

```
In [19]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

pd.set_option('display.max_columns', None)

# Sample sales data
data = {
    "Date": pd.date_range(start="2025-01-01", periods=12, freq='M'),
    "Region": ["North", "South", "East", "West"] * 3,
    "Product": ["A", "B", "C"] * 4,
    "Sales": np.random.randint(1000, 5000, size=12),
    "Profit": np.random.randint(200, 1500, size=12)
}

df = pd.DataFrame(data)
df
```

Out[19]:

|    | Date       | Region | Product | Sales | Profit |
|----|------------|--------|---------|-------|--------|
| 0  | 2025-01-31 | North  | A       | 1269  | 263    |
| 1  | 2025-02-28 | South  | B       | 1281  | 862    |
| 2  | 2025-03-31 | East   | C       | 4505  | 417    |
| 3  | 2025-04-30 | West   | A       | 4074  | 1237   |
| 4  | 2025-05-31 | North  | B       | 2594  | 856    |
| 5  | 2025-06-30 | South  | C       | 2104  | 969    |
| 6  | 2025-07-31 | East   | A       | 4282  | 870    |
| 7  | 2025-08-31 | West   | B       | 1880  | 1136   |
| 8  | 2025-09-30 | North  | C       | 1503  | 1027   |
| 9  | 2025-10-31 | South  | A       | 4771  | 771    |
| 10 | 2025-11-30 | East   | B       | 3922  | 333    |
| 11 | 2025-12-31 | West   | C       | 3053  | 348    |

```
In [13]: # basic stats
df.describe()
```

Out[13]:

|       | Sales       | Profit      |
|-------|-------------|-------------|
| count | 12.000000   | 12.000000   |
| mean  | 3203.250000 | 803.583333  |
| std   | 883.836999  | 391.004873  |
| min   | 1935.000000 | 241.000000  |
| 25%   | 2540.250000 | 449.250000  |
| 50%   | 2964.000000 | 816.500000  |
| 75%   | 3655.750000 | 1099.250000 |
| max   | 4840.000000 | 1342.000000 |

```
In [14]: # Total Sales and Profit by Region
sales_by_region = df.groupby("Region")[["Sales","Profit"]].sum()
sales_by_region
```

Out[14]:

| Sales Profit |       |      |  |
|--------------|-------|------|--|
| Region       |       |      |  |
| East         | 8258  | 2400 |  |
| North        | 11376 | 3519 |  |
| South        | 8988  | 2712 |  |
| West         | 9817  | 1012 |  |

```
In [15]: # Average Sales by Product
avg_sales_product = df.groupby("Product")["Sales"].mean()
avg_sales_product
```

Out[15]:

```
Product
A      2866.50
B      3750.00
C      2993.25
Name: Sales, dtype: float64
```

```
In [16]: # Top performing product
df.sort_values(by="Sales", ascending=False)
```

Out[16]:

|    | Date       | Region | Product | Sales | Profit |
|----|------------|--------|---------|-------|--------|
| 4  | 2025-05-31 | North  | B       | 4840  | 1019   |
| 7  | 2025-08-31 | West   | B       | 4604  | 241    |
| 2  | 2025-03-31 | East   | C       | 3775  | 417    |
| 0  | 2025-01-31 | North  | A       | 3616  | 1160   |
| 9  | 2025-10-31 | South  | A       | 3463  | 790    |
| 1  | 2025-02-28 | South  | B       | 3008  | 1079   |
| 8  | 2025-09-30 | North  | C       | 2920  | 1340   |
| 11 | 2025-12-31 | West   | C       | 2761  | 460    |
| 10 | 2025-11-30 | East   | B       | 2548  | 641    |
| 5  | 2025-06-30 | South  | C       | 2517  | 843    |
| 3  | 2025-04-30 | West   | A       | 2452  | 311    |
| 6  | 2025-07-31 | East   | A       | 1935  | 1342   |

```
In [18]: # Sales above 4000
high_sales = df[df["Sales"] > 4000]
high_sales
```

Out[18]:

|   | Date       | Region | Product | Sales | Profit |
|---|------------|--------|---------|-------|--------|
| 4 | 2025-05-31 | North  | B       | 4840  | 1019   |
| 7 | 2025-08-31 | West   | B       | 4604  | 241    |

```
In [20]: # Profit margin calculation
```

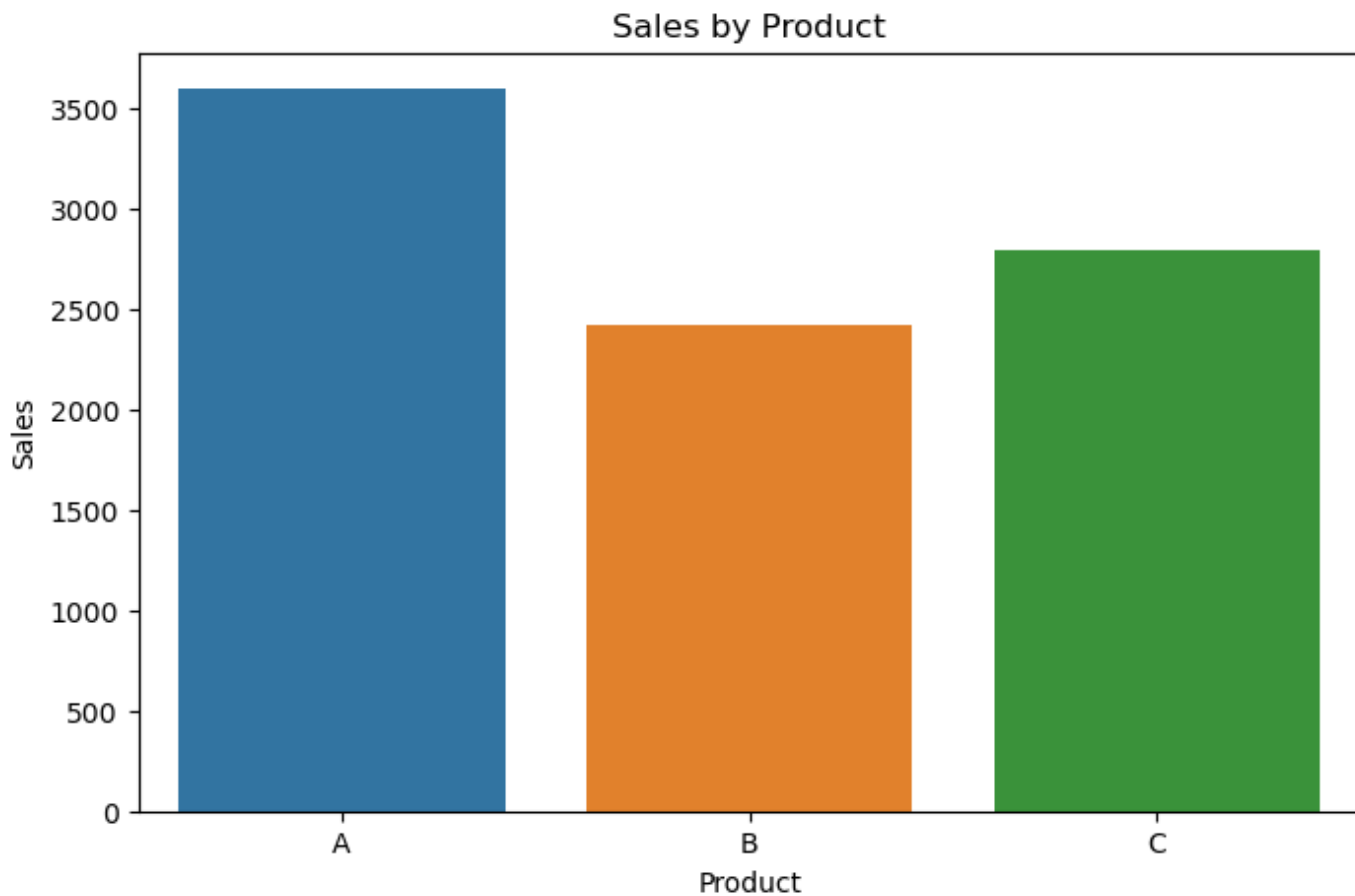
```
df["Profit_Margin"] = df["Profit"] / df["Sales"] * 100
df
```

Out[20]:

|    | Date       | Region | Product | Sales | Profit | Profit_Margin |
|----|------------|--------|---------|-------|--------|---------------|
| 0  | 2025-01-31 | North  | A       | 1269  | 263    | 20.724980     |
| 1  | 2025-02-28 | South  | B       | 1281  | 862    | 67.291179     |
| 2  | 2025-03-31 | East   | C       | 4505  | 417    | 9.256382      |
| 3  | 2025-04-30 | West   | A       | 4074  | 1237   | 30.363279     |
| 4  | 2025-05-31 | North  | B       | 2594  | 856    | 32.999229     |
| 5  | 2025-06-30 | South  | C       | 2104  | 969    | 46.055133     |
| 6  | 2025-07-31 | East   | A       | 4282  | 870    | 20.317609     |
| 7  | 2025-08-31 | West   | B       | 1880  | 1136   | 60.425532     |
| 8  | 2025-09-30 | North  | C       | 1503  | 1027   | 68.330007     |
| 9  | 2025-10-31 | South  | A       | 4771  | 771    | 16.160134     |
| 10 | 2025-11-30 | East   | B       | 3922  | 333    | 8.490566      |
| 11 | 2025-12-31 | West   | C       | 3053  | 348    | 11.398624     |

In [22]:

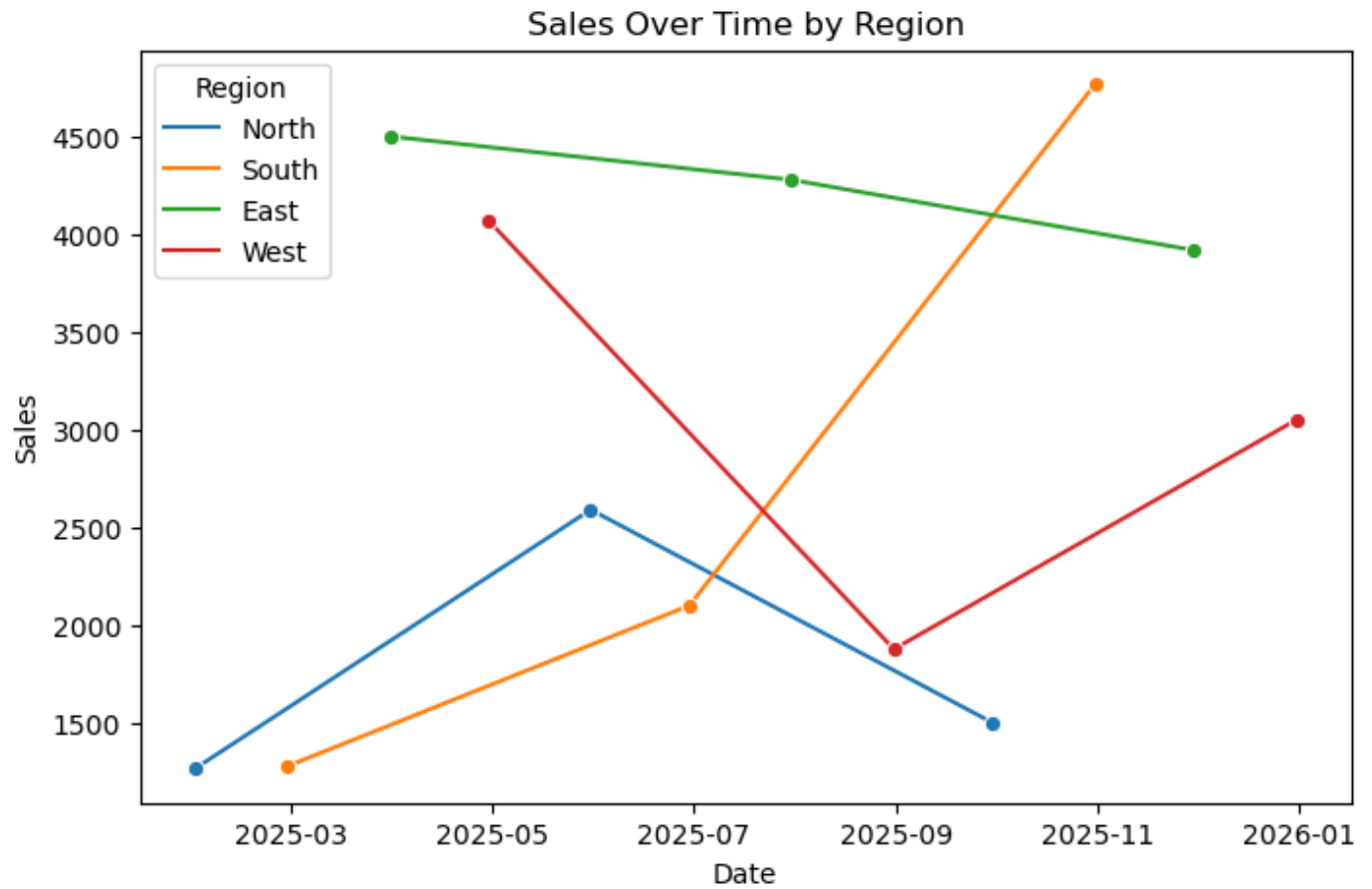
```
#Bar Chart
plt.figure(figsize=(8,5))
sns.barplot(x="Product", y="Sales", data=df, ci=None)
plt.title("Sales by Product")
plt.show()
```



In [23]:

```
#Line Chart
plt.figure(figsize=(8,5))
```

```
sns.lineplot(x="Date", y="Sales", hue="Region", data=df, marker='o')
plt.title("Sales Over Time by Region")
plt.show()
```

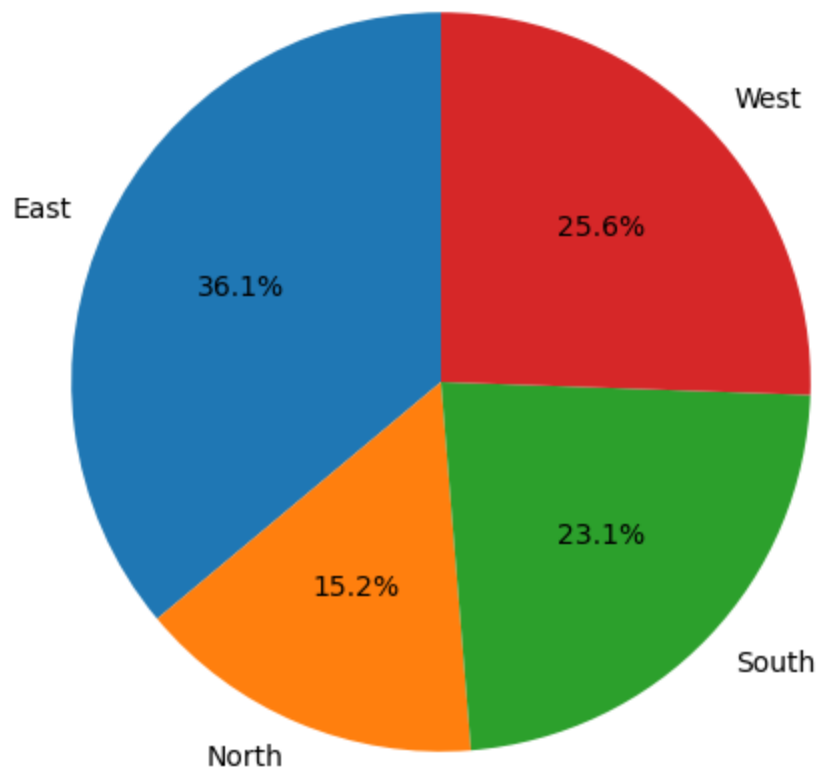


```
In [24]: #Scatter Plot
plt.figure(figsize=(8,5))
sns.scatterplot(x="Sales", y="Profit", hue="Product", size="Profit", data=df)
plt.title("Profit vs Sales")
plt.show()
```



```
In [25]: #Pie Chart
sales_by_region_sum = df.groupby("Region")["Sales"].sum()
plt.figure(figsize=(6,6))
plt.pie(sales_by_region_sum, labels=sales_by_region_sum.index, autopct="%1.1f%%", startangle=90)
plt.title("Sales Distribution by Region")
plt.show()
```

Sales Distribution by Region



```
In [26]: #Heatmap
numeric_df = df.select_dtypes(include='number')
plt.figure(figsize=(6,4))
sns.heatmap(numeric_df.corr(), annot=True, cmap="coolwarm")
plt.title("Correlation Between Sales, Profit, and Profit Margin")
plt.show()
```

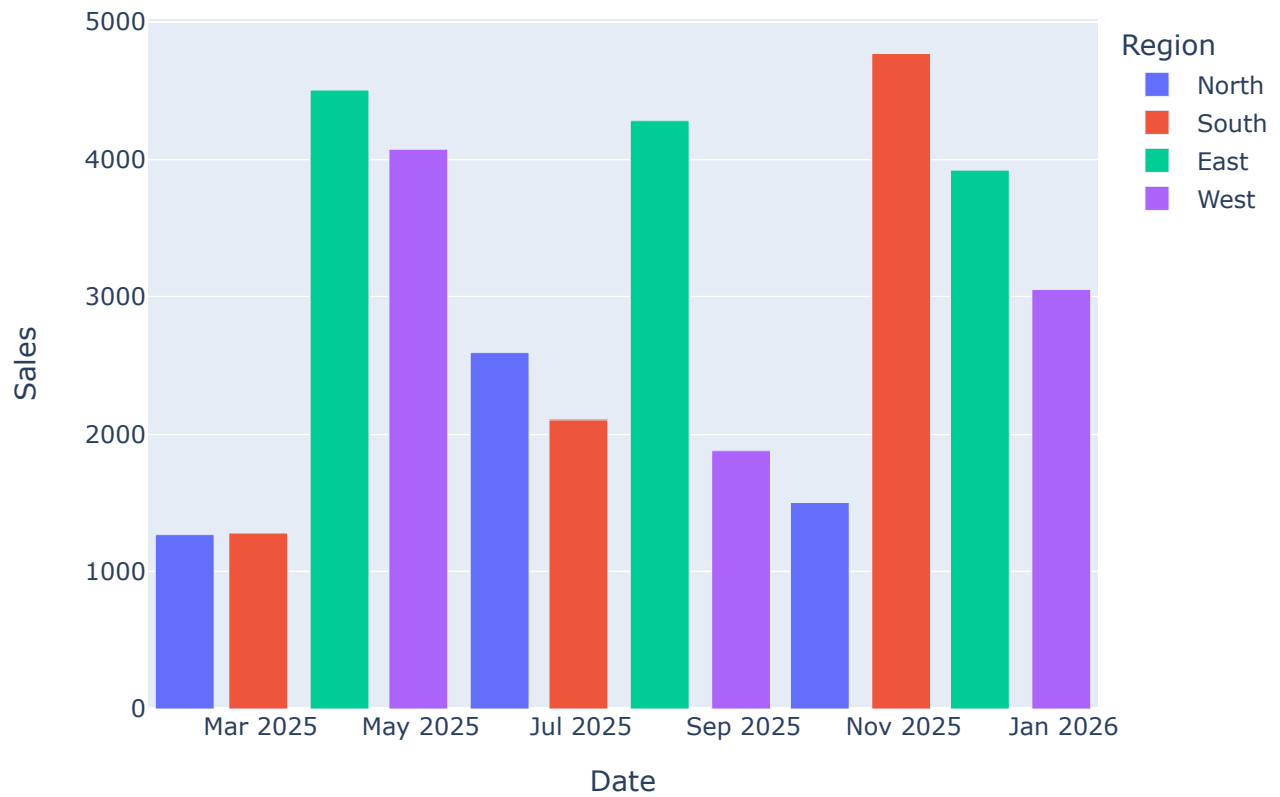
Correlation Between Sales, Profit, and Profit Margin



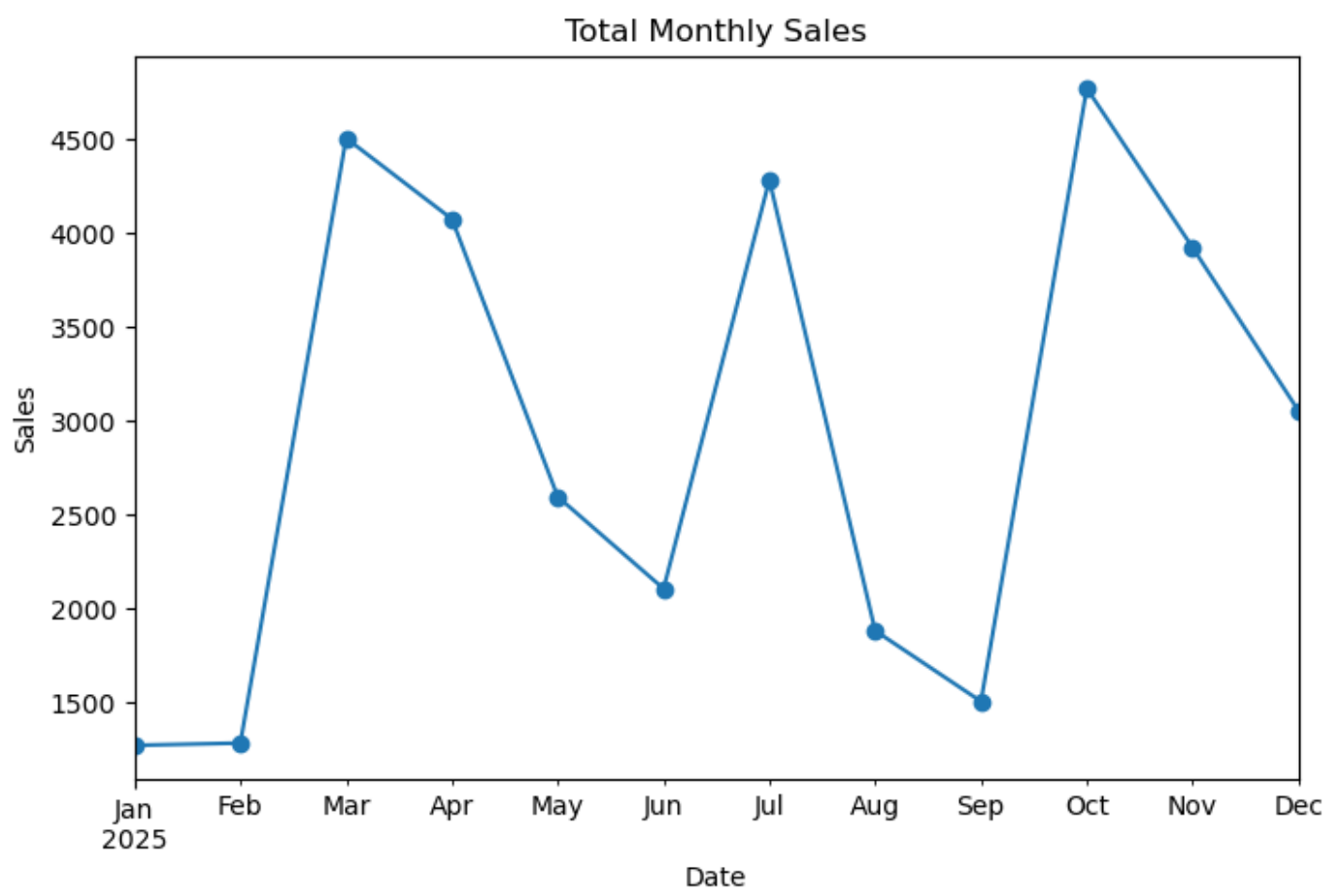
```
In [27]: #Interactive
fig = px.bar(df, x="Date", y="Sales", color="Region", title="Interactive Sales by Region")
```

```
fig.show()
```

## Interactive Sales by Region



```
In [28]: #Trends
# Monthly sales trend
monthly_sales = df.groupby("Date")["Sales"].sum()
monthly_sales.plot(figsize=(8,5), marker='o', title="Total Monthly Sales")
plt.ylabel("Sales")
plt.show()
```



In [ ]: