

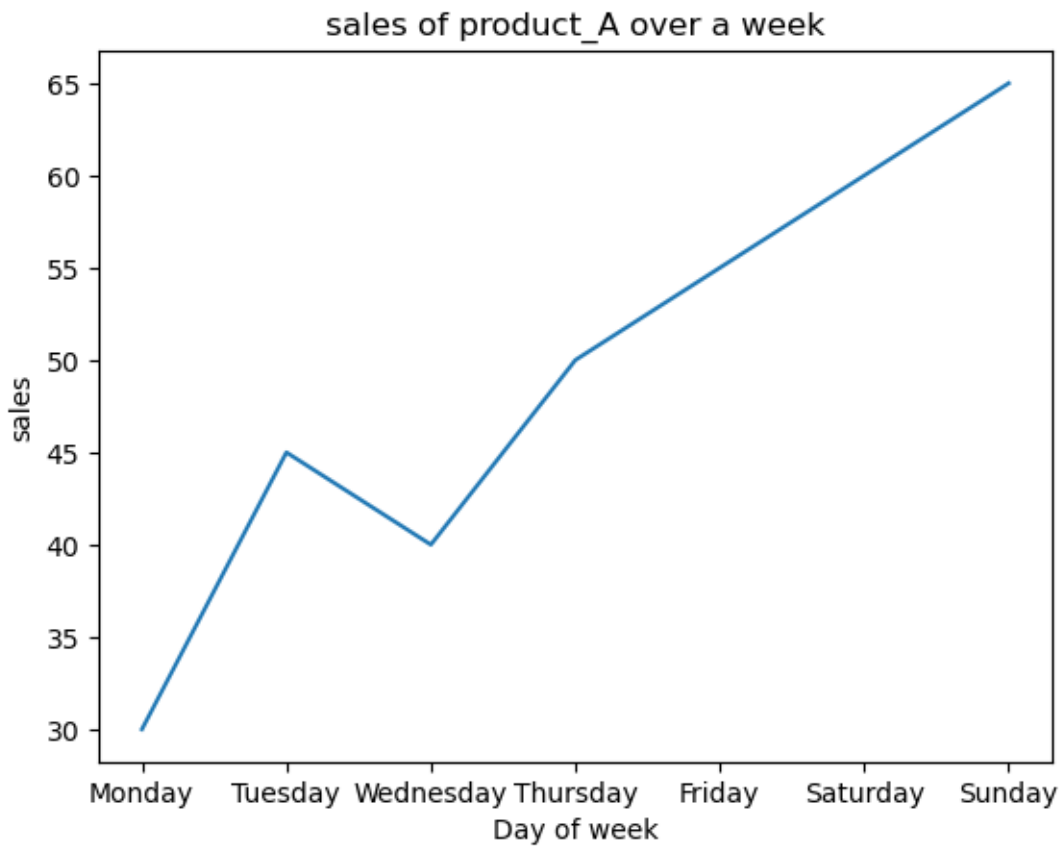
# basic\_ Matplotlib\_project

May 31, 2024

```
[1]: import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv('sales.csv')
```

0.1 1 Create a line plot for the sales of Product\_A over the week.

```
[2]: plt.plot(data["Day"],data['Product_A'])
plt.xlabel("Day of week")
plt.ylabel("sales")
plt.title("sales of product_A over a week")
plt.show()
```

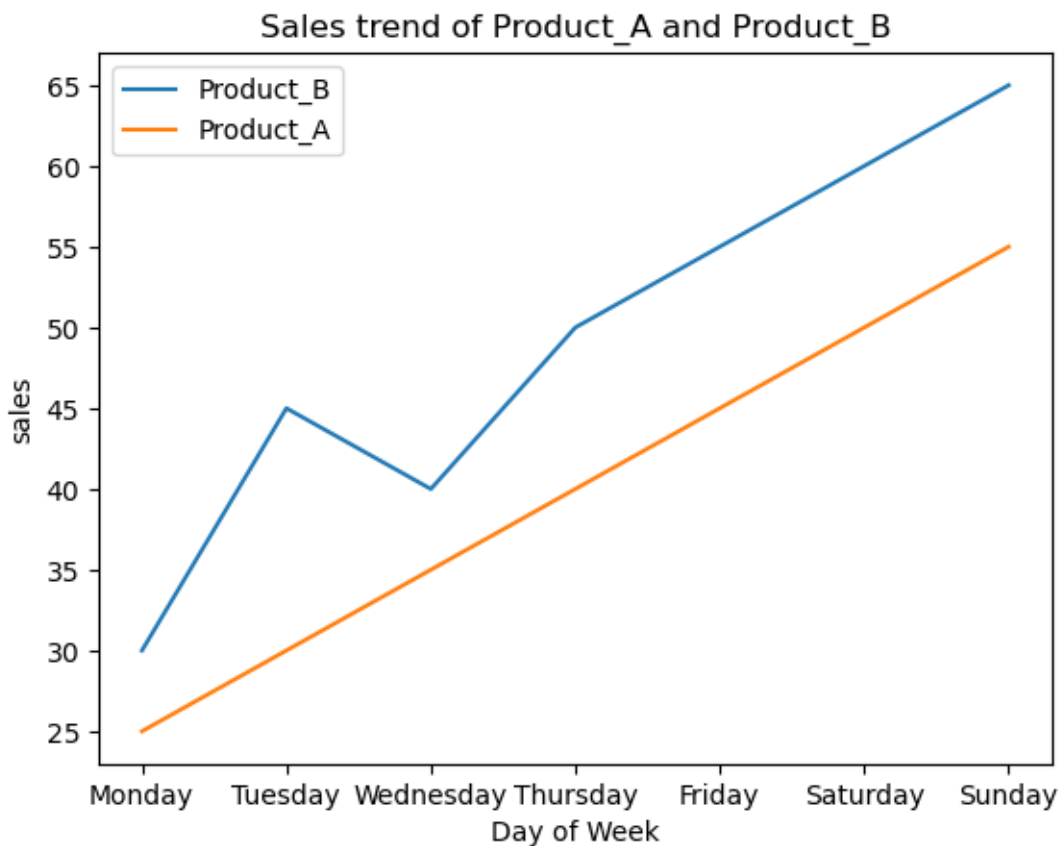


### 0.1.1 Observation:

The sales of Product\_A show a general upward trend throughout the week, starting at 30 units on Monday and reaching 65 units by Sunday.

## 0.2 2 Create a line plot showing the sales trends for Product\_A and Product\_B on the same graph.

```
[3]: plt.plot(data["Day"],data["Product_A"],label = "Product_B")
plt.plot(data["Day"],data["Product_B"],label ="Product_A")
plt.xlabel("Day of Week")
plt.ylabel("sales")
plt.title("Sales trend of Product_A and Product_B")
plt.legend()
plt.show()
```

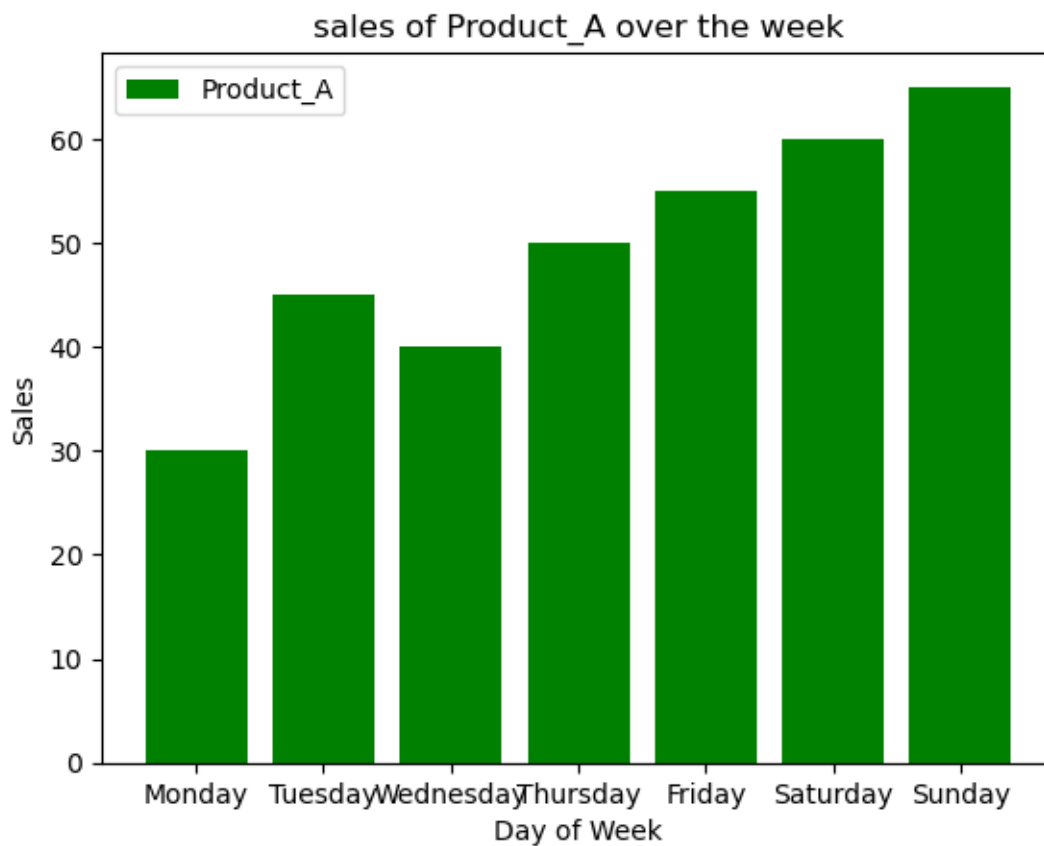


### Observation: Both Product\_A and Product\_B exhibit an upward trend over the week. Product\_B consistently has lower sales compared to Product\_A, but both products follow a

similar trend pattern.

### 0.3 3 Create a bar plot for the sales of Product\_A over the week.

```
[4]: plt.bar(data["Day"],data["Product_A"],label = "Product_A", color = "green")
plt.xlabel("Day of Week")
plt.ylabel("Sales")
plt.title("sales of Product_A over the week")
plt.legend()
plt.show()
```

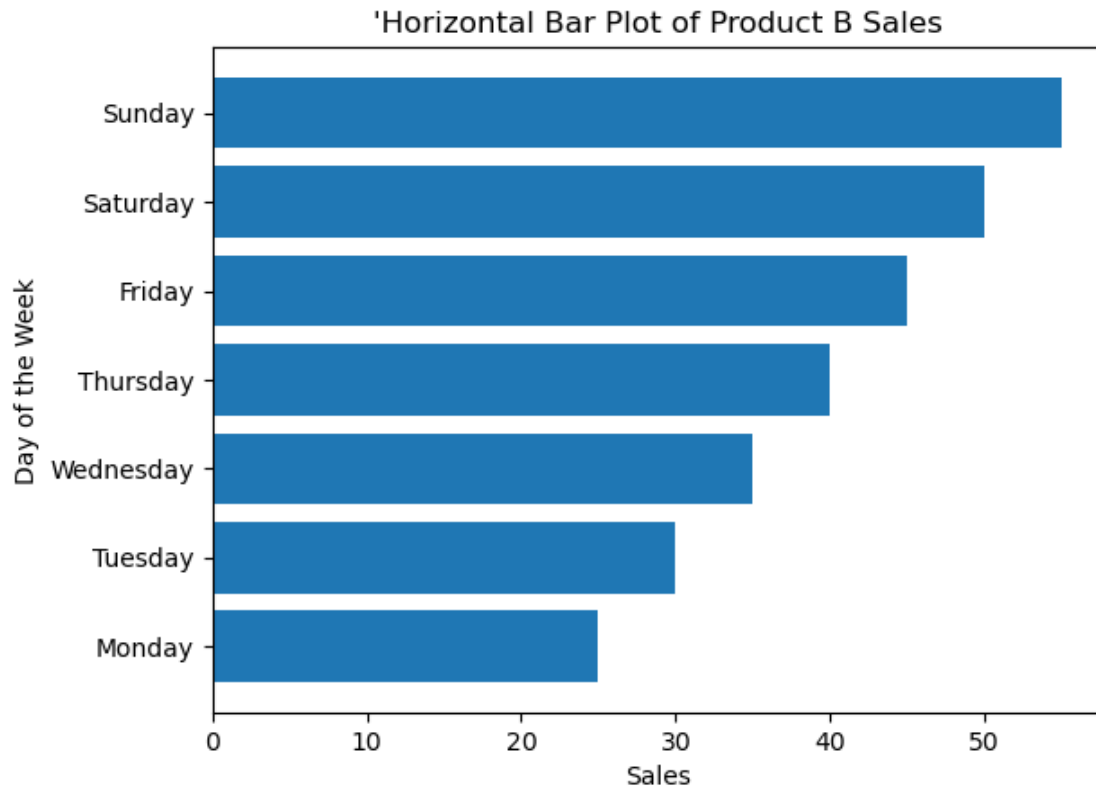


#### 0.3.1 Observation:

The bar plot highlights the increasing sales of Product\_A from Monday to Sunday. Each day shows a progressive increase in sales, with the highest sales on Sunday.

#### 0.4 4 Create a horizontal bar plot for the sales of Product\_B over the week.

```
[5]: plt.barh(data["Day"],data["Product_B"])  
plt.xlabel("Sales")  
plt.ylabel("Day of the Week")  
plt.title("'Horizontal Bar Plot of Product B Sales")  
plt.show()
```

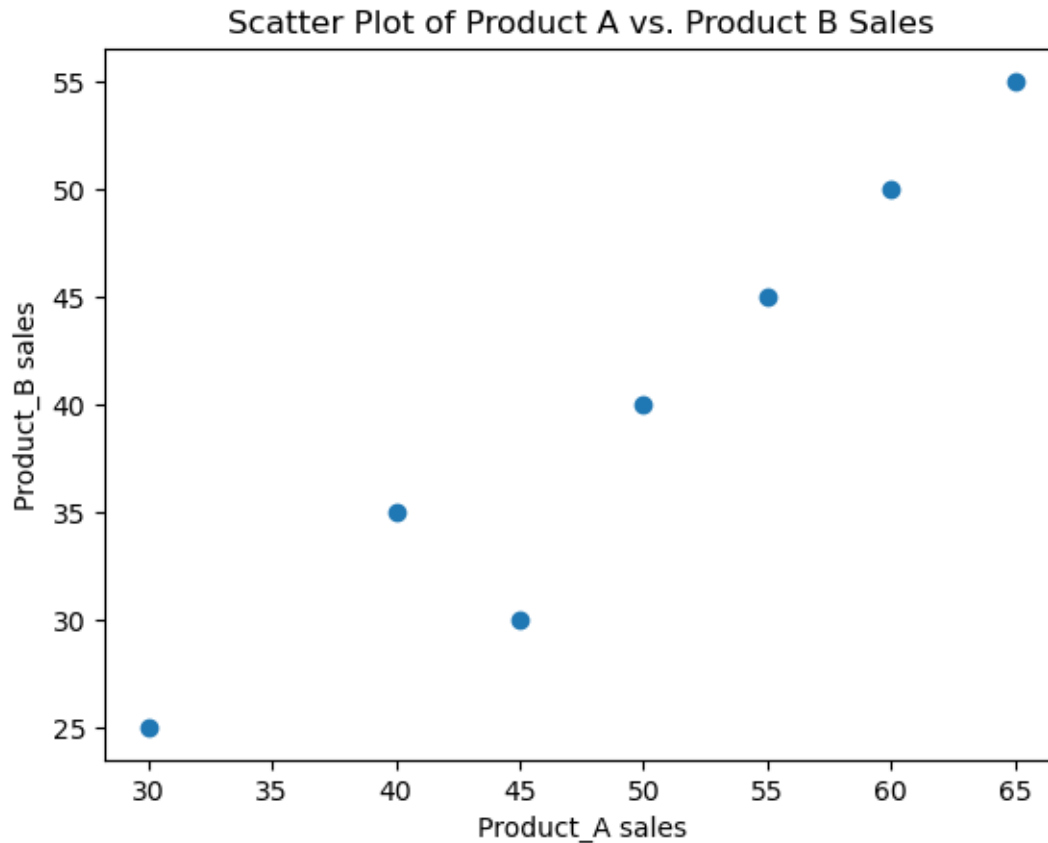


##### 0.4.1 Observation:

The horizontal bar plot demonstrates the steady increase in Product\_B sales over the week. The sales figures increase daily, peaking on Sunday.

#### 0.5 5 Create a scatter plot for Product\_A sales against Product\_B sales.

```
[6]: plt.scatter(data["Product_A"],data["Product_B"])  
plt.xlabel("Product_A sales")  
plt.ylabel("Product_B sales")  
plt.title("Scatter Plot of Product A vs. Product B Sales")  
plt.show()
```



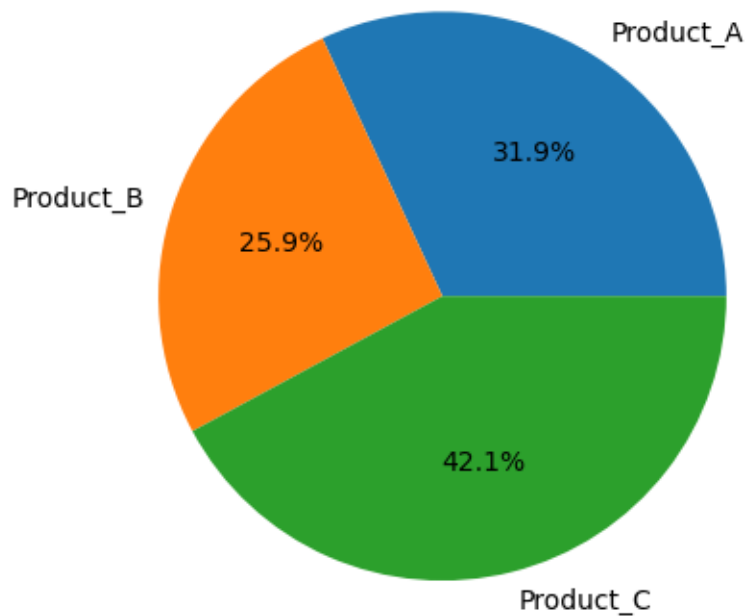
#### 0.5.1 Observation:

The scatter plot shows a positive correlation between the sales of Product\_A and Product\_B. As the sales of Product\_A increase, the sales of Product\_B also tend to increase, indicating a similar sales pattern for both products.

#### 0.6 6 Create a pie chart showing the total sales distribution of Product\_A, Product\_B, and Product\_C over the week.

```
[7]: total_sales = data[["Product_A", "Product_B", "Product_C"]].sum()
plt.pie(total_sales, labels = total_sales.index, autopct="%1.1f%")
plt.title("Total Sales Distribution of Products")
plt.show()
```

Total Sales Distribution of Products

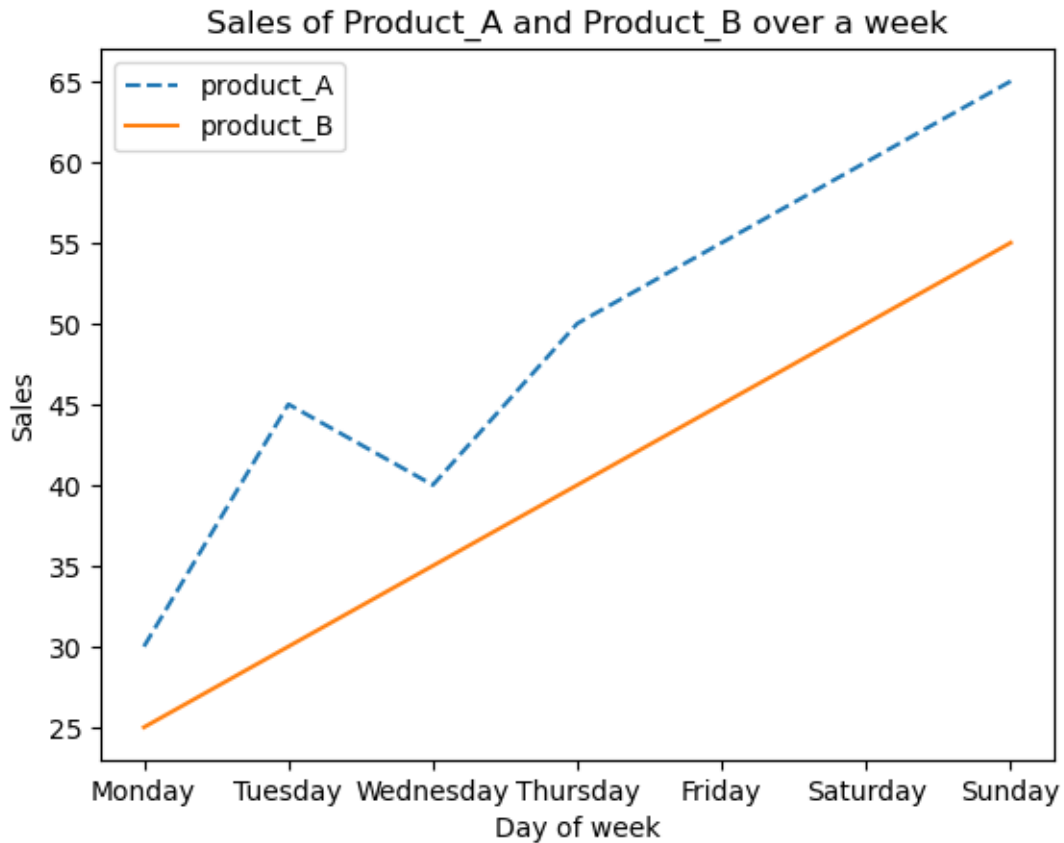


#### 0.6.1 Observation:

The pie chart indicates that Product\_C has the highest total sales over the week, followed by Product\_A and Product\_B. Product\_C accounts for the largest portion of the total sales distribution.

#### 0.7 7 Plot the sales of Product\_A and Product\_B with different line styles (e.g., solid, dashed).

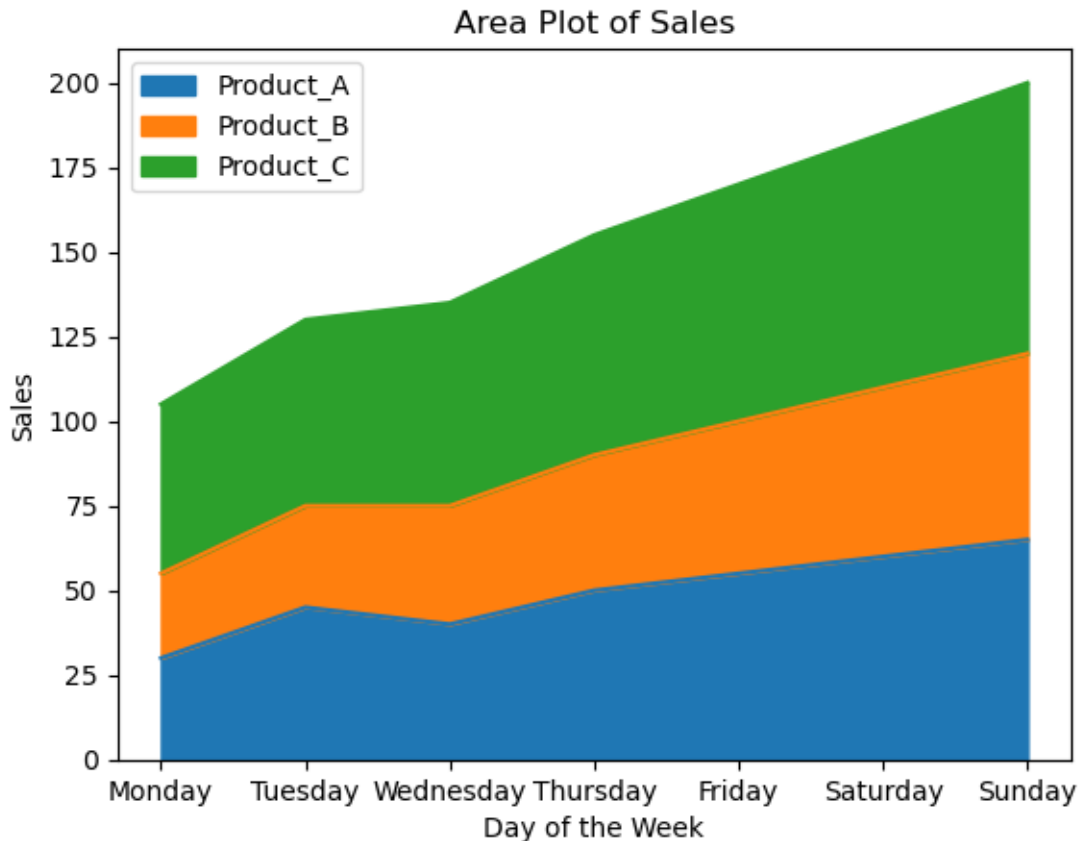
```
[8]: plt.plot(data["Day"],data["Product_A"],linestyle="--",label = "product_A")
plt.plot(data["Day"],data["Product_B"],linestyle = "-",label = "product_B")
plt.xlabel("Day of week")
plt.ylabel("Sales")
plt.title("Sales of Product_A and Product_B over a week")
plt.legend()
plt.show()
```



### Observation: The plot shows the sales trends of Product\_A (dashed line) and Product\_B (solid line) with different line styles. The different styles help in distinguishing between the two products' sales trends, which both show an upward trend over the week.

**0.8 8 Create an area plot to show the cumulative sales of Product\_A, Product\_B, and Product\_C over the week.**

```
[9]: data.plot.area(x="Day",y = ["Product_A","Product_B","Product_C"])
plt.xlabel('Day of the Week')
plt.ylabel('Sales')
plt.title('Area Plot of Sales')
plt.show()
```

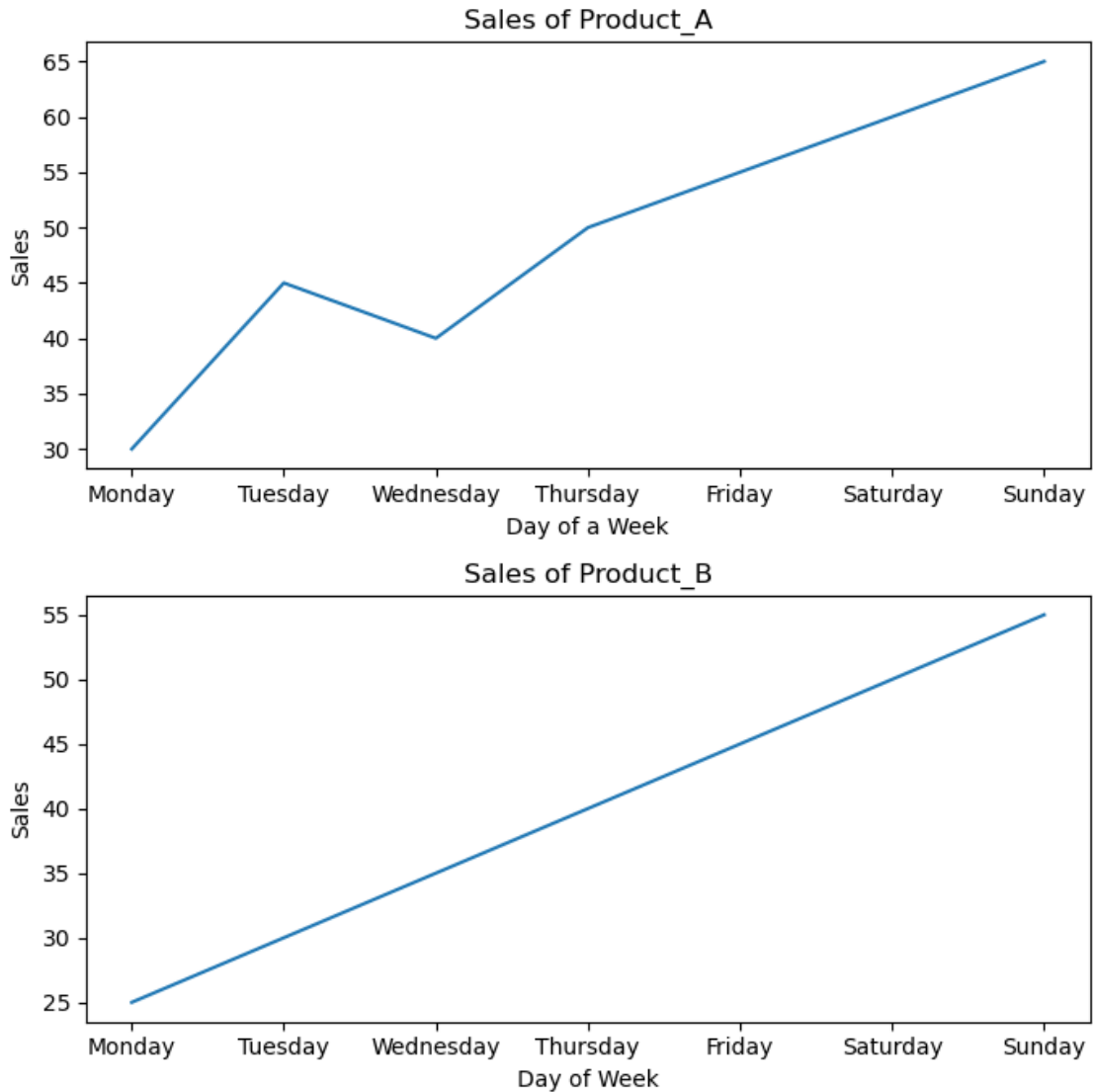


### Observation: The area plot depicts the cumulative sales of the three products. Product\_C has the largest area, indicating the highest sales, followed by Product\_A and Product\_B. The plot visually emphasizes the cumulative contribution of each product to total sales.

**0.9 9 Create two subplots in a single figure: one for Product\_A sales and one for Product\_B sales.**

```
[10]: fig, (plt1,plt2)= plt.subplots(2,1, figsize= (7,7))
plt1.plot(data["Day"],data["Product_A"],label = "Product_A")
plt1.set_title("Sales of Product_A")
plt1.set_xlabel("Day of a Week")
plt1.set_ylabel("Sales")
plt2.plot(data["Day"],data["Product_B"],label = "Product_B")
plt2.set_title("Sales of Product_B")
plt2.set_xlabel("Day of Week")
plt2.set_ylabel("Sales")
plt.tight_layout()
plt.show()
```





### 0.9.1 Observation:

The first subplot shows the upward trend in Product\_A sales over the week, while the second subplot shows a similar trend for Product\_B. Both subplots illustrate the increasing sales pattern for the respective products.

**0.10 10 Create three subplots in a single figure: one for Product\_A sales , one for Product\_B sales and one for Product\_C.**

[ ]:

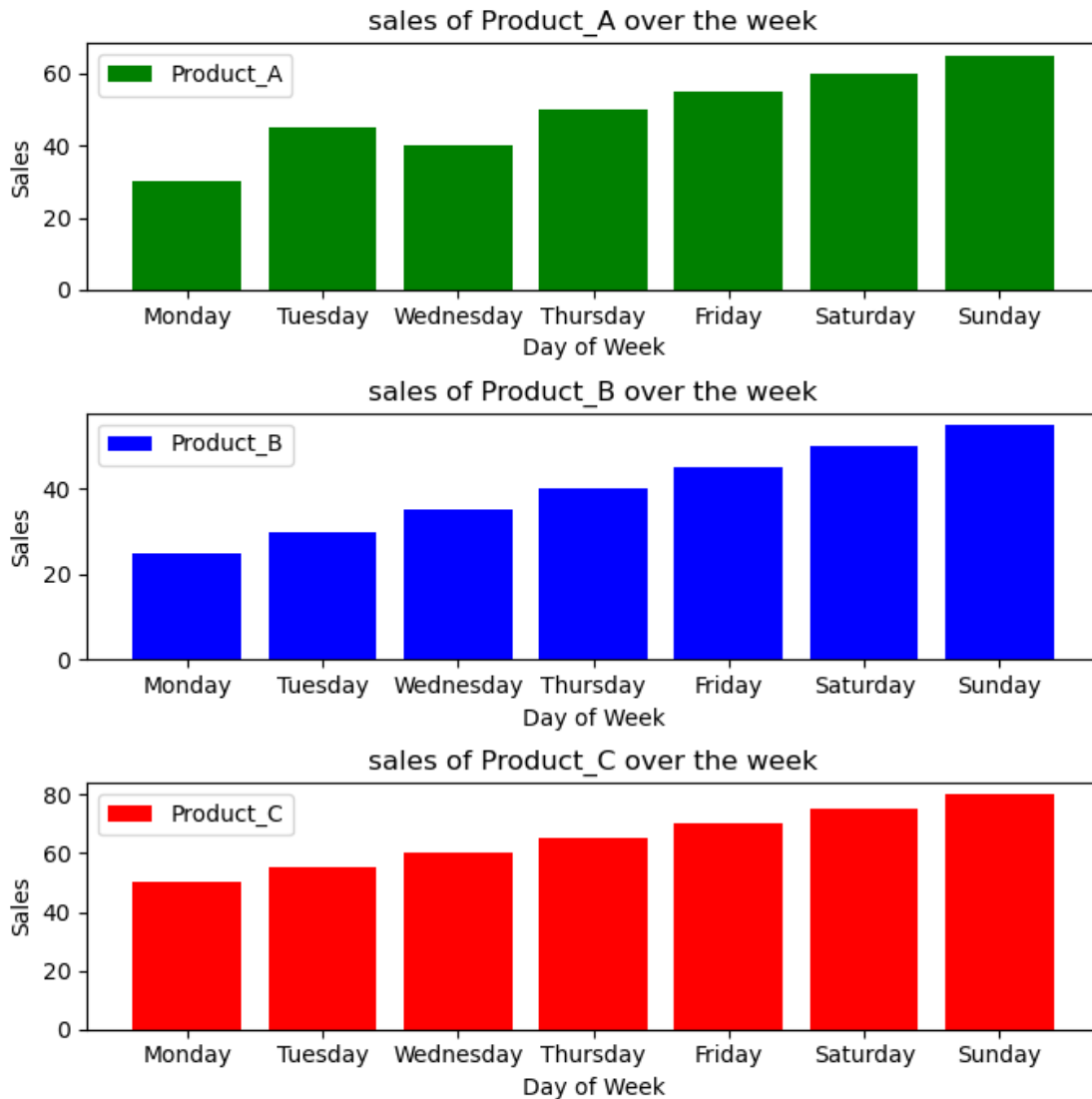
```
[11]: fig, (plt1,plt2, plt3)= plt.subplots(3,1, figsize= (7,7))
plt1.bar(data["Day"],data["Product_A"],label = "Product_A", color = "green")
```

```
plt1.set_xlabel("Day of Week")
plt1.set_ylabel("Sales")
plt1.set_title("sales of Product_A over the week")
plt1.legend()

plt2.bar(data["Day"],data["Product_B"],label = "Product_B", color = "blue")
plt2.set_xlabel("Day of Week")
plt2.set_ylabel("Sales")
plt2.set_title("sales of Product_B over the week")
plt2.legend()

plt3.bar(data["Day"],data["Product_C"],label = "Product_C", color = "red")
plt3.set_xlabel("Day of Week")
plt3.set_ylabel("Sales")
plt3.set_title("sales of Product_C over the week")
plt3.legend()

plt.tight_layout()
plt.subplots_adjust(hspace=0.5)
plt.show()
```



#### 0.10.1 Observation:

Each subplot individually highlights the sales trend of Product\_A, Product\_B, and Product\_C over the week. Product\_C shows the highest sales, followed by Product\_A and Product\_B. All products exhibit an increasing sales trend from Monday to Sunday.

[ ]: