Amazon Sales Data Analysis using Python by Prachi Khartade

Unified Mentor Internship

```
#Objective: The purpose of this analysis is to gain insights into Amazon's sales performance, identify trends, and draw actionable conclu
#Tools Used: Python libraries like Pandas, NumPy, Matplotlib, and Seaborn.
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

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

```
df=pd.read_csv('Amazon Sales data.csv')
```

+ Code + Text

```
df
```

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost
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
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
2	Europe	Russia	Office Supplies	Offline	L	5/2/2014	341417157	5/8/2014	1779	651.21	524.96	1158502.59	933903.84
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
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
...	...	...	...	...	...	...	...	...	...	...	...	...	...
95	Sub-Saharan Africa	Mali	Clothes	Online	M	7/26/2011	512878119	9/3/2011	888	109.28	35.84	97040.64	31825.92
96	Asia	Malaysia	Fruits	Offline	L	11/11/2011	810711038	12/28/2011	6267	9.33	6.92	58471.11	43367.64

```
#Display total number of rows and columns

df.shape
print("Total Number of Rows: ",df.shape[0])
print("Total Number of Coulmns: ",df.shape[1])
```

```
Total Number of Rows:  100
Total Number of Coulmns:  14
```

```
# Display first rows.   ---- head()

df.head(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	P
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	9514
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	2484
2	Europe	Russia	Office Supplies	Offline	L	5/2/2014	341417157	5/8/2014	1779	651.21	524.96	1158502.59	933903.84	2245
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	195
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	6390
5	Australia and	Solomon Islands	Baby Food	Online	C	2/4/2015	547995746	2/21/2015	2974	255.28	159.42	759202.72	474115.08	2850

```
#Display last 10 rows  ---- tail()

df.tail(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
90	Sub-Saharan Africa	Sierra Leone	Office Supplies	Offline	H	12/6/2016	621386563	12/14/2016	948	651.21	524.96	617347.08	497662.08
91	Australia and Oceania	Australia	Beverages	Offline	H	7/7/2014	240470397	7/11/2014	9389	47.45	31.79	445508.05	298476.31
92	Middle East and North Africa	Azerbaijan	Office Supplies	Online	M	6/13/2012	423331391	7/24/2012	2021	651.21	524.96	1316095.41	1060944.16
93	Europe	Romania	Cosmetics	Online	H	11/26/2010	660643374	12/25/2010	7910	437.20	263.33	3458252.00	2082940.30
94	Central America and the Caribbean	Nicaragua	Beverages	Offline	C	2/8/2011	963392674	3/21/2011	8156	47.45	31.79	387002.20	259279.24
95	Sub-Saharan	Mali	Clothes	Online	M	7/26/2011	512878119	9/3/2011	888	109.28	35.84	97040.64	31825.92

```
#Display random 10 rows. ---- sample()

df.sample(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
47	Europe	Switzerland	Personal Care	Online	M	12/23/2010	617667090	1/31/2011	273	81.73	56.67	22312.29	15470.91
23	Australia and Oceania	New Zealand	Fruits	Online	H	9/8/2014	142278373	10/4/2014	2187	9.33	6.92	20404.71	15134.04
33	Asia	Myanmar	Household	Offline	H	1/16/2015	177713572	3/1/2015	8250	668.27	502.54	5513227.50	4145955.00
68	Europe	Lithuania	Office Supplies	Offline	H	10/24/2010	166460740	11/17/2010	8287	651.21	524.96	5396577.27	4350343.52
65	Sub-Saharan Africa	Rwanda	Cosmetics	Offline	H	10/11/2013	699358165	11/25/2013	4477	437.20	263.33	1957344.40	1178928.41
70	Asia	Turkmenistan	Office Supplies	Online	M	4/23/2013	462405812	5/20/2013	5010	651.21	524.96	3262562.10	2630049.60
20	Europe	Norway	Baby Food	Online	L	5/14/2014	819028031	6/28/2014	7450	255.28	159.42	1901836.00	1187679.00
	Middle East												

Data cleaning

```
#Identify whether any columns has null values

df.isnull().sum()
```

```
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
```

```
#Display all column names

df.columns
```

```
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'],
      dtype='object')
```

```
#Display datatypes of each column

df.dtypes
```

```
Region             object
Country            object
Item Type          object
Sales Channel      object
Order Priority     object
Order Date         object
Order ID            int64
Ship Date          object
Units Sold          int64
Unit Price        float64
Unit Cost         float64
Total Revenue     float64
Total Cost        float64
Total Profit      float64
dtype: object
```

There are no missing/null values and data types are also correct hence data is already clean.

```
df['Order Date']
```

```
0      5/28/2010
1      8/22/2012
2       5/2/2014
3      6/20/2014
4       2/1/2013
         ...
95     7/26/2011
96    11/11/2011
97     6/1/2016
98     7/30/2015
99     2/10/2012
Name: Order Date, Length: 100, dtype: object
```

```
#Convert order date to datetime

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

```
df['Order Date']
```

```
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]
```

```
#Extract Year and Month from Order Date

df['Year']=df['Order Date'].dt.year
df['Month']=df['Order Date'].dt.month
df.head(5)
```

	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
	Sub-	Sao Tome				2014-								

Data Analysis

```
#Calculate the unique Number of regions

regions=df['Region'].nunique()
print("Number of regions: ",regions)
```

```
Number of regions:  7
```

```
#Calculate the Number of countries

country=df['Country'].nunique()
print("Number of countries: ",country)
```

```
Number of countries:  76
```

```
#Calculate the items

item_type=df['Item Type'].nunique()
print("Number of Item Types: ",item_type)
```

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

```
#Calculate the total unit sold

unit_sold=df['Units Sold'].sum()
print("Total Unit Sold: ",unit_sold)
```

```
Total Unit Sold:  512871
```

```
#Calculate the Total unit cost

unit_cost=df['Unit Cost'].sum()
print("Total Unit Cost: ",unit_cost)
```

```
Total Unit Cost:  19104.8
```

```
#Calculate the total revenue

total_revenue=df['Total Revenue'].sum()
print("Total Revenue: ",total_revenue)
```

```
Total Revenue:  137348768.31
```

```
#Calculte the total cost

total_cost=df['Total Cost'].sum()
print("Total Cost: ",total_cost)
```

```
Total Cost:  93180569.91000001
```

```
#Calculate the total Profit

total_profit=df['Total Profit'].sum()
print("Total Profit: ",total_profit)
```

```
Total Profit:  44168198.39999999
```

```
df.groupby(['Region','Sales Channel'])['Total Profit'].sum()
```

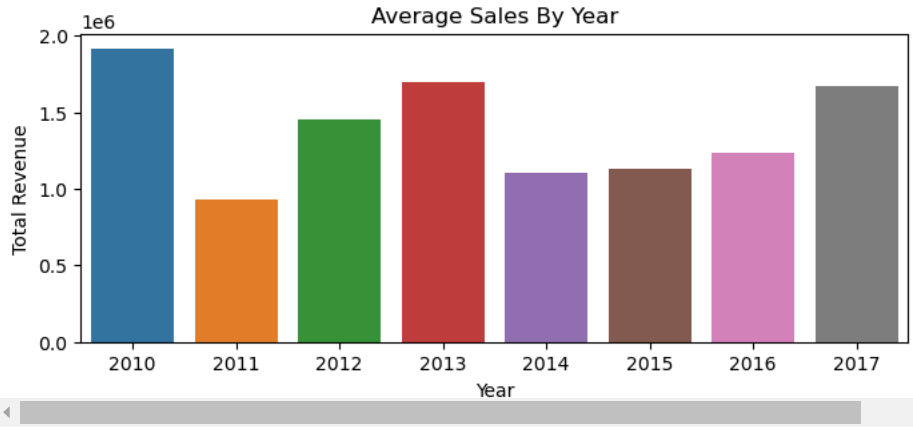
```
Region                           Sales Channel
Asia                             Offline          3584286.33
                                 Online           2529559.54
Australia and Oceania            Offline          1886283.82
                                 Online           2835876.21
Central 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
```

Hence Europe leads in total profit, driven by strong online sales, while other regions like Asia and Sub-Saharan Africa excel more in offline sales

Data Visualisations

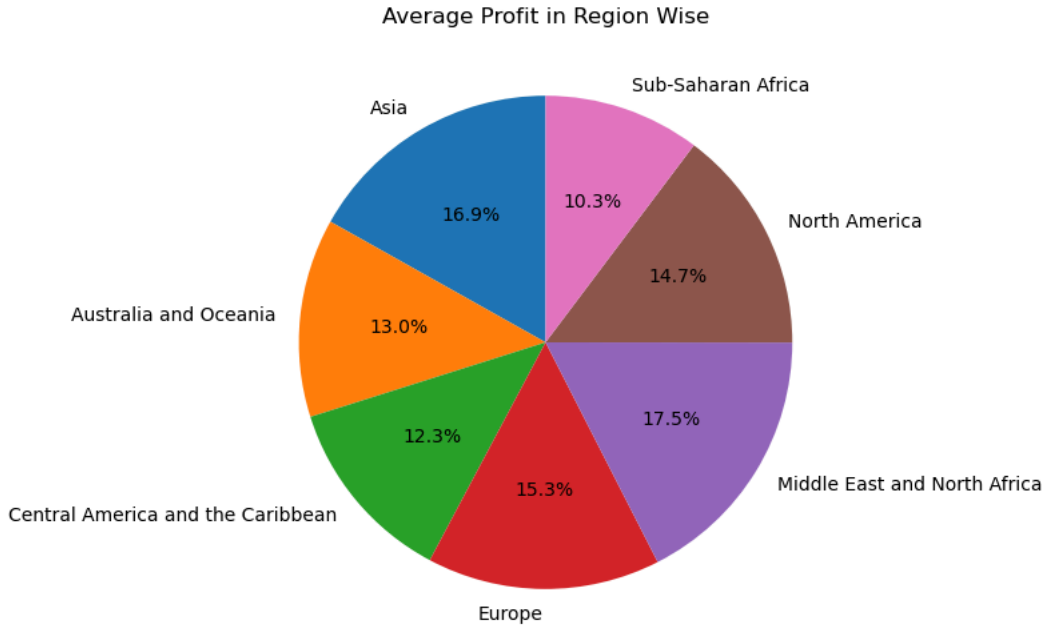
```
#year wise sales

year_sales=df.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")
```

```
Text(0, 0.5, 'Total Revenue')
```

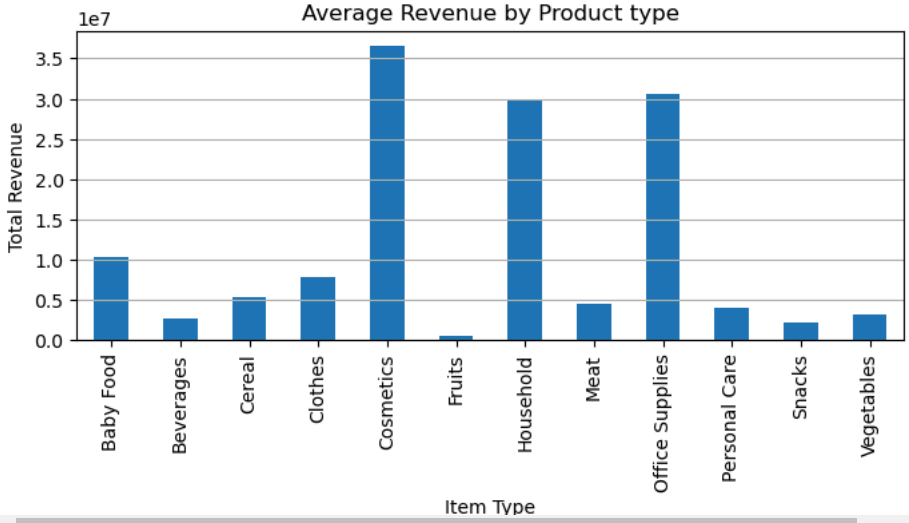



```
#Pie chart of Total Prfit in region wise

plt.figure(figsize=(6,6))
region_TotalRevenue=df.groupby('Region')['Total Profit'].mean()
plt.pie(region_TotalRevenue,startangle=90,labels=region_TotalRevenue.index,autopct='%1.1f%%')
plt.title("Average Profit in Region Wise")
```

```
Text(0.5, 1.0, 'Average Profit in Region Wise')
```

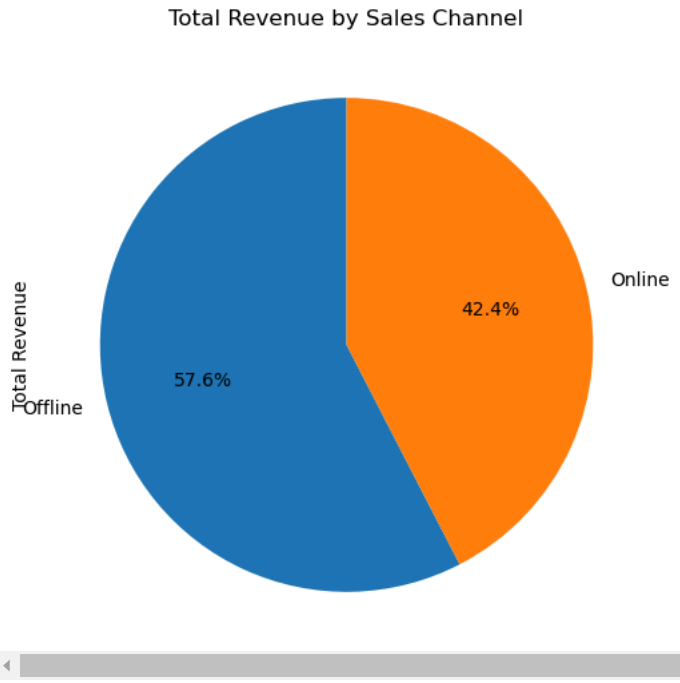



```
#Bar chart for Total revenue by item type

#group Total Revenue By Item Type
TotalRevenue_ItemType=df.groupby('Item Type')['Total Revenue'].sum()

plt.figure(figsize=(8,3))
TotalRevenue_ItemType.plot(kind='bar')
plt.xlabel("Item Type")
plt.ylabel("Total Revenue")
plt.title("Average Revenue by Product type")
plt.grid(axis='y')
```



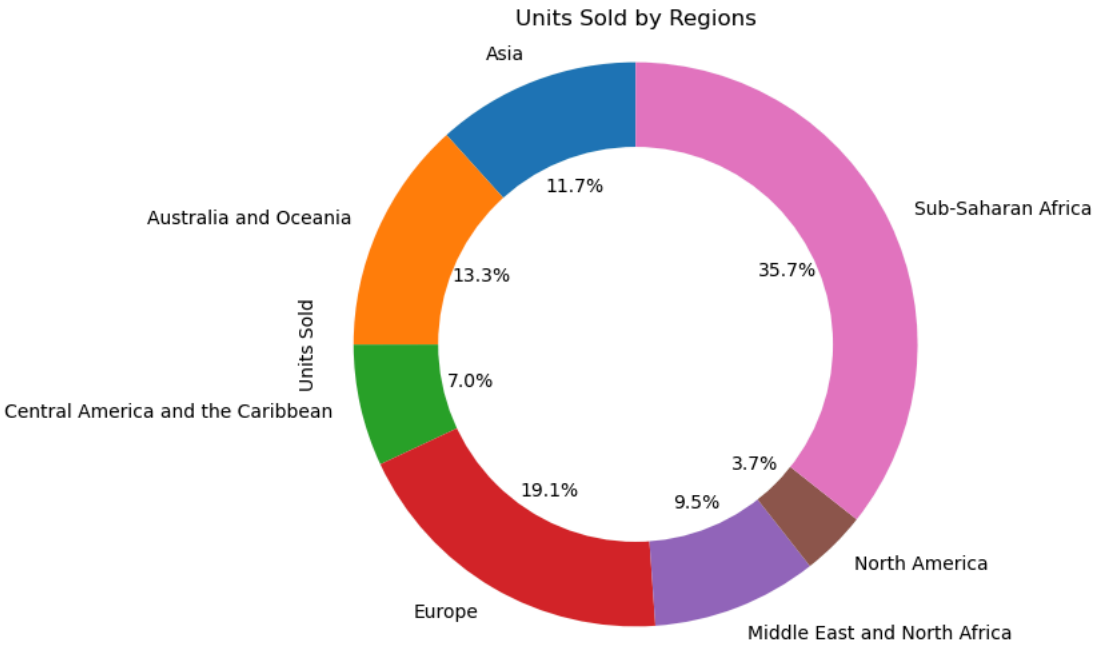

```
#Bar Chart for total revenue by item type

#group by total revenue by sales channel
TotalRevenue_SalesChannel=df.groupby('Sales Channel')['Total Revenue'].mean()

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")
```

```
Text(0.5, 1.0, 'Total Revenue by Sales Channel')
```

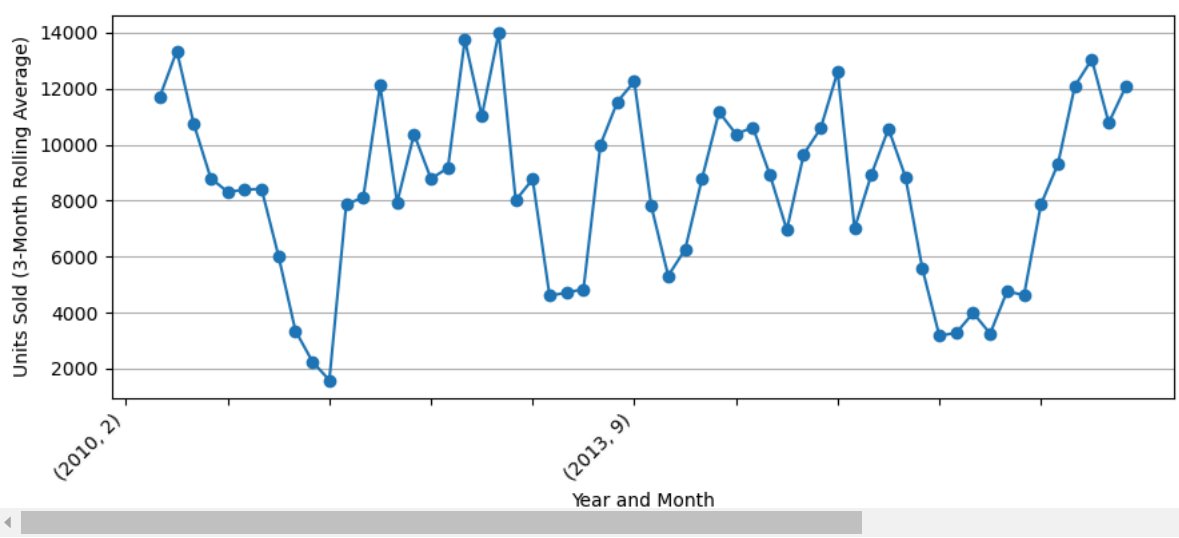



```
#Create a pie chart for a donut chart

Region_UnitSold=df.groupby('Region')['Units Sold'].sum()
plt.figure(figsize=(6,6))
Region_UnitSold.plot(kind="pie",labels=Region_UnitSold.index,autopct="%1.1f%%",startangle=90)

#Draw a circle at the centre of a pie chart
cntr_circle=plt.Circle((0,0),(0.70),fc="white")
fig=plt.gcf()
fig.gca().add_artist(cntr_circle)

#Equal aspect ratio ensures that pie is drwan as a circle
plt.title("Units Sold by Regions")
plt.axis("equal")
```

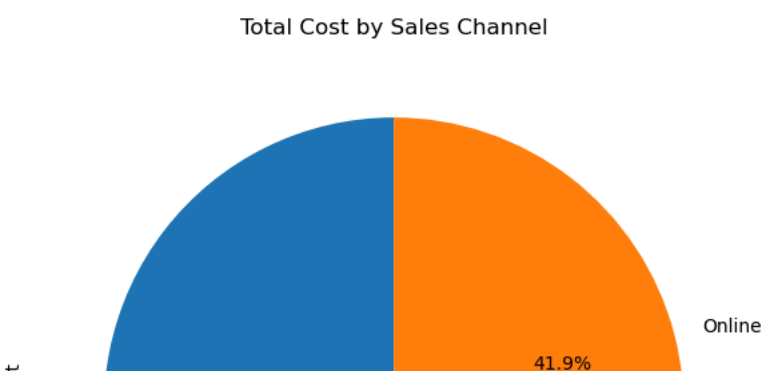
```
(-1.0999999530116766,
 1.0999990132545814,
 -1.0999995737000883,
 1.0999999797000042)
```




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

# Applying a rolling average with a window of 3 months
YearMonth_UnitsSold_Rolling = YearMonth_UnitsSold.rolling(window=3).mean()

# Plotting
plt.figure(figsize=(9, 4))
YearMonth_UnitsSold_Rolling.plot(kind='line', marker='o')
plt.xlabel("Year and Month")
plt.ylabel("Units Sold (3-Month Rolling Average)")
plt.xticks(ticks=range(0, len(YearMonth_UnitsSold), 6), rotation=45, ha='right')
plt.tight_layout()
plt.grid(axis="y")
plt.show()
```




```
#gropy total cost by sales channel

TotalCost_SalesChannel=df.groupby("Sales Channel")['Total Cost'].sum()

#pie chart for total cost by sales channel
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()
```


Total Cost by Sales Channel
Online
41.9%