Practical 2 : Data Visualization Using Matplotlib

Name : $Deepak\ Kumar\ Bharti$

Roll No. : 25056765013

Group : A

Dataset for All Problems

Use the following dataset in all the problems below :

$import\ matplotlib.pyplot\ as\ plt$

months = $['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec']$

product_A = $[120, 135, 150, 160, 170, 180, 175, 165, 155, 145, 140, 130]$

product_B = $[100, 110, 115, 130, 140, 150, 160, 155, 145, 135, 125, 115]$

product_C = $[90, 95, 100, 105, 110, 120, 125, 130, 128, 120, 110, 100]$

product_D = $[80, 85, 90, 95, 100, 105, 110, 108, 102, 95, 90, 85]$

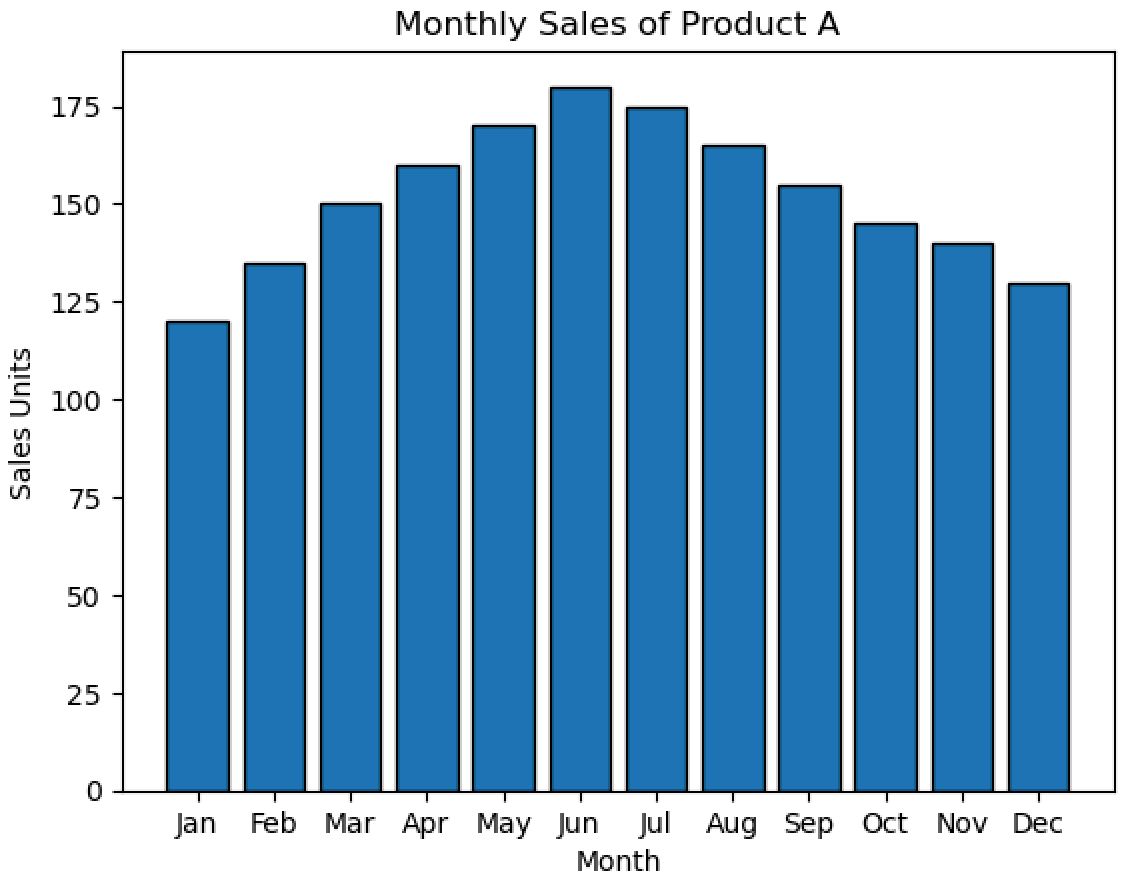
Problem 1 : Bar Chart

(a) Plot a **bar chart** showing the **monthly sales of Product A**.

(b) Label the x-axis as $Month$ and y-axis as $SalesUnits$.

In [1]:
```python
# Importing required libraries
import matplotlib.pyplot as plt
import numpy as np

# Given Dataset
months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'No

product_A = [120, 135, 150, 160, 170, 180, 175, 165, 155, 145, 140, 130]
product_B = [100, 110, 115, 130, 140, 150, 160, 155, 145, 135, 125, 115]
product_C = [90, 95, 100, 105, 110, 120, 125, 130, 128, 120, 110, 100]
product_D = [80, 85, 90, 95, 100, 105, 110, 108, 102, 95, 90, 85]
```

In [2]:
```python
# (a) Creating a bar chart for Product A
plt.figure()
plt.bar(months, product_A, edgecolor="black")

# (b) Adding labels and title
plt.xlabel("Month")
plt.ylabel("Sales Units")
plt.title("Monthly Sales of Product A")

# Displaying the chart
plt.show()
```



Problem 2 : Histogram

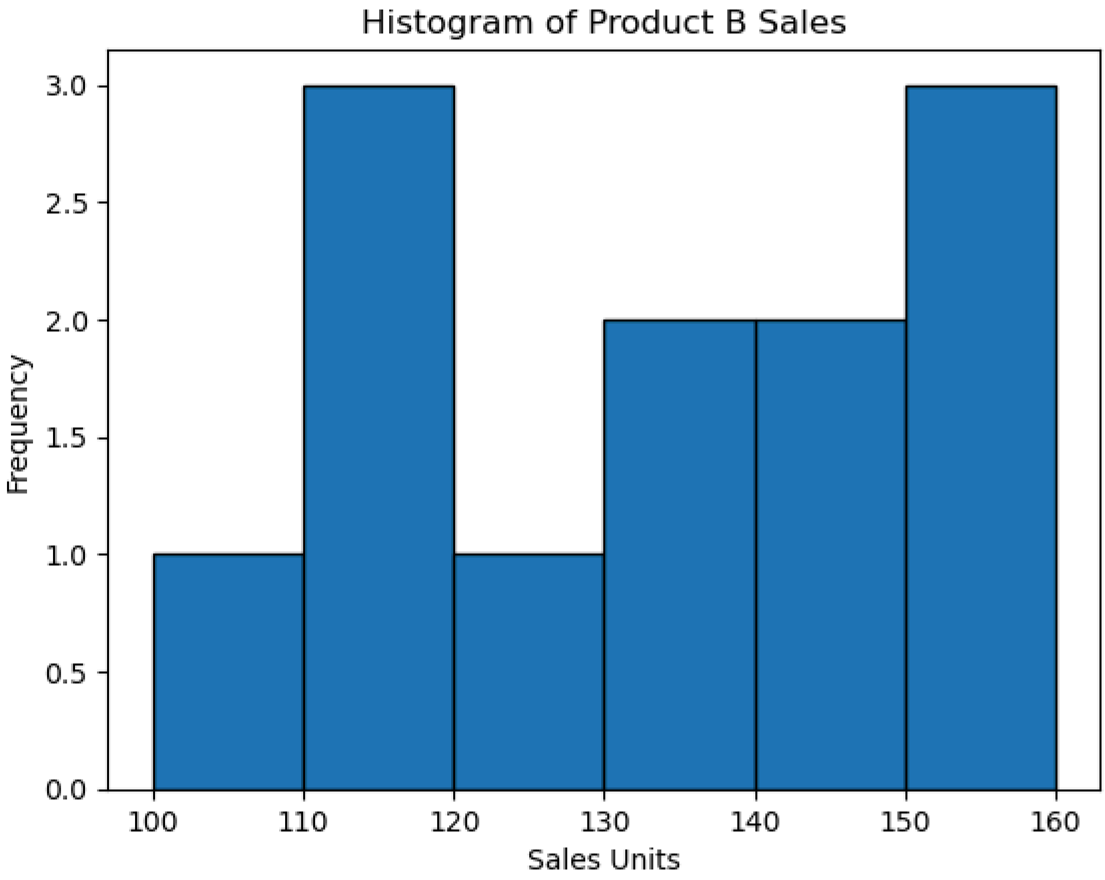
(a) Plot a **histogram** for the sales data of **Product B**.

(b) Use an appropriate number of bins.

(c) Label both axes and give a suitable title.

In [3]:

```
# (a) Creating histogram for Product B sales
plt.figure()

# (b) Using an appropriate number of bins.
plt.hist(product_B, bins=6, edgecolor="black")

# (c) Adding labels and title
plt.xlabel("Sales Units")
plt.ylabel("Frequency")
plt.title("Histogram of Product B Sales")

# Displaying the chart
plt.show()
```



Problem 3 : Pie Chart

(a) Compute the **total annual sales** for each product.

(b) Plot a **pie chart** showing the **percentage** contribution of each product to total sales.

(c) Display percentage values on the chart.

In [4]:

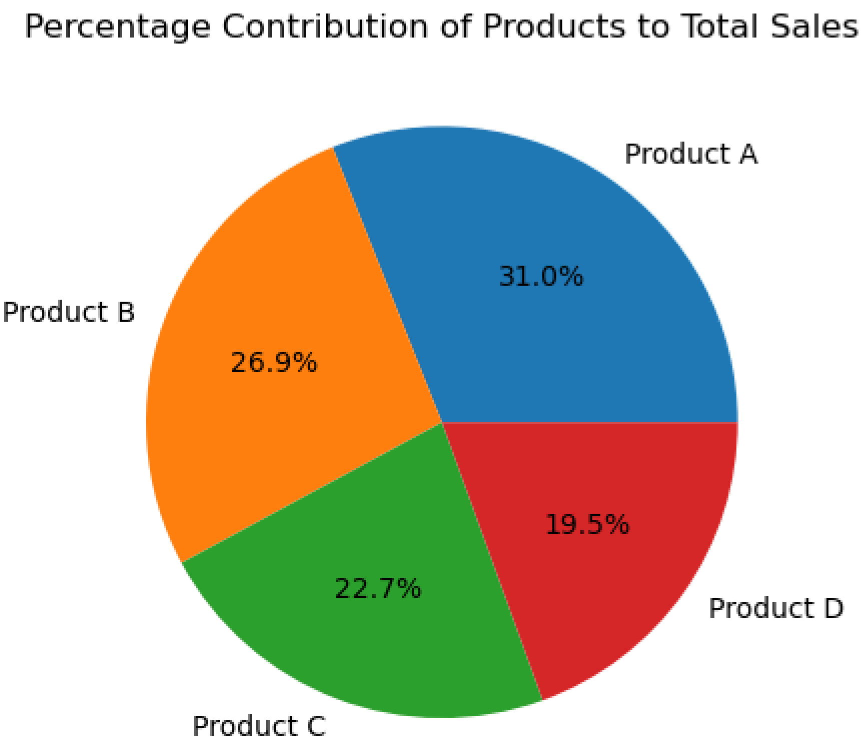
```
# (a) Computing total annual sales for each product
total_A = sum(product_A)
print("Total annual sales for Product A : ",total_A)
total_B = sum(product_B)
print("Total annual sales for Product B : ",total_B)
total_C = sum(product_C)
print("Total annual sales for Product C : ",total_C)
total_D = sum(product_D)
print("Total annual sales for Product D : ",total_D)
```

```
Total annual sales for Product A :  1825
Total annual sales for Product B :  1580
Total annual sales for Product C :  1333
Total annual sales for Product D :  1145
```

In [5]:

```
# (b) Combine totals and labels
sales = [total_A, total_B, total_C, total_D]
labels = ['Product A', 'Product B', 'Product C', 'Product D']

# (c) Create pie chart
```

```
plt.figure()
plt.pie(sales, labels=labels, autopct='%1.1f%%')

# Add title
plt.title("Percentage Contribution of Products to Total Sales")

# Display the chart
plt.show()
```



Problem 4 : Line Chart

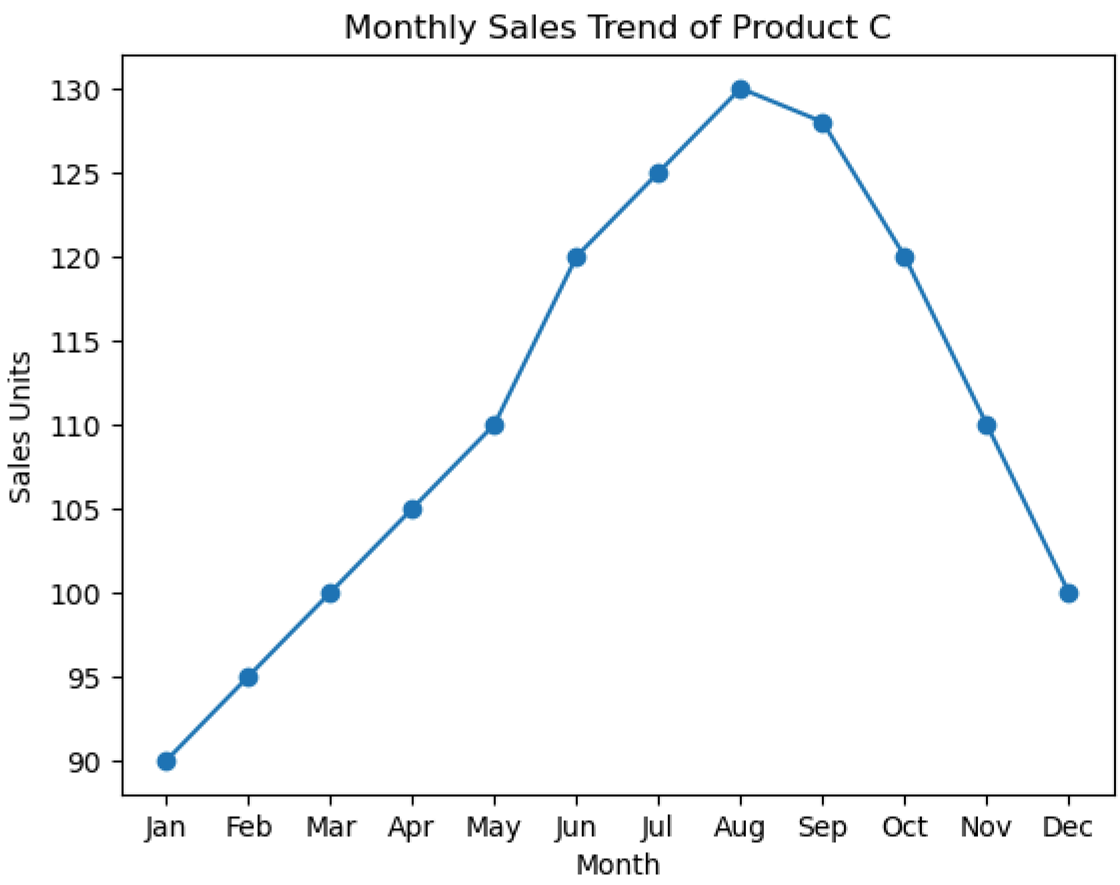
(a) Plot a **line chart** showing the **monthly sales trend of Product C**.

(b) Use markers to highlight data points.

(c) Label axes and add a suitable title.

In [6]:

```
# (a) Ploting a line chart for Product C
plt.figure()

# (b) Using markers to highlight data points.
plt.plot(months, product_C, marker='o')

# (c) Adding labels and title
plt.xlabel("Month")
plt.ylabel("Sales Units")
plt.title("Monthly Sales Trend of Product C")

# Display the chart
plt.show()
```



Problem 5 : Multiple Line Chart

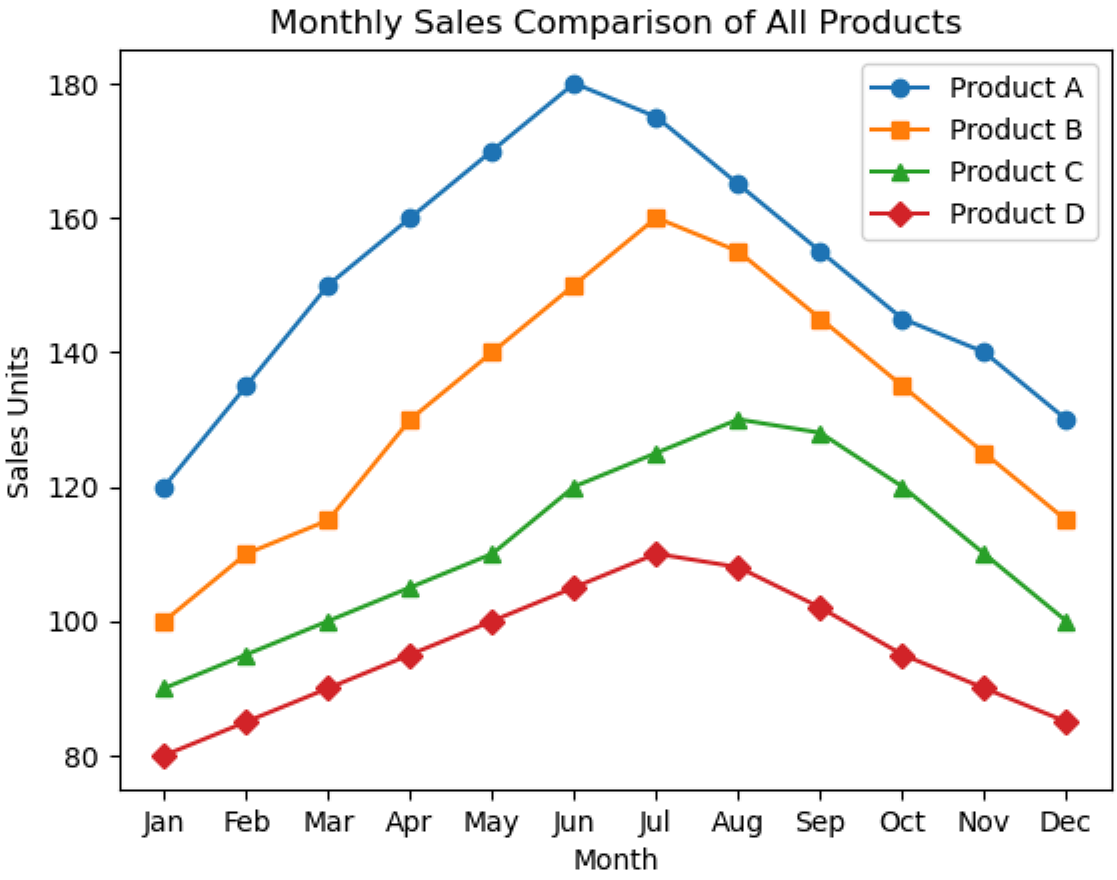
(a) Plot a **multiple line chart** comparing monthly sales of **Products A, B, C, and D**o.

(b) Add a legend to distinguish between products.

(c) Label axes and add an appropriate title.

In [7]:

```
# (a) Ploting a multiple line chart
plt.figure()

# Plot each product with a different marker
plt.plot(months, product_A, marker='o', label='Product A')   # Circle marker
plt.plot(months, product_B, marker='s', label='Product B')   # Square marker
plt.plot(months, product_C, marker='^', label='Product C')   # Triangle marker
plt.plot(months, product_D, marker='D', label='Product D')   # Diamond marker

# (b) Adding legend to distinguish products
plt.legend()

# (c) Add labels and title
plt.xlabel("Month")
plt.ylabel("Sales Units")
plt.title("Monthly Sales Comparison of All Products")

# Display the chart
plt.show()
```



Problem 6 : Stacked Bar Chart

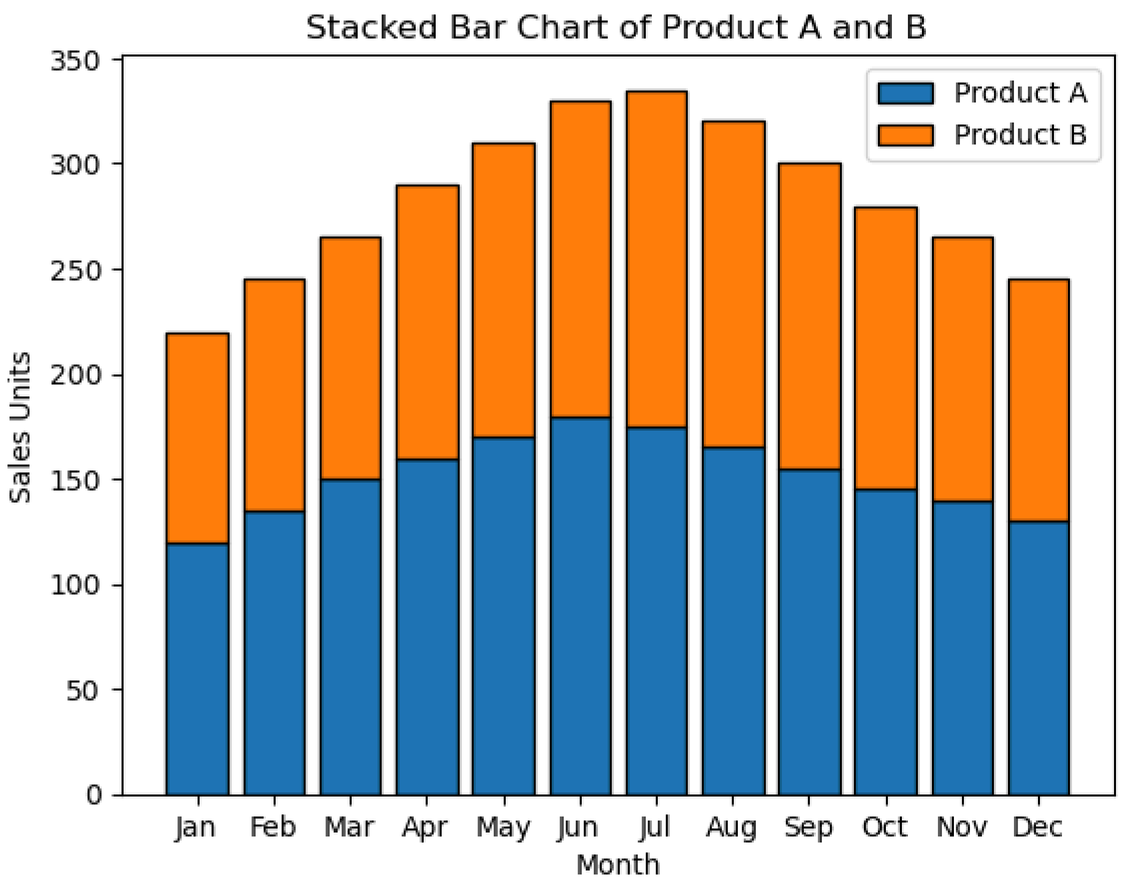
(a) Create a stacked bar chart showing monthly sales of **Product A and Product B**.

(b) Clearly indicate which portion of the bar corresponds to each product using a legend.

(c) Label axes and provide a suitable title.

In [8]:

```
# (a) Creating stacked bar chart
plt.figure()

# (b) Indicating the portion of the bars corresponds to each product using a legend
plt.bar(months, product_A, label='Product A', edgecolor="black")
plt.bar(months, product_B, bottom=product_A, label='Product B', edgecolor="black")

# Add legend
plt.legend()

# (c) Adding labels and title
plt.xlabel("Month")
plt.ylabel("Sales Units")
plt.title("Stacked Bar Chart of Product A and B")

# Display the chart
plt.show()
```



Problem 7 : Subplots

(a) Create a single figure containing **four subplots (2 × 2 layout)**.

(b) Display the following plots:

- Bar chart for Product A
- Histogram for Product B
- Line chart for Product C
- Stacked bar chart for Products A and B

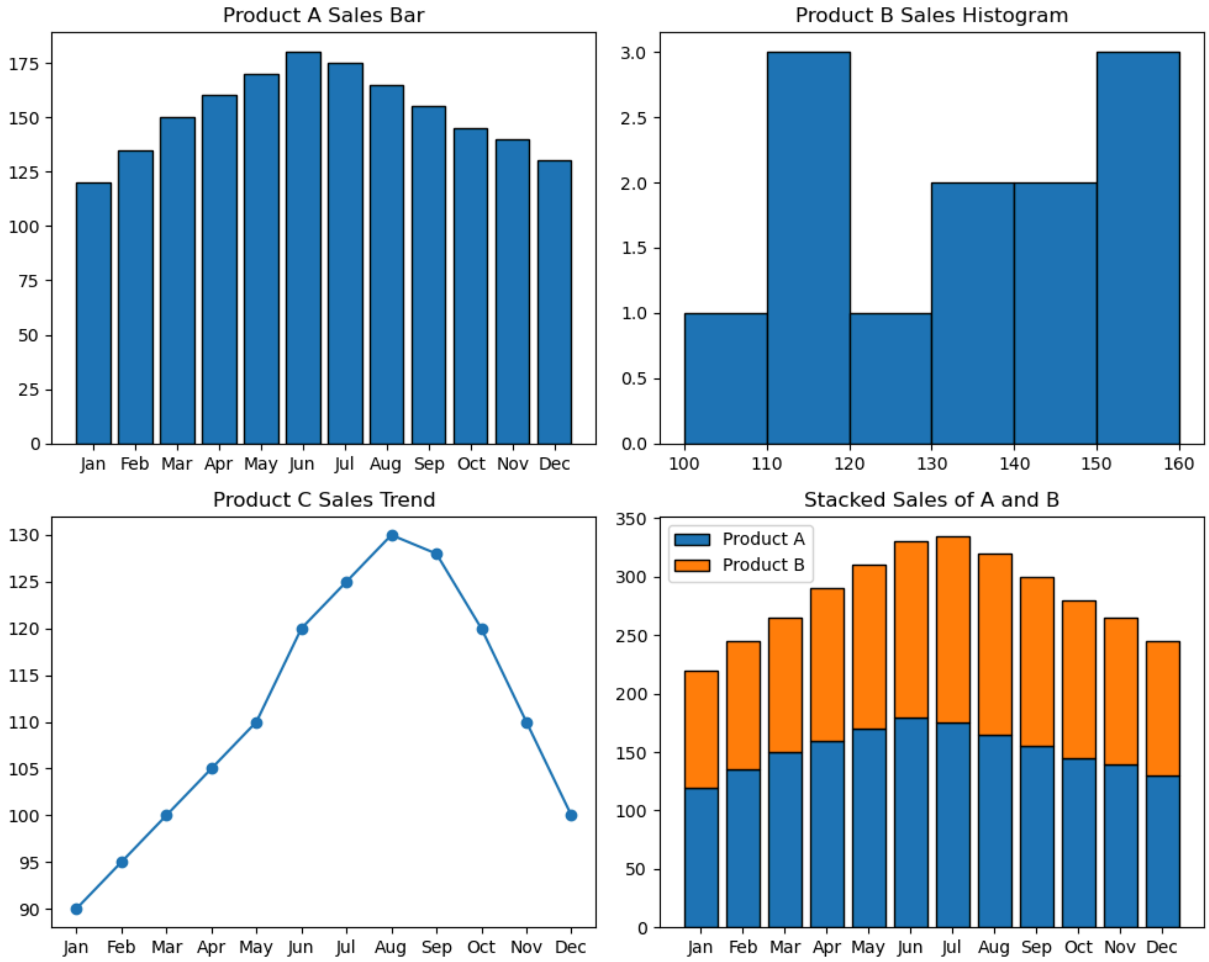
(c) Assign an appropriate title to each subplot.

(d) Adjust spacing so that plots do not overlap.

In [9]:

```
# Creating a single figure containing four subplots (2 × 2 layout).
plt.figure(figsize=(10, 8))

# Subplot 1: Bar chart (Product A)
plt.subplot(2, 2, 1)
plt.bar(months, product_A, edgecolor="black")
plt.title("Product A Sales Bar")

# Subplot 2: Histogram (Product B)
plt.subplot(2, 2, 2)
plt.hist(product_B, bins=6, edgecolor="black")
plt.title("Product B Sales Histogram")
```

```
# Subplot 3: Line chart (Product C)
plt.subplot(2, 2, 3)
plt.plot(months, product_C, marker='o')
plt.title("Product C Sales Trend")

# Subplot 4: Stacked bar chart (A & B)
plt.subplot(2, 2, 4)
plt.bar(months, product_A, label='Product A', edgecolor="black")
plt.bar(months, product_B, bottom=product_A, label='Product B', edgecolor="black")
plt.title("Stacked Sales of A and B")
plt.legend()

# Adjust layout to avoid overlap
plt.tight_layout()

# Display all subplots
plt.show()
```



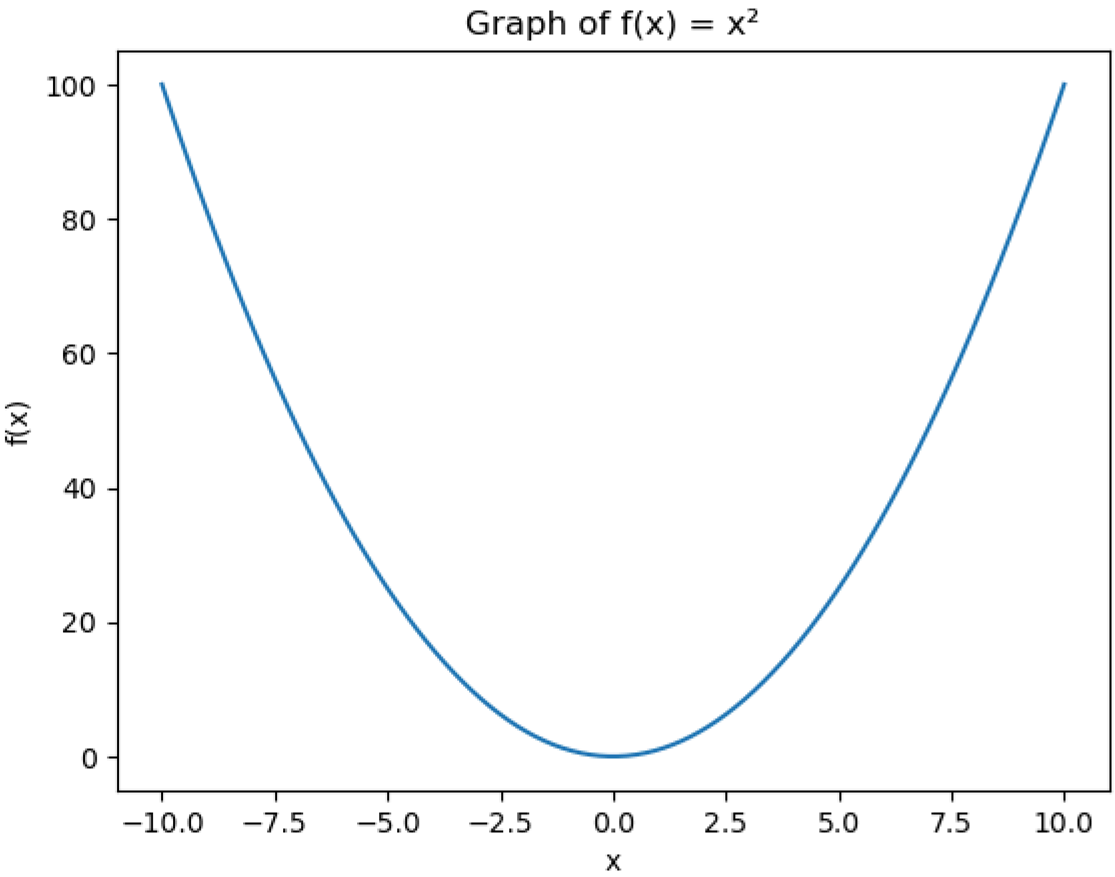
Problem 8 : Plotting a Mathematical Function (Not based on previous data)

(a) Plot a graph for the function f(x) = x^2. Define the function in Python as:

```
def f(x):
    return x**2
```

In [10]:

```python
# Define the function
def f(x):
    return x**2

# Generate x values
x = np.linspace(-10, 10, 100)

# Compute y values
y = f(x)

# Plot the function
plt.figure()
plt.plot(x, y)

# Add labels and title
plt.xlabel("x")
plt.ylabel("f(x)")
plt.title("Graph of f(x) = x²")

# Display the graph
plt.show()
```


Graph of f(x) = x²
f(x)
x