Importing the Libraries

In [48]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Importing Dataset

How to import CSV File ?

In [49]:
```
sd = pd.read_csv("sales_data.csv")
```

Read the DataFrame

Q:- How to view the first 3 Rows ?

In [50]:
```
sd.head(3) # head() method shows the firsh Rows of the dataset
```

Out[50]:

	Row ID+O6G3A1:R6	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segmer
0	4918	CA-2019-160304	1/1/2019	1/7/2019	Standard Class	BM-11575	Brendan Murry	Corporat
1	4919	CA-2019-160304	1/2/2019	1/7/2019	Standard Class	BM-11575	Brendan Murry	Corporat
2	4920	CA-2019-160304	1/2/2019	1/7/2019	Standard Class	BM-11575	Brendan Murry	Corporat

3 rows × 23 columns

Q:- How to view the last 3 Rows ?

In [51]:
```
sd.tail(3) # tail() method shows the last Rows of the dataset
```

Out[51]:

	Row ID+O6G3A1:R6	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	S
5898	5092	CA-2020-156720	12/30/2020	1/3/2021	Standard Class	JM-15580	Jill Matthias	Cc
5899	909	CA-2020-143259	12/30/2020	1/3/2021	Standard Class	PO-18865	Patrick O'Donnell	Cc
5900	5093	CA-2020-151450	12/31/2020	1/4/2021	Standard Class	JM-15580	Jill Matthias	Cc

3 rows × 23 columns

Q:- How to view total number of Rows&Columns ?

In [52]:
```
sd.shape
```

Out[52]:
```
(5901, 23)
```

In [53]:
```
Q:- How to Show the Dtypes of the dataset?
```

```
Object `dataset` not found.
```

In [54]:
```
sd.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5901 entries, 0 to 5900
Data columns (total 23 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Row ID+O6G3A1:R6  5901 non-null   int64  
 1   Order ID          5901 non-null   object 
 2   Order Date        5901 non-null   object 
 3   Ship Date         5901 non-null   object 
 4   Ship Mode         5901 non-null   object 
 5   Customer ID       5901 non-null   object 
 6   Customer Name     5901 non-null   object 
 7   Segment           5901 non-null   object 
 8   Country           5901 non-null   object 
 9   City              5901 non-null   object 
 10  State             5901 non-null   object 
 11  Region            5901 non-null   object 
 12  Product ID        5901 non-null   object 
 13  Category          5901 non-null   object 
 14  Sub-Category      5901 non-null   object 
 15  Product Name      5901 non-null   object 
 16  Sales             5901 non-null   float64
 17  Quantity          5901 non-null   int64  
 18  Profit            5901 non-null   float64
 19  Returns           287 non-null    float64
 20  Payment Mode      5901 non-null   object 
 21  ind1              0 non-null      float64
 22  ind2              0 non-null      float64
dtypes: float64(5), int64(2), object(16)
memory usage: 1.0+ MB
```

Data Cleaning

Q:- How to find out the Null Values from given dataSet?

In [55]:
```
# isnull() object finds all NA/null values & Sum() method calculate the sum of v
# Checking for Null Values in Dataset:
sd.isnull().sum()
```

Out[55]:
```
Row ID+O6G3A1:R6       0
Order ID               0
Order Date             0
Ship Date              0
Ship Mode              0
Customer ID            0
Customer Name          0
Segment                0
Country                0
City                   0
State                  0
Region                 0
Product ID             0
Category               0
Sub-Category           0
Product Name           0
Sales                  0
Quantity               0
Profit                 0
Returns             5614
Payment Mode           0
ind1                5901
ind2                5901
dtype: int64
```

In [56]:
```
sd.drop(columns=['ind1','ind2'],axis=1,inplace=True)
```

In [57]:
```
sd.isnull().sum()
```

Out[57]:
```
Row ID+O6G3A1:R6       0
Order ID               0
Order Date             0
Ship Date              0
Ship Mode              0
Customer ID            0
Customer Name          0
Segment                0
Country                0
City                   0
State                  0
Region                 0
Product ID             0
Category               0
Sub-Category           0
Product Name           0
Sales                  0
Quantity               0
Profit                 0
Returns             5614
Payment Mode           0
dtype: int64
```

In [58]:
```
sd.replace('#N/A', np.nan, inplace=True)
sd.fillna(10,inplace=True)
sd.isnull().sum()
```

Out[58]:
```
Row ID+O6G3A1:R6    0
Order ID            0
Order Date          0
Ship Date           0
Ship Mode           0
Customer ID         0
Customer Name       0
Segment             0
Country             0
City                0
State               0
Region              0
Product ID          0
Category            0
Sub-Category        0
Product Name        0
Sales               0
Quantity            0
Profit              0
Returns             0
Payment Mode        0
dtype: int64
```

In [59]:
```
sd.shape
```

Out[59]:
```
(5901, 21)
```

In [60]:
```
sd.duplicated().sum()
```

Out[60]:
```
0
```

In [61]:
```
sd.columns
```

Out[61]:
```
Index(['Row ID+O6G3A1:R6', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode',
       'Customer ID', 'Customer Name', 'Segment', 'Country', 'City', 'State',
       'Region', 'Product ID', 'Category', 'Sub-Category', 'Product Name',
       'Sales', 'Quantity', 'Profit', 'Returns', 'Payment Mode'],
      dtype='object')
```

In [62]:
```
sd.rename(columns={'Product Name': 'Product_name'},inplace=True)
```

In [63]:
```
sd.columns
```

Out[63]:
```
Index(['Row ID+O6G3A1:R6', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode',
       'Customer ID', 'Customer Name', 'Segment', 'Country', 'City', 'State',
       'Region', 'Product ID', 'Category', 'Sub-Category', 'Product_name',
       'Sales', 'Quantity', 'Profit', 'Returns', 'Payment Mode'],
      dtype='object')
```

Data Analysis

What are the top-selling Sub-Category AND categories ?

In [64]:
```
sd.groupby(['Category'])['Sales'].sum()
```

Out[64]:
```
Category
Furniture          451508.6452
Office Supplies    643707.6870
Technology         470587.9910
Name: Sales, dtype: float64
```

In [65]:
```
sd.groupby(['Sub-Category'])['Sales'].sum()
```

Out[65]:
```
Sub-Category
Accessories    122301.0860
Appliances      80305.2470
Art             50762.9760
Binders        174978.3900
Bookcases       57577.6862
Chairs         181945.9980
Copiers         59735.7980
Envelopes       16542.4640
Fasteners       15205.2380
Furnishings     92691.2180
Labels          19397.4560
Machines        91987.5610
Paper           99453.6120
Phones         196563.5460
Storage        150341.3180
Supplies        36720.9860
Tables         119293.7430
Name: Sales, dtype: float64
```

In [66]:
```
sd.columns
```

Out[66]:
```
Index(['Row ID+O6G3A1:R6', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode',
       'Customer ID', 'Customer Name', 'Segment', 'Country', 'City', 'State',
       'Region', 'Product ID', 'Category', 'Sub-Category', 'Product_name',
       'Sales', 'Quantity', 'Profit', 'Returns', 'Payment Mode'],
      dtype='object')
```

What is the most profitable product in sub-category?

In [67]:
```
sub_cat_pro = sd.groupby(['Sub-Category'])['Profit'].sum().sort_values(ascending
sub_cat_pro.idxmax
```

Out[67]:
```
<bound method Series.idxmax of Sub-Category
Copiers        42774.5828
Accessories    25336.6455
Phones         22308.9179
Paper          21112.3779
Binders        17885.3759
Storage        13607.0875
Chairs         13406.7032
Appliances     13166.6098
Furnishings     8034.4328
Art             3635.9257
Envelopes       3508.5073
Labels          2937.2212
Fasteners        598.4175
Machines          38.1024
Bookcases       -342.8883
Supplies       -1654.2767
Tables        -11091.6365
Name: Profit, dtype: float64>
```

What is the average sales per customer?

In [68]:
```
Avg_sales = sd.groupby(['Customer Name'])['Sales'].sum().mean()
Avg_sales
```

Out[68]:
```
2025.620081759379
```

What is the top 5 most sold products by quantity?

In [69]:
```
hig_sold = sd.groupby(['Product_name'])['Quantity'].sum().sort_values(ascending=
hig_sold
```

Out[69]:
```
Product_name
Staples                                 124
Easy-staple paper                        89
Staple envelope                          73
Staples in misc. colors                  60
Chromcraft Round Conference Tables       59
Name: Quantity, dtype: int64
```

Who is the most profitable customer?

In [70]:
```
pro_cust = sd.groupby(['Customer Name'])['Profit'].sum().sort_values(ascending=F
pro_cust
```

Out[70]:
```
'Tamara Chand'
```

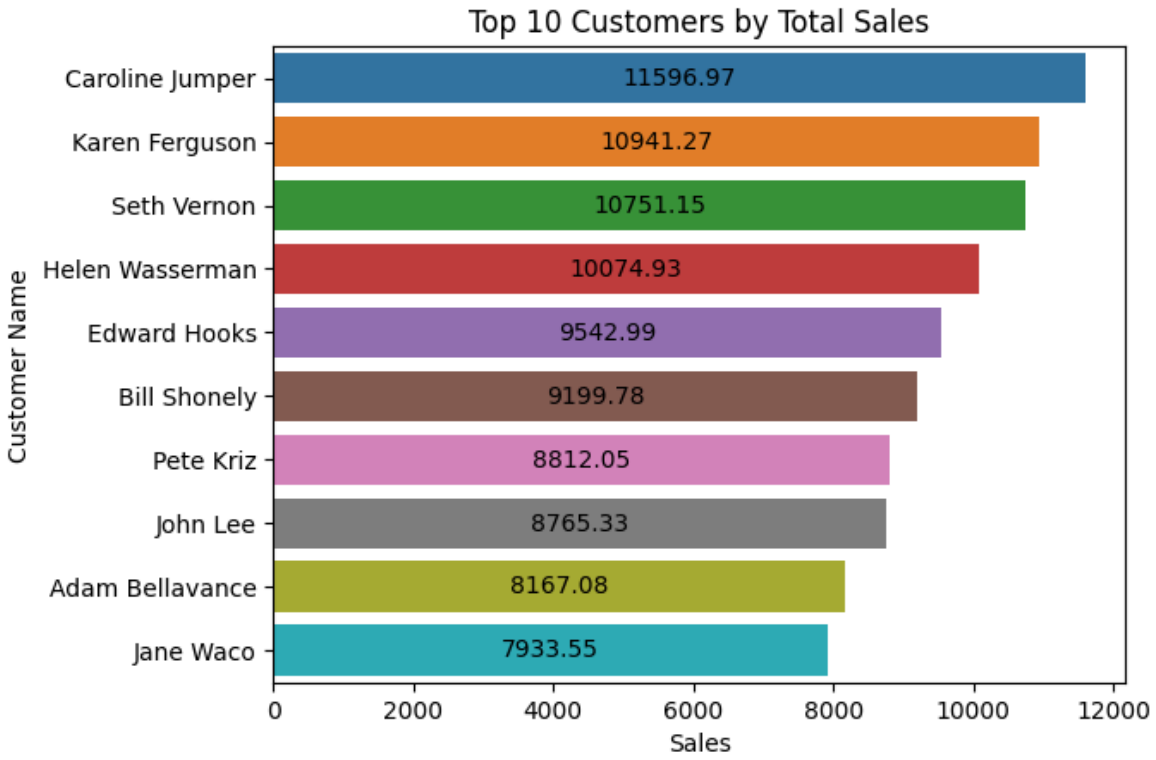
Data Analysis & Visualization

In [71]:
```
sd.columns # To View columns
```

Out[71]:
```
Index(['Row ID+O6G3A1:R6', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode',
       'Customer ID', 'Customer Name', 'Segment', 'Country', 'City', 'State',
       'Region', 'Product ID', 'Category', 'Sub-Category', 'Product_name',
       'Sales', 'Quantity', 'Profit', 'Returns', 'Payment Mode'],
      dtype='object')
```

What are the total sales per customer?

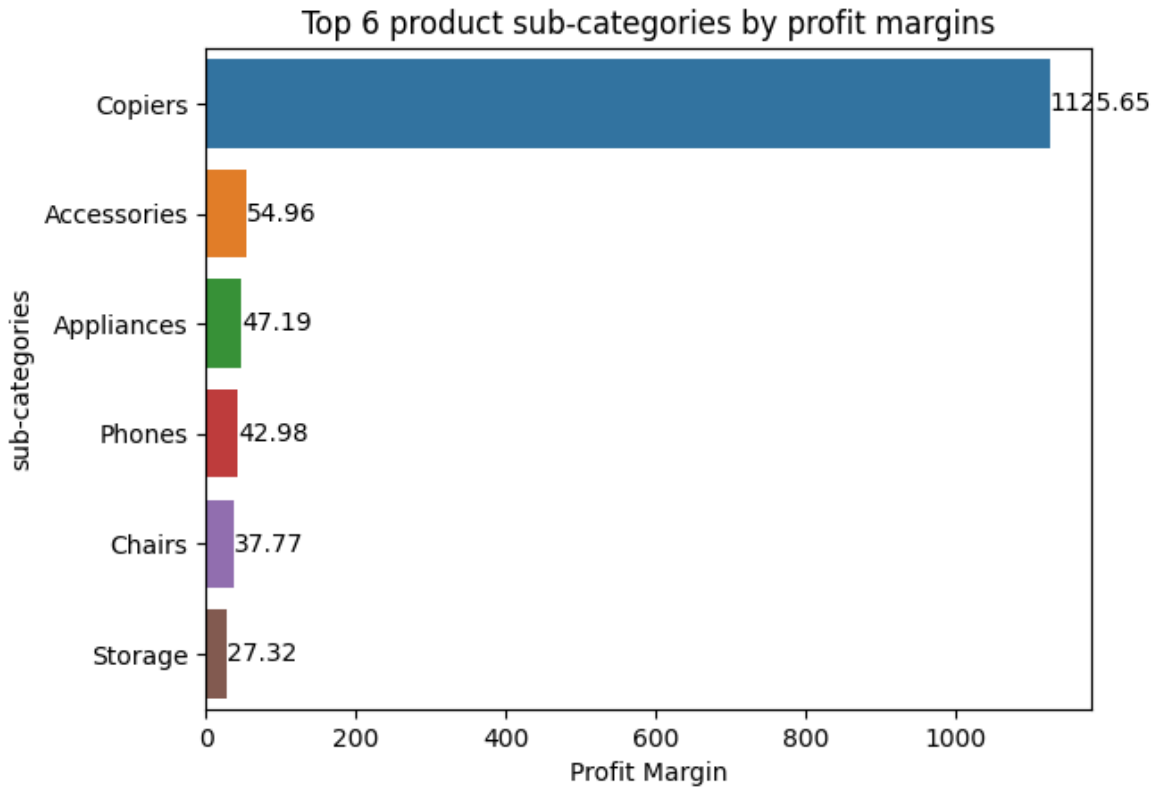
In [72]:
```
cust_sales = sd.groupby(['Customer Name'])['Sales'].sum().sort_values(ascending=

# Visualizing the top 10 customers by total sales
ds=sns.barplot(x=cust_sales.head(10).values, y=cust_sales.head(10).index)
plt.title('Top 10 Customers by Total Sales') # Title for Graph pic
plt.xlabel('Sales')
plt.ylabel('Customer Name')
plt.savefig('Top 10 Customers by Total Sales.jpg')
# Adding value labels on each bar
plt.bar_label(ds.containers[0], fmt='%.2f', label_type='center')

plt.figure(figsize= (1,1))

plt.show()
```




```
<Figure size 100x100 with 0 Axes>
```

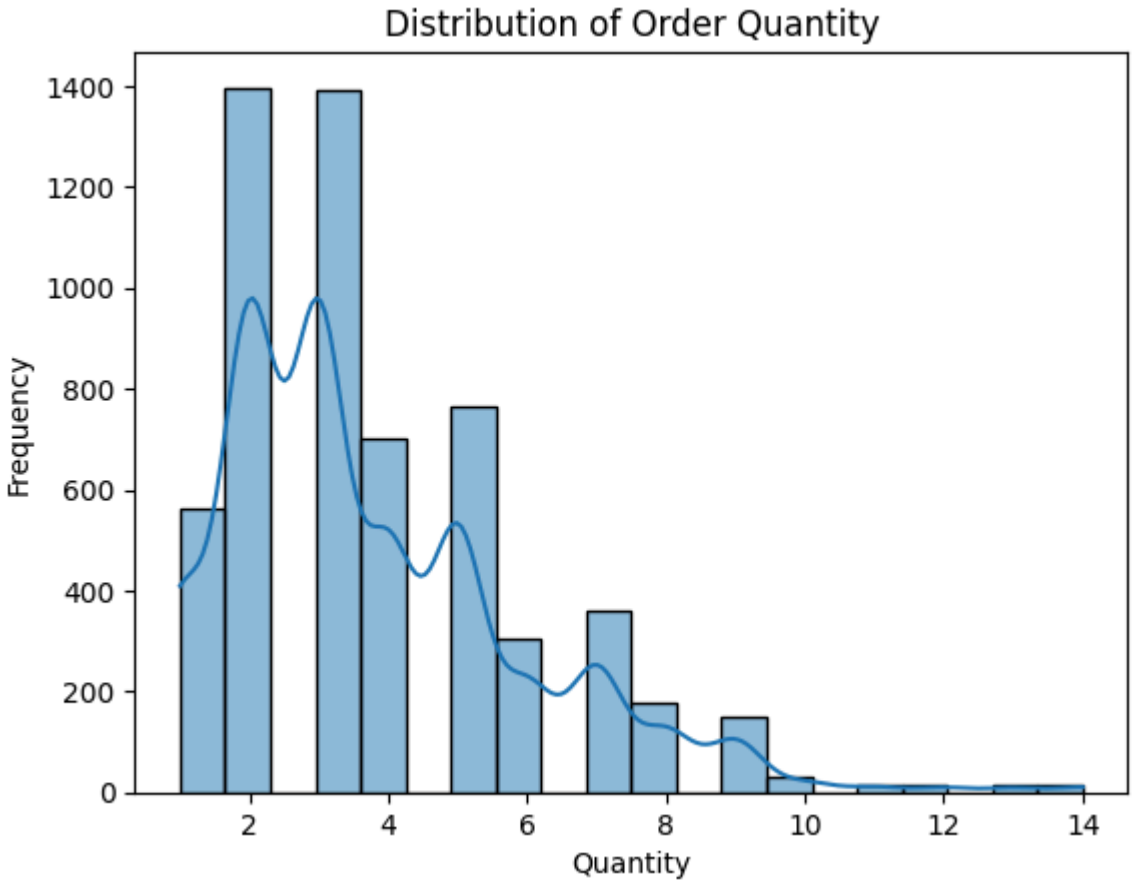
Which product sub-categories have the highest profit margins?

In [73]:
```
Profit_Margin = sd.groupby(['Sub-Category'])['Profit'].mean().sort_values(ascend
ax=sns.barplot(x=Profit_Margin.head(6).values,y=Profit_Margin.head(6).index)
plt.title("Top 6 product sub-categories by profit margins")
plt.xlabel('Profit Margin')
plt.ylabel('sub-categories')
plt.savefig('Top 6 product sub-categories by profit margins.jpg')
plt.bar_label(ax.containers[0], fmt='%.2f', label_type='edge')
plt.show()
```



What is the distribution of order quantity?

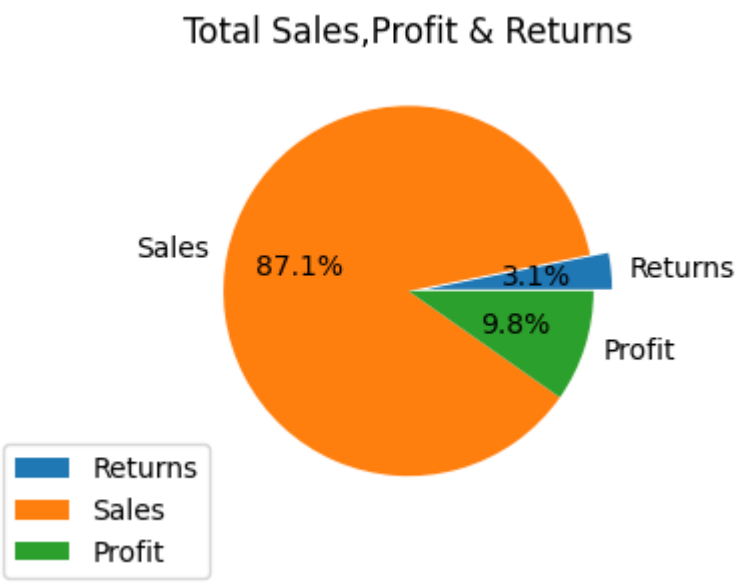
In [90]:
```
sns.histplot(sd['Quantity'], kde=True, bins=20)
plt.title('Distribution of Order Quantity')
plt.xlabel('Quantity')
plt.ylabel('Frequency')
plt.figure(figsize=(10, 6))
plt.savefig('Distribution of Order Quantity.jpg')
plt.show()
```




```
<Figure size 1000x600 with 0 Axes>
```

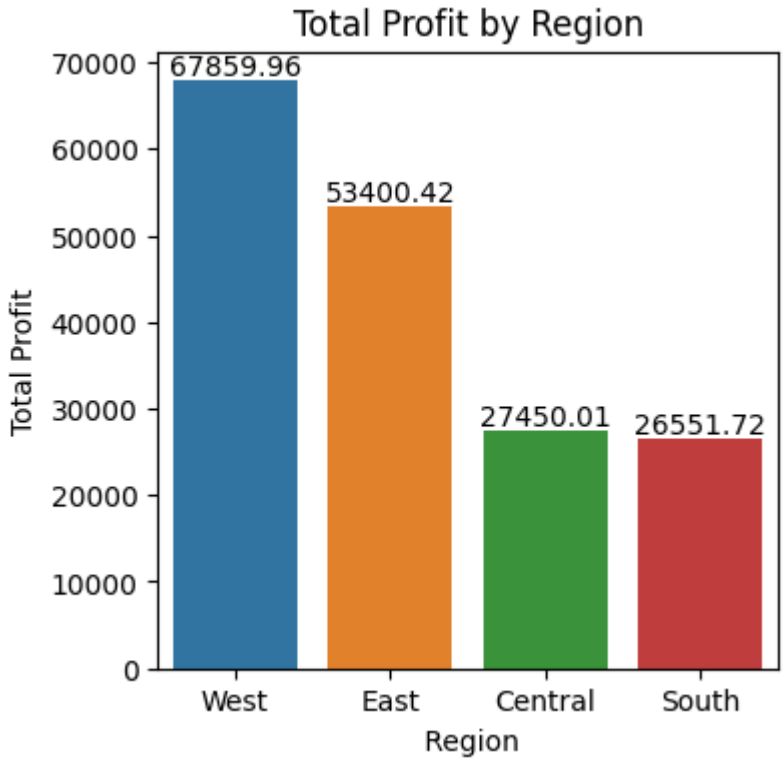
What is the total sales,profit percentage and N.of Returns percentage

In [88]:
```
total_Returns = sd['Returns'].sum()
total_sales = sd['Sales'].sum()
total_profit = sd['Profit'].sum()

plt.figure(figsize=(4, 3))
plt.pie(summary['Total'], labels=summary['Metric'], autopct='%1.1f%%', startangl
plt.title('Total Sales,Profit & Returns')
plt.legend(bbox_to_anchor=(0.1, 0.2))
plt.savefig('Total Sales,Profit & Returns.jpg')
plt.show()
```



Which regions contribute the most to total profit?

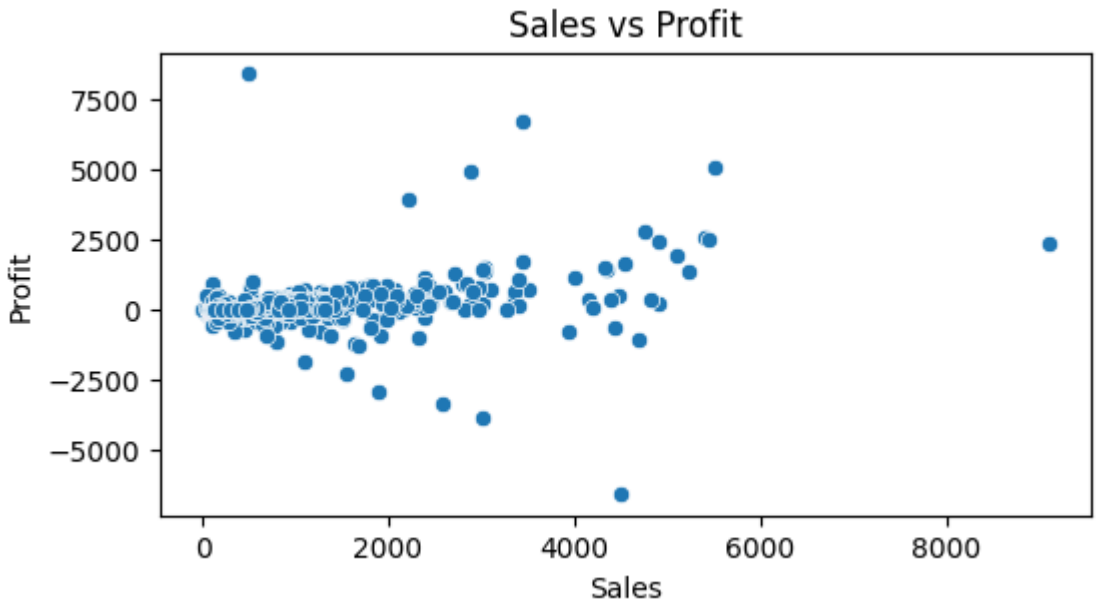
In [86]:
```
region_profit = sd.groupby('Region')['Profit'].sum().sort_values(ascending=False
plt.figure(figsize=(4, 4))
ax=sns.barplot(x=region_profit.index, y=region_profit.values)
plt.bar_label(ax.containers[0], fmt='%.2f', label_type='edge')
plt.title('Total Profit by Region')
plt.xlabel(' Region')
plt.ylabel('Total Profit')
plt.savefig('Total Profit by Region.jpg')
plt.show()
```



Sales vs Profit

In [85]:
```
plt.figure(figsize=(6, 3))
sns.scatterplot(x='Sales', y='Profit', data=sd)
plt.title('Sales vs Profit')
plt.xlabel('Sales')
plt.ylabel('Profit')
```

```
plt.savefig('Sales vs Profit.jpg')
plt.show()
```



What is the sales trend over time?

In [84]:
```
sd['Order Date'] = pd.to_datetime(sd['Order Date'])
sd['Year-Month'] = sd['Order Date'].dt.to_period('M')
monthly_sales = sd.groupby('Year-Month')['Sales'].sum()
plt.figure(figsize=(6, 3))
monthly_sales.plot(kind='line')
plt.title('Sales Trend Over Time')
plt.xlabel('Year-Month')
plt.ylabel('Total Sales')
plt.xticks(rotation=45)
plt.savefig('Sales Trend Over Time.jpg')
plt.show()
```