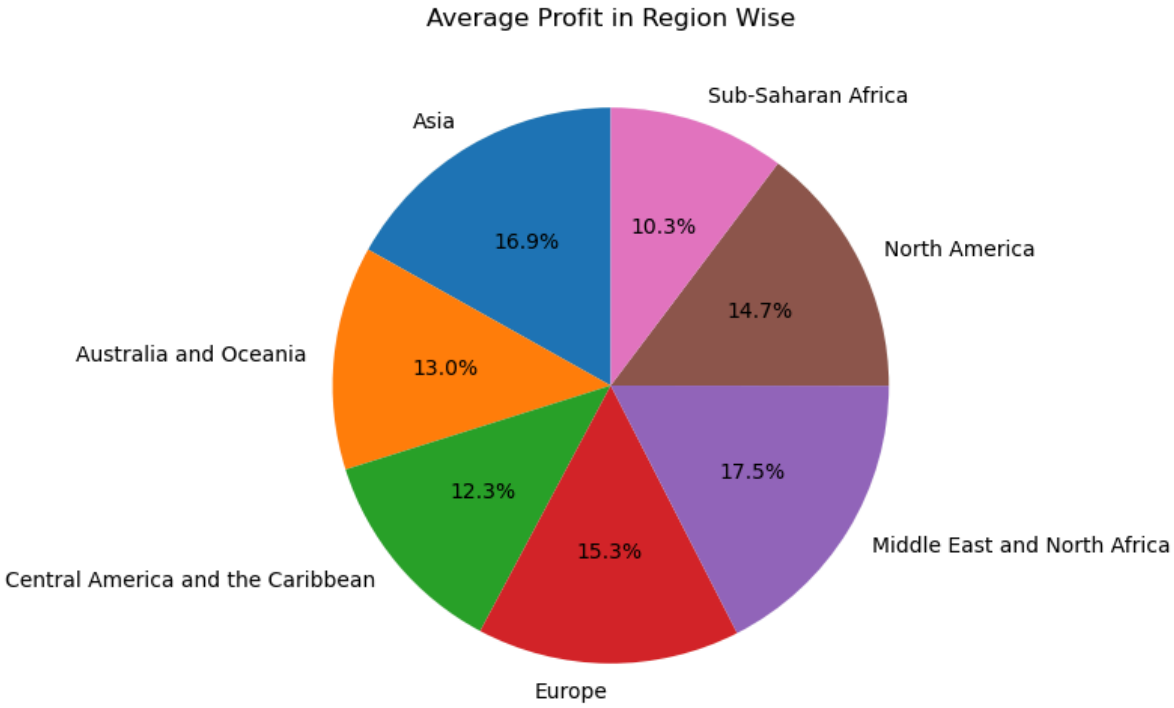
In [44]:
```
#year wise sales
year_sales=data.groupby('Year')['Total Revenue'].mean()
plt.figure(figsize=(8,3))
sns.barplot(x=year_sales.index,y=year_sales.values,)
plt.title('Average sales by year')
plt.xlabel('Year')
plt.ylabel('Total Revenue')
```

Out[44]:
```
Text(0, 0.5, 'Total Revenue')
```



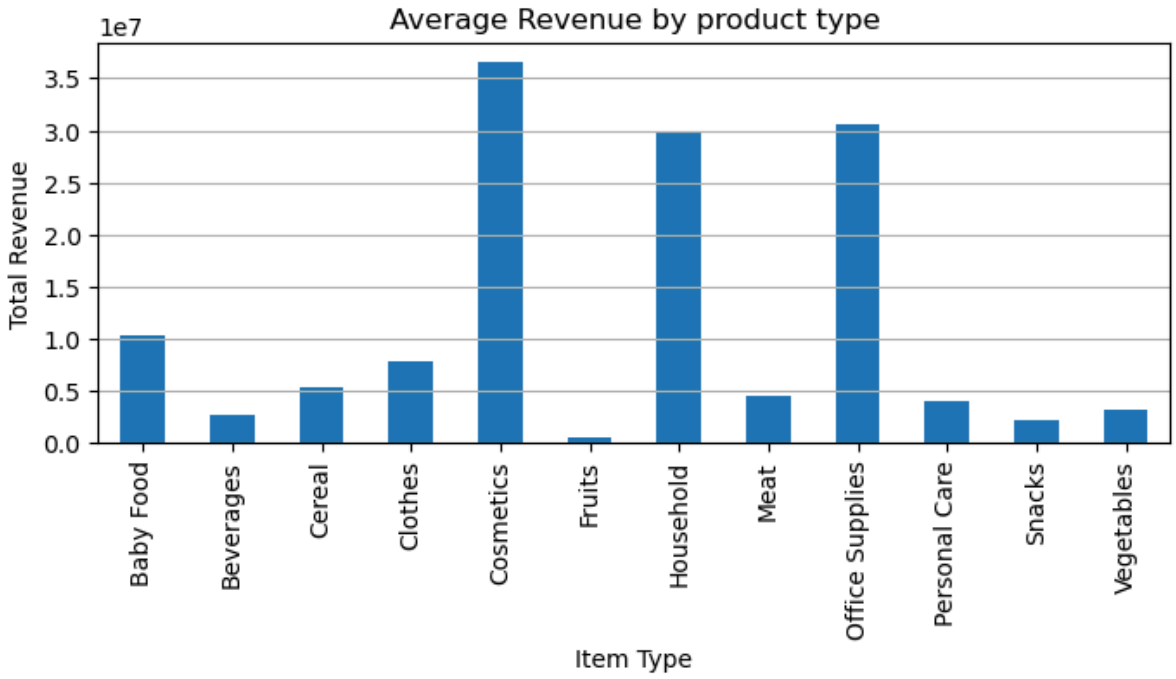
In [45]:
```
# pie chart of total profit in region wise
plt.figure(figsize=(6,6))
region_TotalRevenue=data.groupby('Region')['Total Profit'].mean()
plt.pie(region_TotalRevenue,startangle=90,labels=region_TotalRevenue.index,autopct=
plt.title('Average Profit in Region Wise')
```

Out[45]:
```
Text(0.5, 1.0, 'Average Profit in Region Wise')
```



In [46]:
```
#group total revenue by item type
TotalRevenue_ItemType=data.groupby('Item Type')['Total Revenue'].sum()
```

In [47]:
```
#bar chat for total revenue by item type
plt.figure(figsize=(8,3))
TotalRevenue_ItemType.plot(kind='bar')
plt.title('Average Revenue by product type')
plt.xlabel('Item Type')
plt.ylabel('Total Revenue')
plt.grid(axis='y')
```

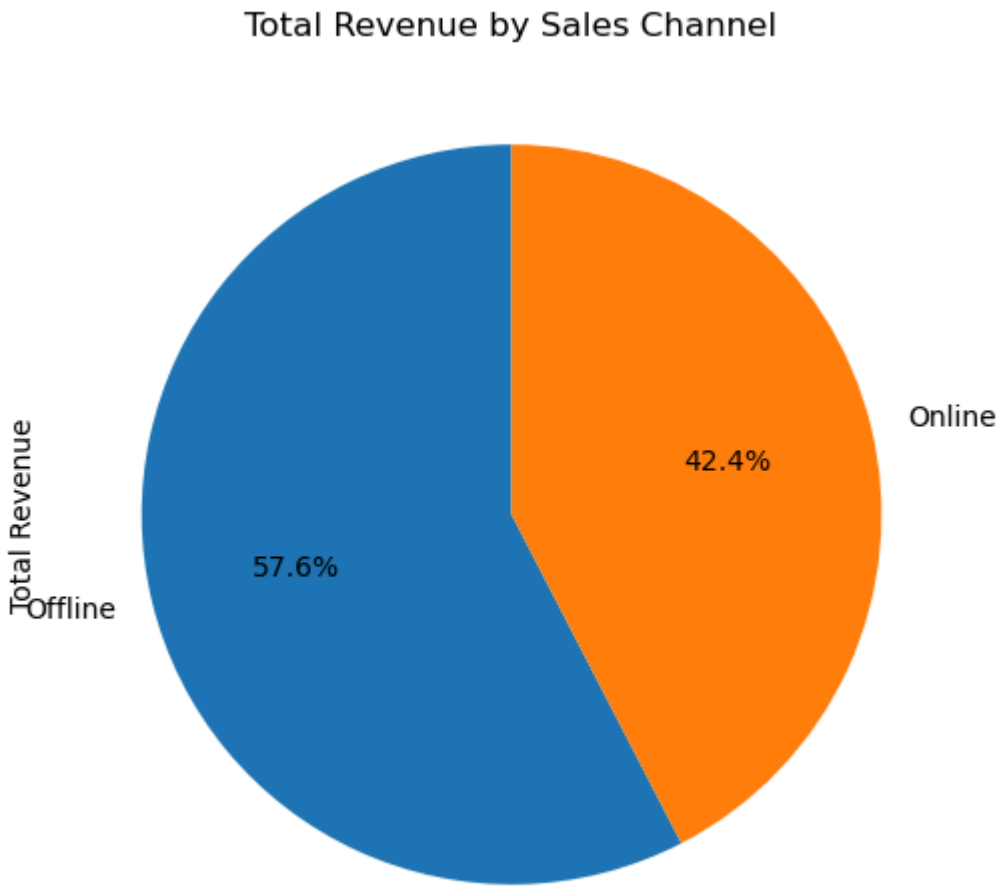


In [48]:
```
#group total revenue by sales channel
TotalRevenue_SalesChannel=data.groupby('Sales Channel')['Total Revenue'].mean()
```

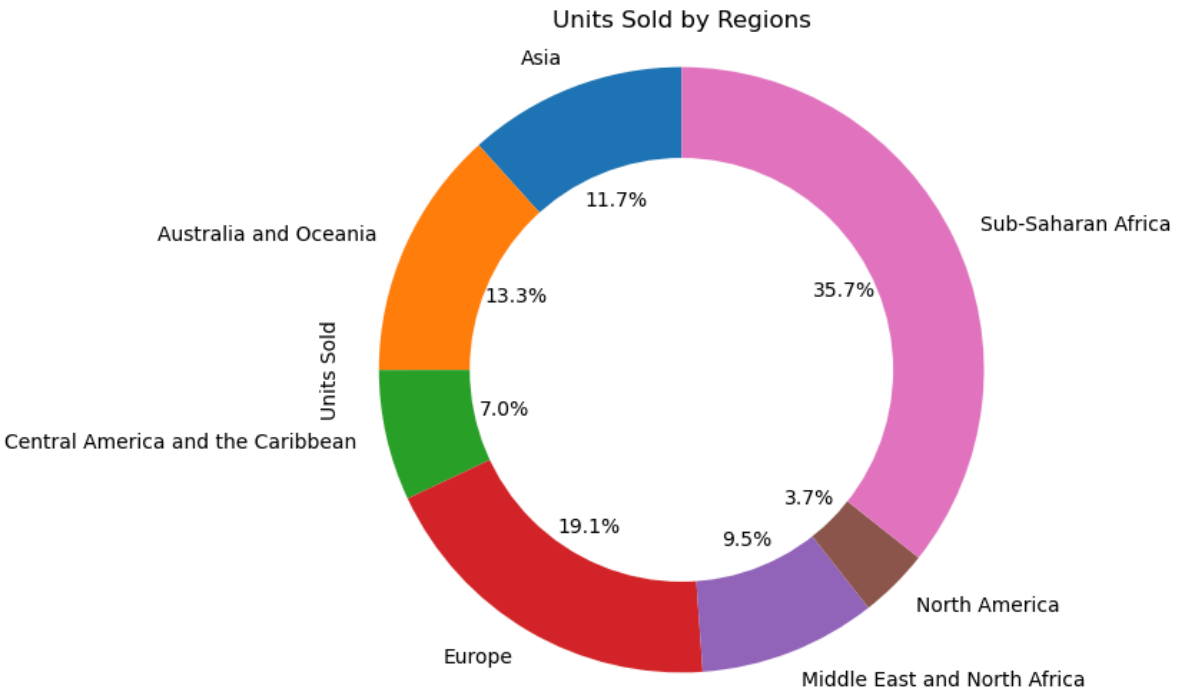
In [49]:
```
#bar chat for total revenue by item type
plt.figure(figsize=(6,6))
```

```
plt.tight_layout()
TotalRevenue_SalesChannel.plot(kind='pie',autopct='%1.1f%%',startangle=90)
plt.title('Total Revenue by Sales Channel')
```

Out[49]:
```
Text(0.5, 1.0, 'Total Revenue by Sales Channel')
```

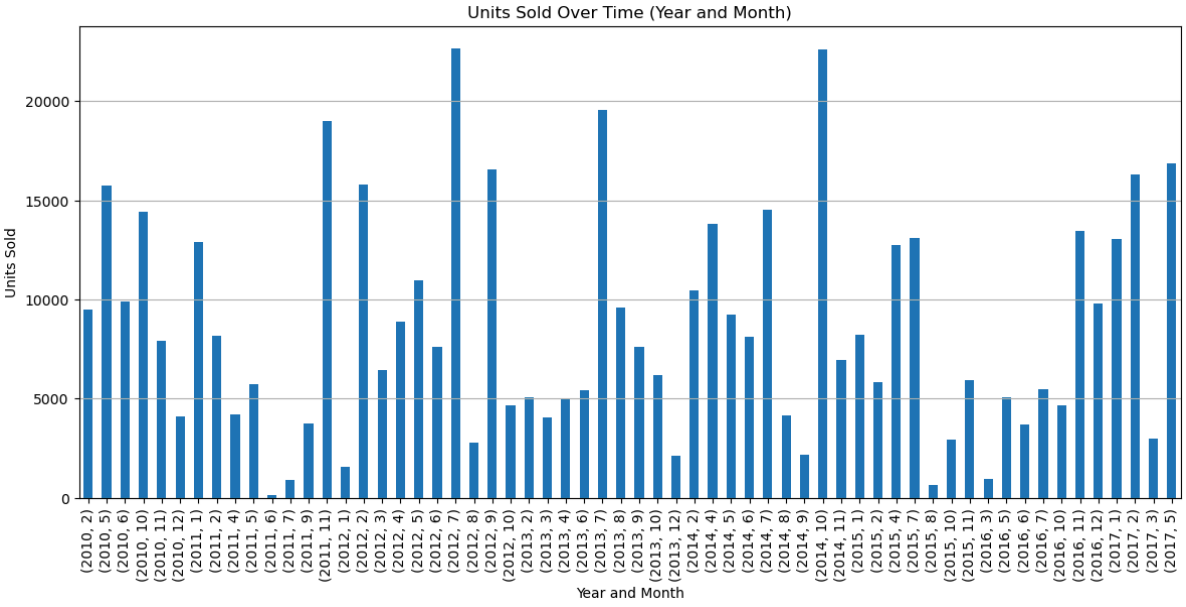


In [52]:
```
import matplotlib.pyplot as plt

# Aggregate data by region for Units Sold
Region_UnitSold = data.groupby('Region')['Units Sold'].sum()

# Plot pie chart for Units Sold by region
plt.figure(figsize=(6, 6))
Region_UnitSold.plot(kind='pie', labels=Region_UnitSold.index, autopct='%1.1f%%', s
cntr_circle = plt.Circle((0, 0), 0.70, fc='white')
fig = plt.gcf()
fig.gca().add_artist(cntr_circle)
plt.title('Units Sold by Regions')
plt.axis('equal')
plt.show()
```

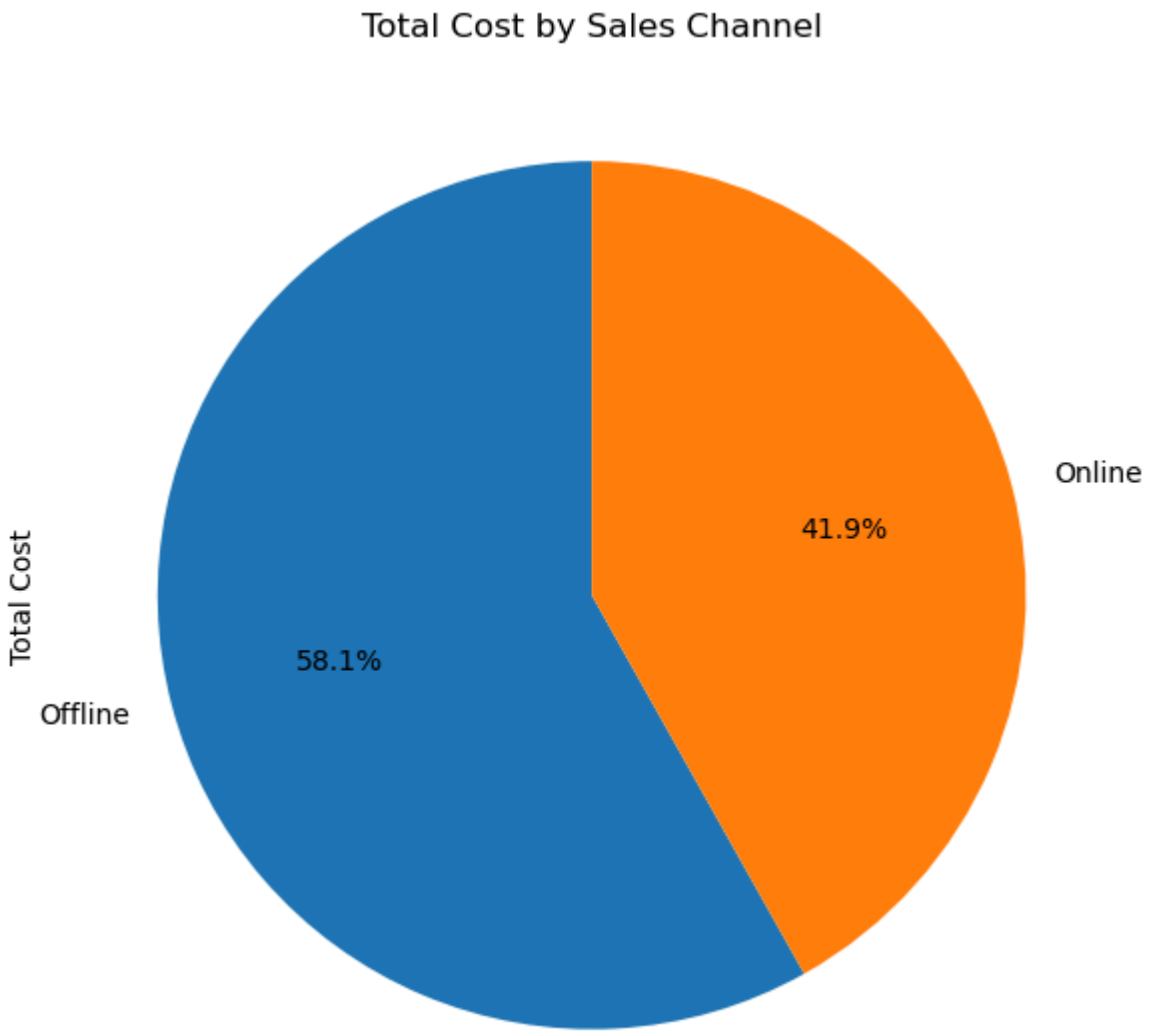


In [53]:
```
#group units sold by year and month
YearMonth_UnitsSold=data.groupby(['Year','Month'])['Units Sold'].sum()
```

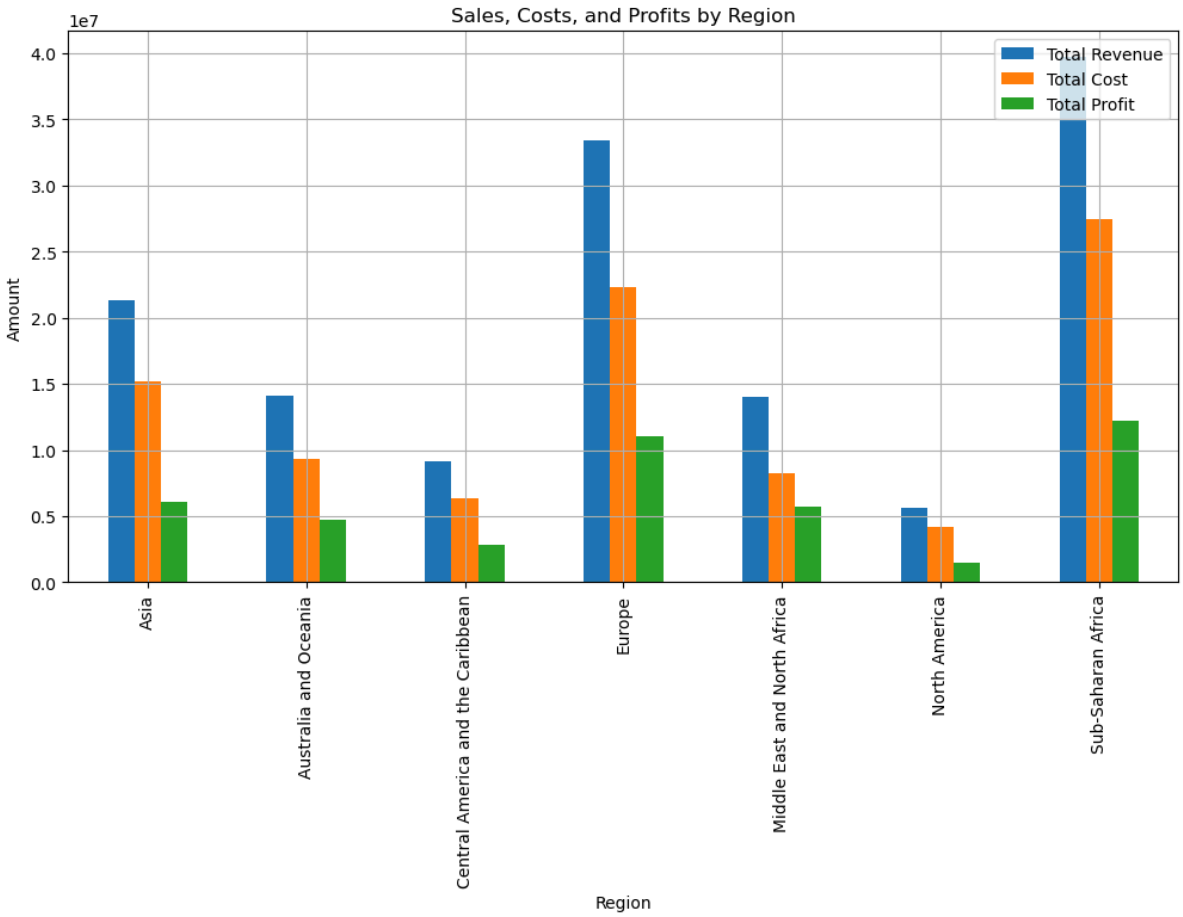
In [55]:
```
# Plot bar chart for Units Sold by Year and Month
plt.figure(figsize=(12, 6))
YearMonth_UnitsSold.plot(kind='bar')
plt.xlabel('Year and Month')
plt.ylabel('Units Sold')
plt.tight_layout()
plt.grid(axis='y')
plt.title('Units Sold Over Time (Year and Month)')
plt.show()
```



In [56]:
```
TotalCost_SalesChannel=data.groupby('Sales Channel')['Total Cost'].sum()
plt.figure(figsize=(6, 6))
TotalCost_SalesChannel.plot(kind='pie',autopct='%1.1f%%',startangle=90)
plt.title('Total Cost by Sales Channel')
plt.tight_layout()
```



In [57]:
```
# Aggregate data by region
regional_analysis =data.groupby('Region').sum(numeric_only=True)[['Total Revenue',

# Plot regional analysis
regional_analysis.plot(kind='bar', figsize=(12, 6))
plt.xlabel('Region')
plt.ylabel('Amount')
plt.title('Sales, Costs, and Profits by Region')
plt.legend(loc='upper right')
plt.grid(True)
plt.show()

regional_analysis
```



Out[57]:

Region	Total Revenue	Total Cost	Total Profit
Asia	21347091.02	15233245.15	6113845.87
Australia and Oceania	14094265.13	9372105.10	4722160.03
Central America and the Caribbean	9170385.49	6323477.64	2846907.85
Europe	33368932.11	22285993.48	11082938.63
Middle East and North Africa	14052706.58	8291514.72	5761191.86
North America	5643356.55	4185413.79	1457942.76
Sub-Saharan Africa	39672031.43	27488820.03	12183211.40

Conclusion

The analysis of the Amazon sales data reveals valuable insights into sales trends, regional performance, product popularity, and the effectiveness of sales channels and order prioritization.

These insights can inform strategic decisions to enhance sales performance, optimize inventory, improve customer satisfaction, and increase profitability.

Regular analysis and visualization of sales data are crucial for maintaining a competitive edge and making informed business decisions.

In [ ]: