VISUALIZATION

1.Introduction :

Data visualization is an important part of data science and analytics.
It helps us present complex information in a clear and visual way using **graphs, charts, and plots**.
Through visualization, we can easily identify patterns, trends, and relationships that are not immediately obvious from raw numbers.

Python provides several libraries for visualization. Among them, **Matplotlib** and **Plotly** are two of the most popular.
This guide explains both libraries in detail, covering their features, use cases, various graph types, code examples, and a comparison to help you choose the right tool.

2. Matplotlib

Overview

Matplotlib is one of the most widely used and foundational Python libraries for creating visualizations.
It was developed by *John D. Hunter* in 2003 to offer MATLAB-style plotting features in Python.
With Matplotlib, you can create line plots, bar charts, scatter plots, histograms, pie charts, heatmaps, and more — all with complete customization.

Key Features

- Supports both 2D and basic 3D plotting.
- Fully customizable (colors, labels, titles, grids, etc.).
- Works smoothly with **NumPy** and **Pandas**.

- The foundation for higher-level libraries like Seaborn and Pandas' plotting functions.

Some Graph Types and Examples in Matplotlib :

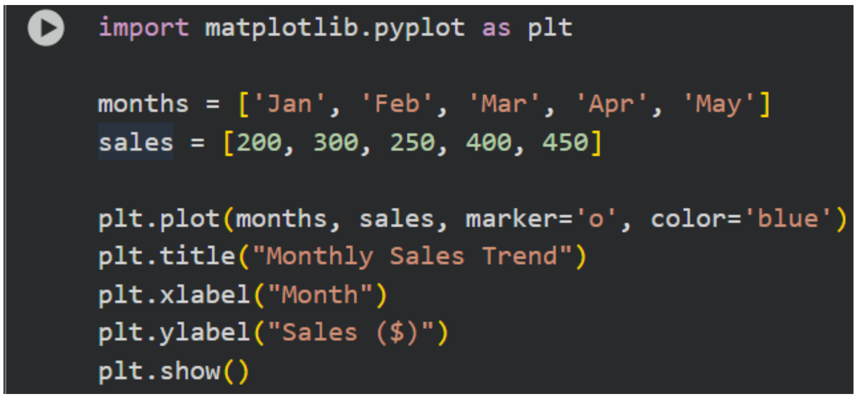
Use Case: ***Retail Sales Analysis***

We'll use retail data examples to visualize sales performance, trends, and distribution using Matplotlib.

2.1. Line Graph

Purpose: To show the trend of monthly sales over time.

Code snippet:

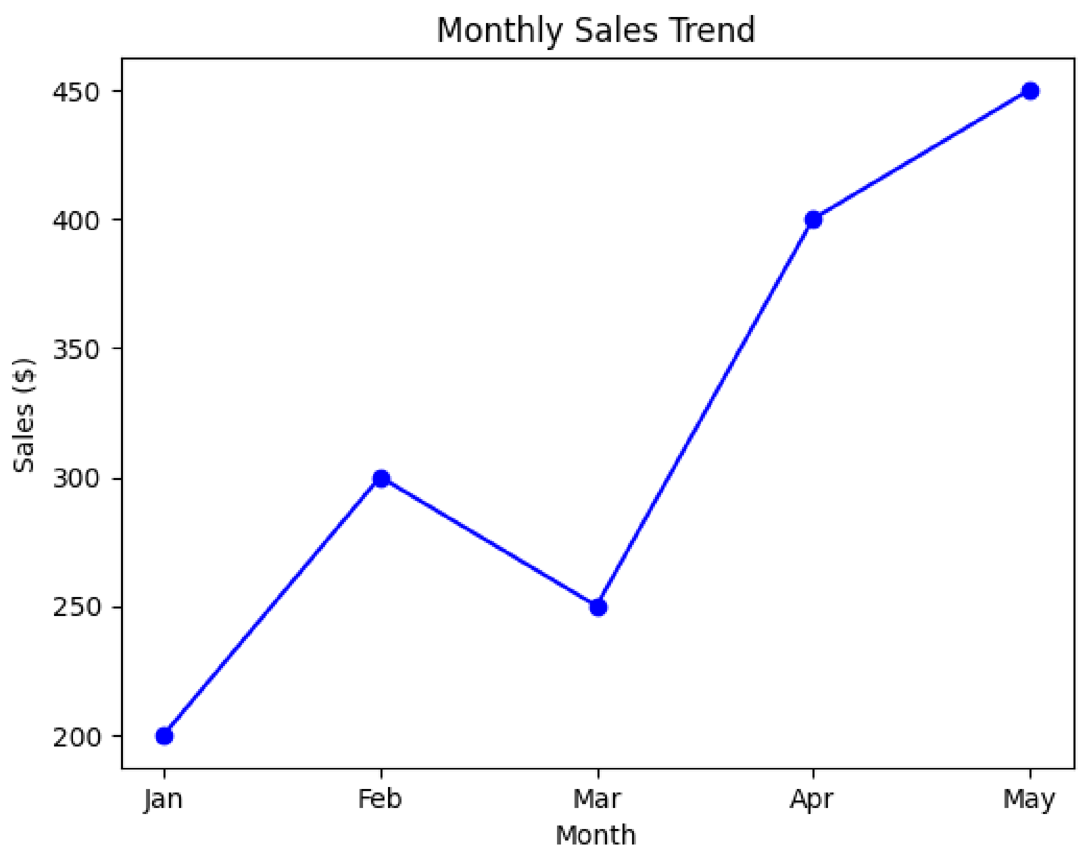


```python
import matplotlib.pyplot as plt

months = ['Jan', 'Feb', 'Mar', 'Apr', 'May']
sales = [200, 300, 250, 400, 450]

plt.plot(months, sales, marker='o', color='blue')
plt.title("Monthly Sales Trend")
plt.xlabel("Month")
plt.ylabel("Sales ($)")
plt.show()
```

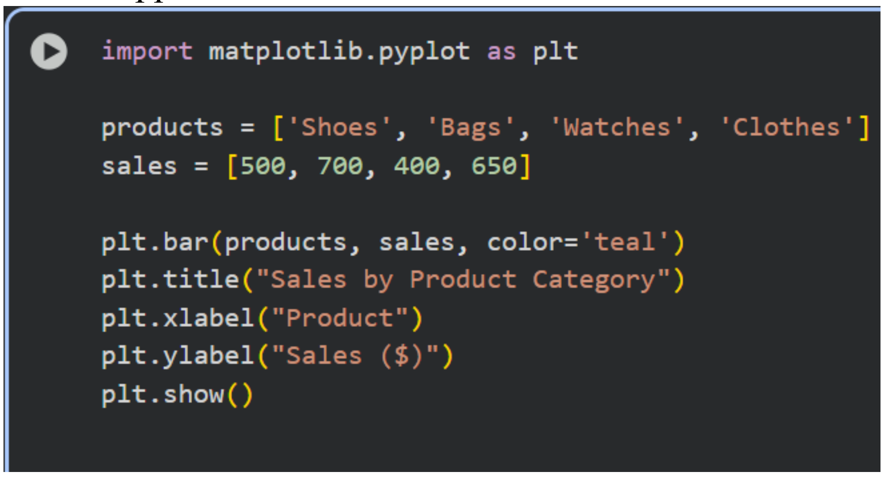
Output:



2.2. Bar Graph

Purpose: To compare product sales across different categories.

Code snippet:

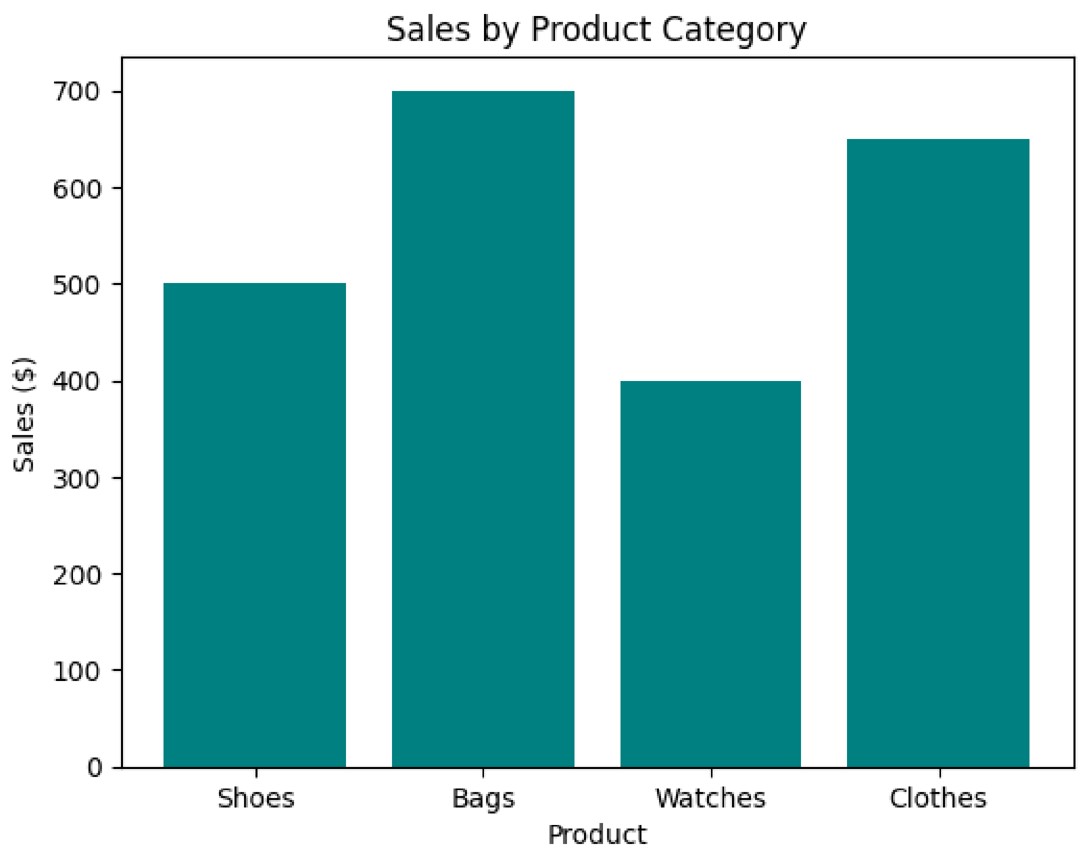


```
import matplotlib.pyplot as plt

products = ['Shoes', 'Bags', 'Watches', 'Clothes']
sales = [500, 700, 400, 650]

plt.bar(products, sales, color='teal')
plt.title("Sales by Product Category")
plt.xlabel("Product")
plt.ylabel("Sales ($)")
plt.show()
```

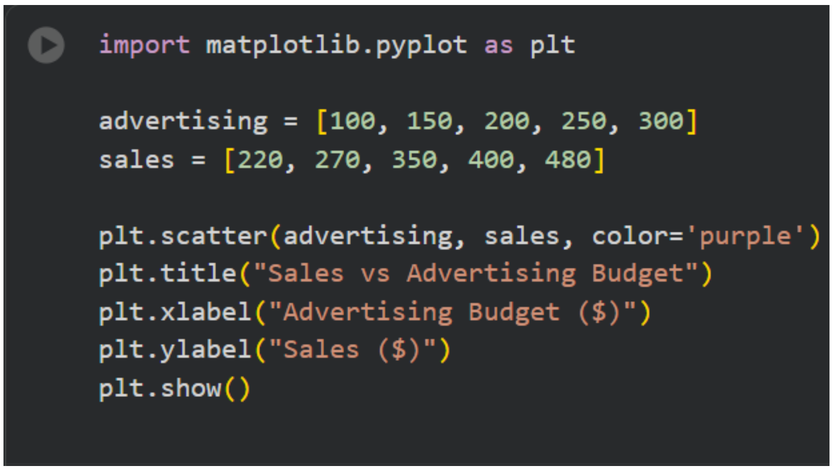
Output:



2.3. Scatter Plot

Purpose: To visualize the relationship between sales and advertising budget.

Code snippet:

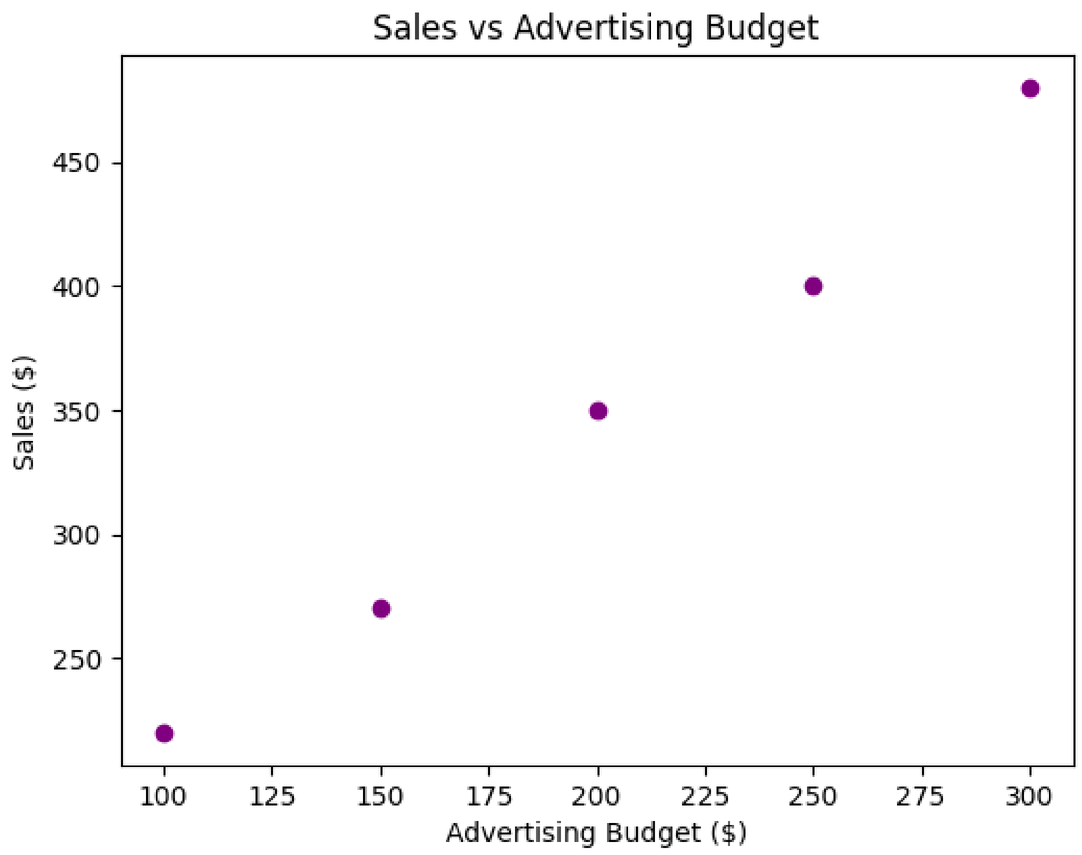


```
import matplotlib.pyplot as plt

advertising = [100, 150, 200, 250, 300]
sales = [220, 270, 350, 400, 480]

plt.scatter(advertising, sales, color='purple')
plt.title("Sales vs Advertising Budget")
plt.xlabel("Advertising Budget ($)")
plt.ylabel("Sales ($)")
plt.show()
```

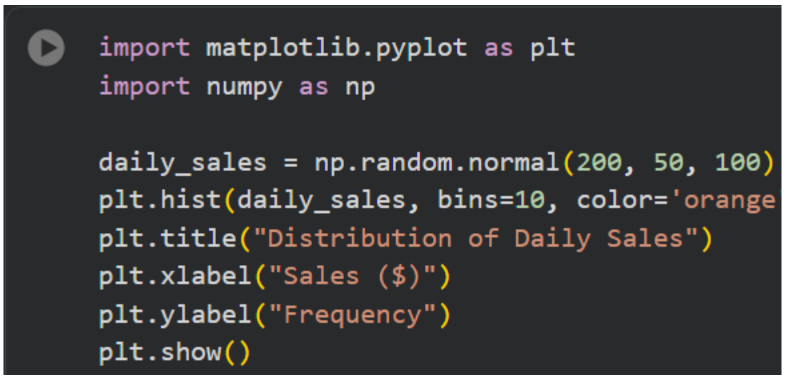
Output:



2.4. Histogram

Purpose: To analyze the frequency distribution of daily sales.

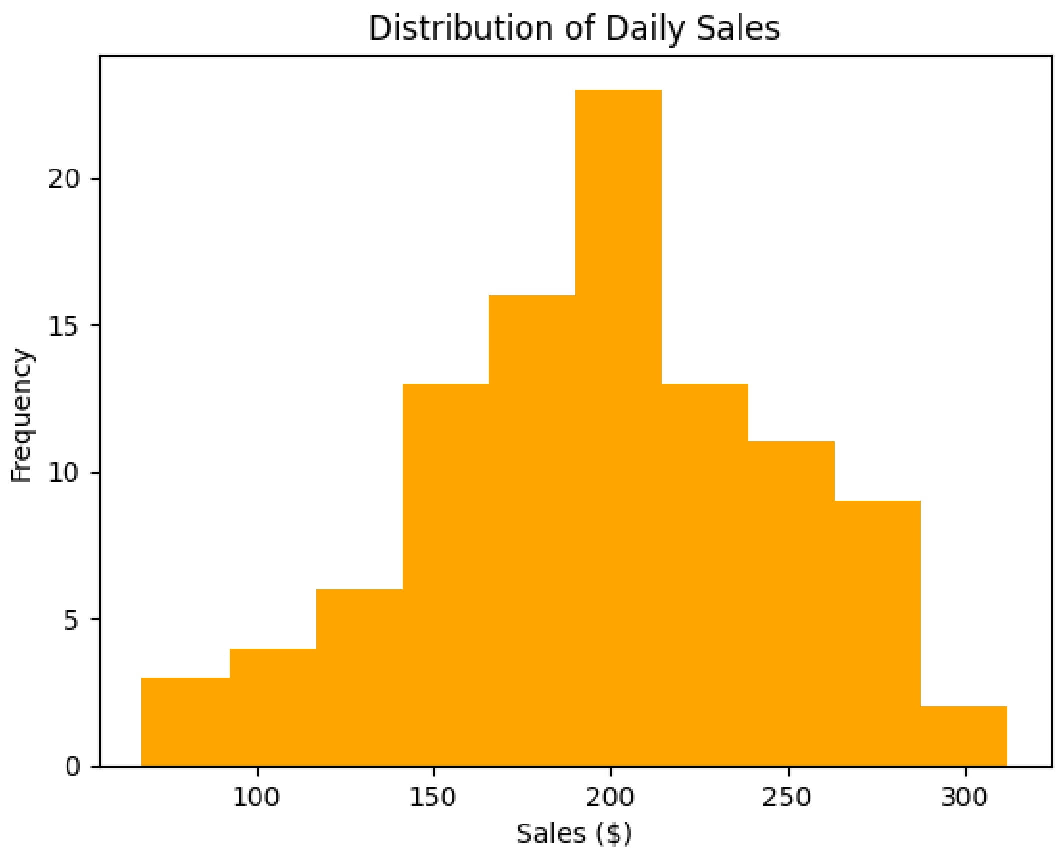
Code snippet:



```
import matplotlib.pyplot as plt
import numpy as np

daily_sales = np.random.normal(200, 50, 100)
plt.hist(daily_sales, bins=10, color='orange
plt.title("Distribution of Daily Sales")
plt.xlabel("Sales ($)")
plt.ylabel("Frequency")
plt.show()
```

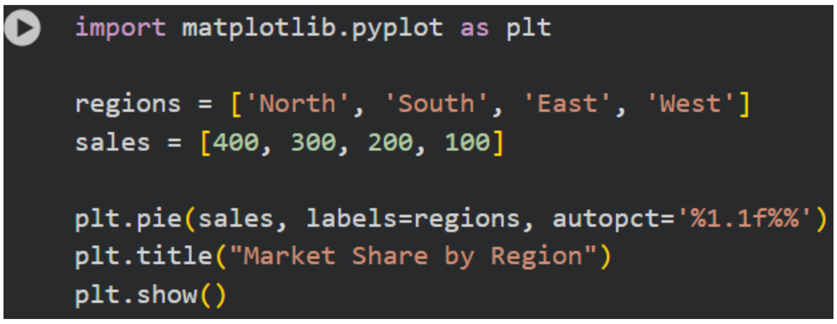
Output:



2.5. Pie Chart

Purpose: To show market share percentage by region.

Code snippet:

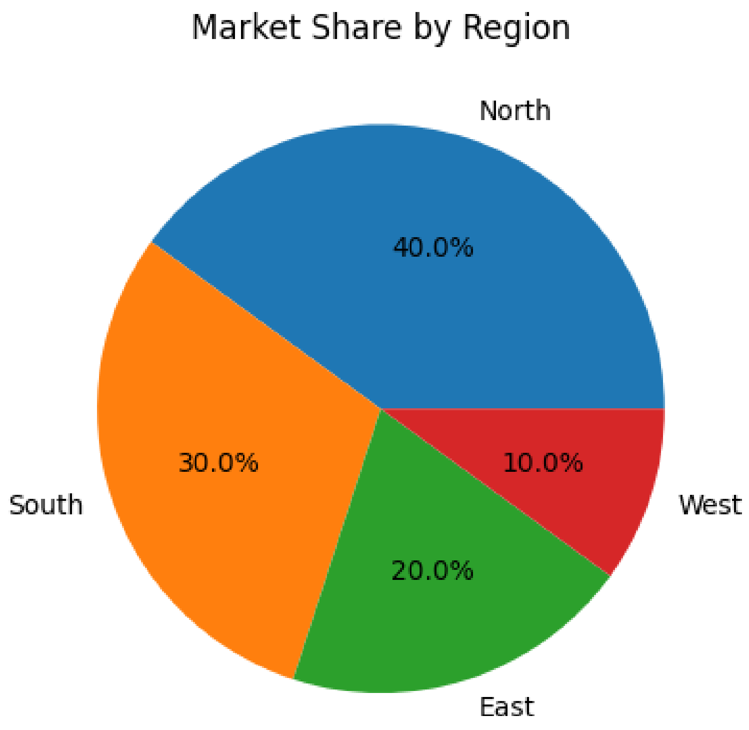


```
import matplotlib.pyplot as plt

regions = ['North', 'South', 'East', 'West']
sales = [400, 300, 200, 100]

plt.pie(sales, labels=regions, autopct='%1.1f%%')
plt.title("Market Share by Region")
plt.show()
```

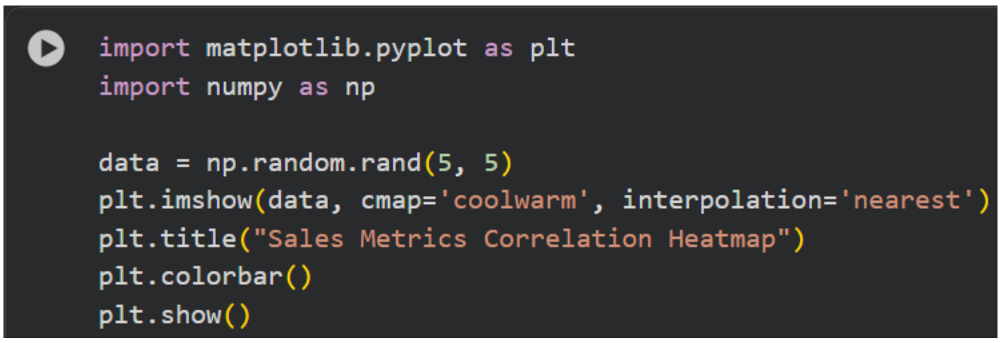
Output:



2.6. Heat Map

Purpose: To visualize correlation among multiple sales metrics.

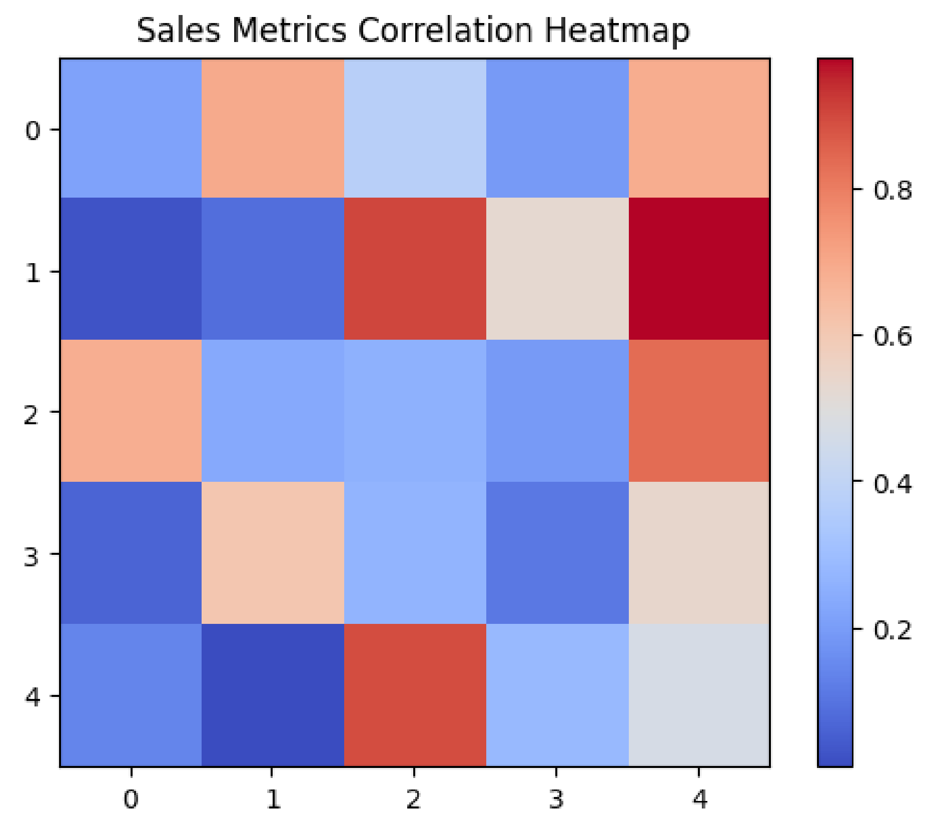
Code snippet:



```
import matplotlib.pyplot as plt
import numpy as np

data = np.random.rand(5, 5)
plt.imshow(data, cmap='coolwarm', interpolation='nearest')
plt.title("Sales Metrics Correlation Heatmap")
plt.colorbar()
plt.show()
```

Output:



3. Plotly

Overview

Plotly is a high-level Python library that creates interactive and web-based visualizations.
It allows users to create dashboards and presentations where you can hover, zoom, and explore data easily.
Plotly is great for storytelling and sharing insights online.

Key Features

- Built for interactivity — hover, zoom, and export easily.
- Supports 2D, 3D, and map visualizations.
- Works well with Pandas DataFrames.
- Can be combined with Dash for dashboard applications.

Some Graph Types and Examples in Plotly :

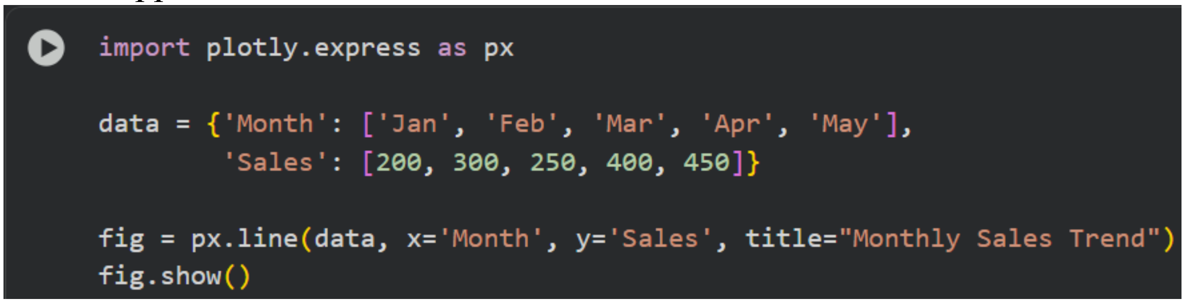
Use Case: ***Retail Sales Analysis***

We'll use the same retail sales data to create interactive versions of Matplotlib's charts.

3.1. Line Graph

Purpose: To show the trend of monthly sales over time.

Code snippet:

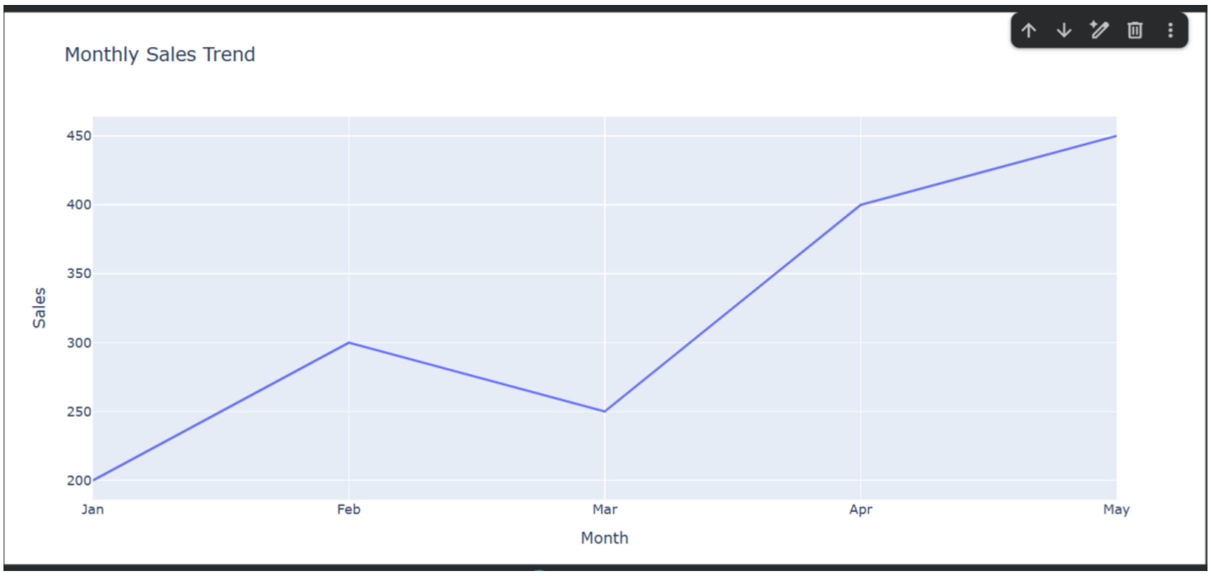


```python
import plotly.express as px

data = {'Month': ['Jan', 'Feb', 'Mar', 'Apr', 'May'],
        'Sales': [200, 300, 250, 400, 450]}

fig = px.line(data, x='Month', y='Sales', title="Monthly Sales Trend")
fig.show()
```

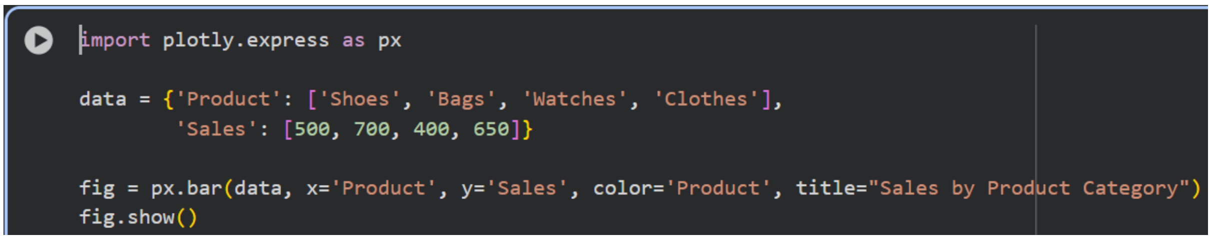
Output:



3.2. Bar Graph

Purpose: To compare product sales across different categories.

Code snippet:

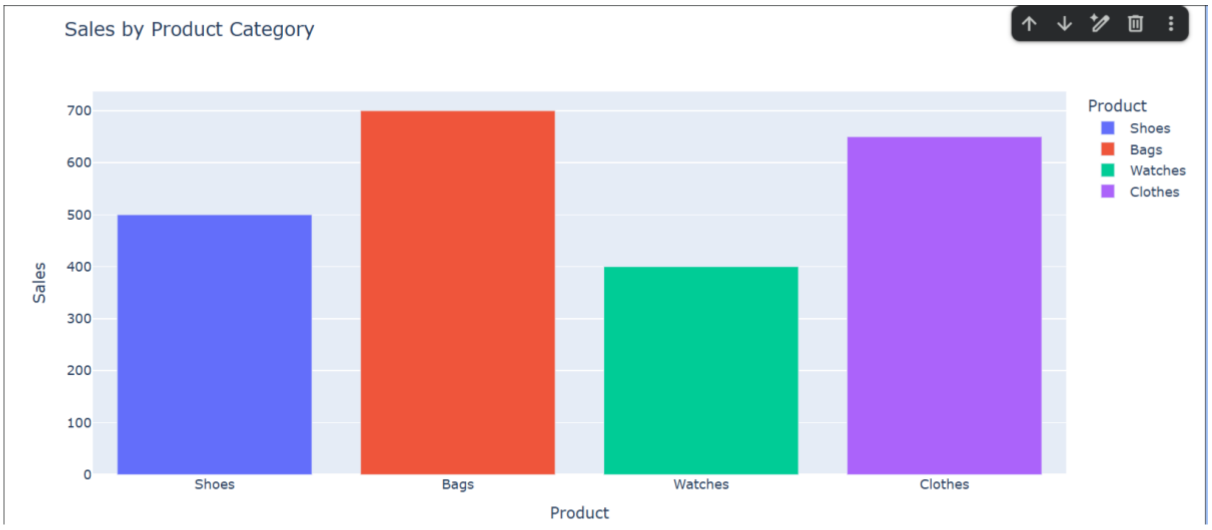


```
import plotly.express as px

data = {'Product': ['Shoes', 'Bags', 'Watches', 'Clothes'],
        'Sales': [500, 700, 400, 650]}

fig = px.bar(data, x='Product', y='Sales', color='Product', title="Sales by Product Category")
fig.show()
```

Output:



3.3. Scatter Plot

Purpose: To show the relationship between advertising and sales.

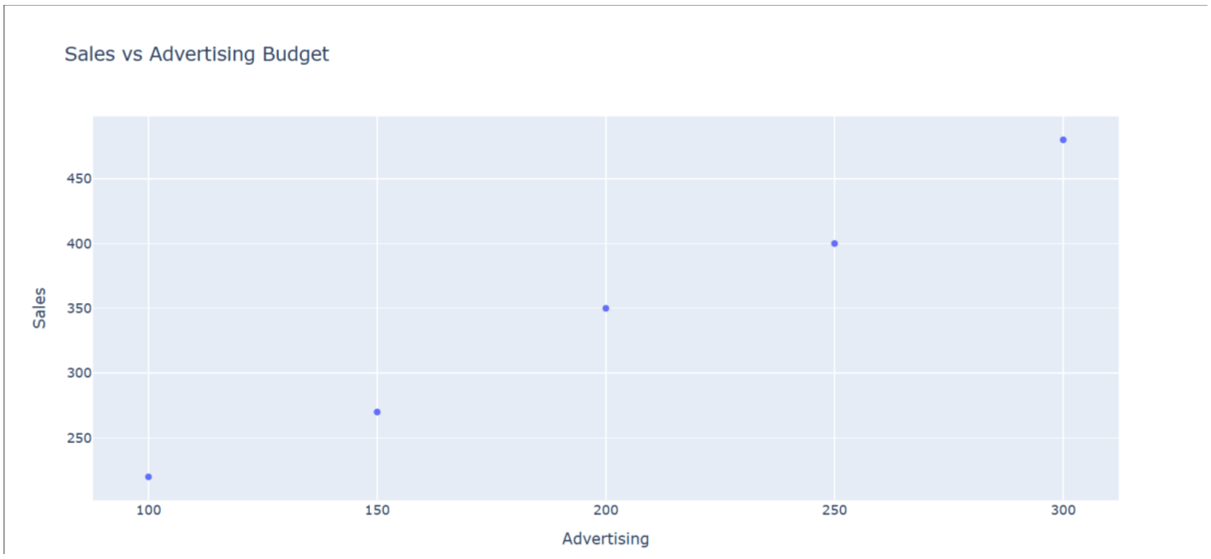
Code snippet:

```python
import plotly.express as px

data = {'Advertising': [100, 150, 200, 250, 300],
        'Sales': [220, 270, 350, 400, 480]}

fig = px.scatter(data, x='Advertising', y='Sales', title="Sales vs Advertising Budget")
fig.show()
```

Output:



3.4. Histogram

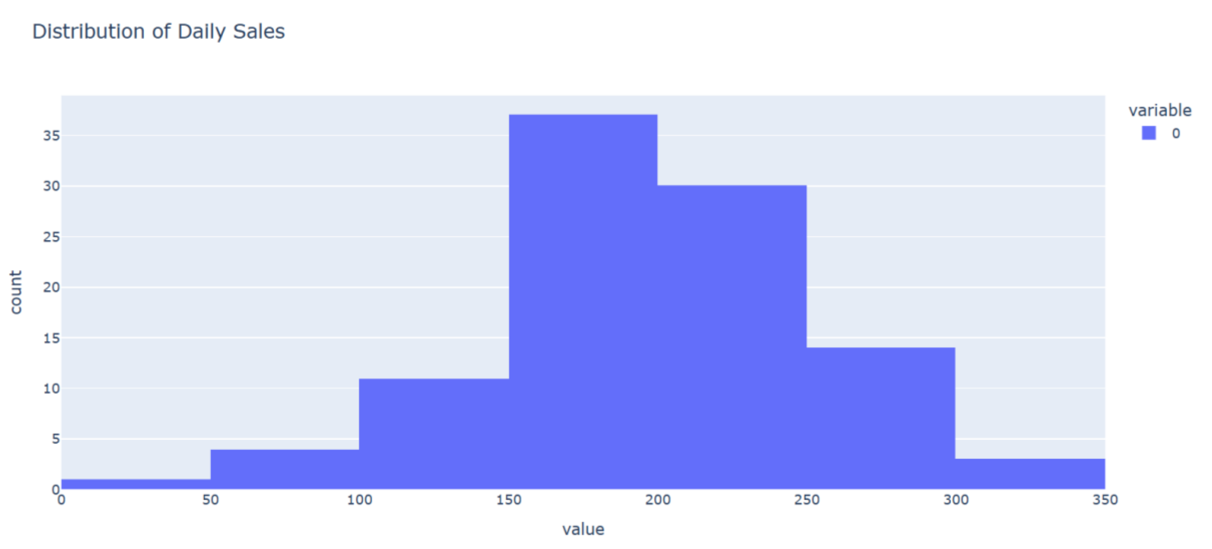
Purpose: To show the distribution of daily sales.

Code snippet:

```python
import plotly.express as px
import numpy as np

daily_sales = np.random.normal(200, 50, 100)
fig = px.histogram(daily_sales, nbins=10, title="Distribution of Daily Sales")
fig.show()
```

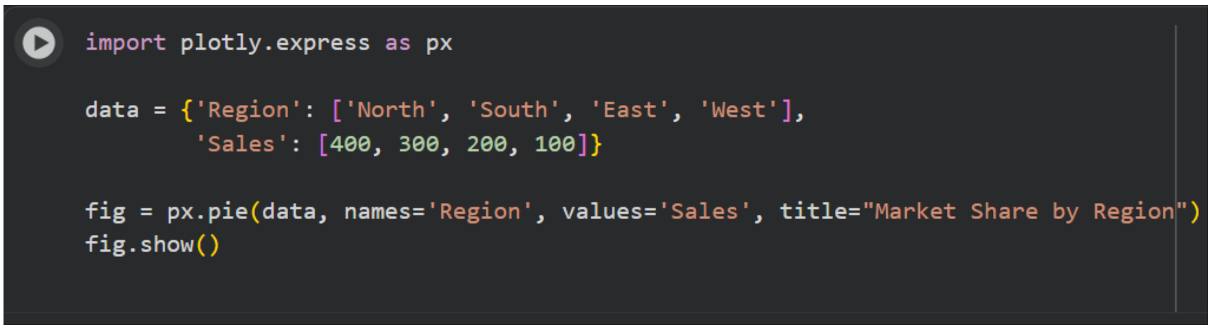
Output:



3.5. Pie Chart

Purpose: To display market share percentage by region.

Code snippet:

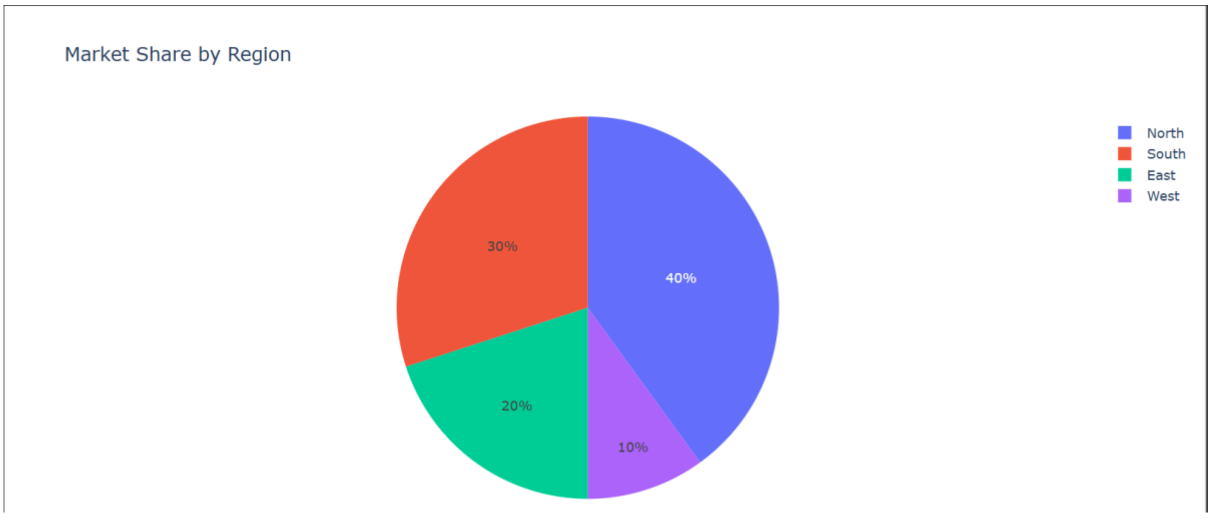


```
import plotly.express as px

data = {'Region': ['North', 'South', 'East', 'West'],
        'Sales': [400, 300, 200, 100]}

fig = px.pie(data, names='Region', values='Sales', title="Market Share by Region")
fig.show()
```

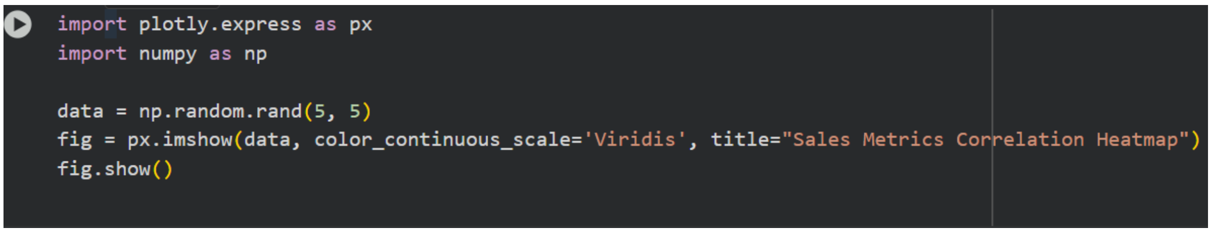
Output:



3.6. Heatmap

Purpose: To visualize correlation among multiple sales metrics.

Code snippet:

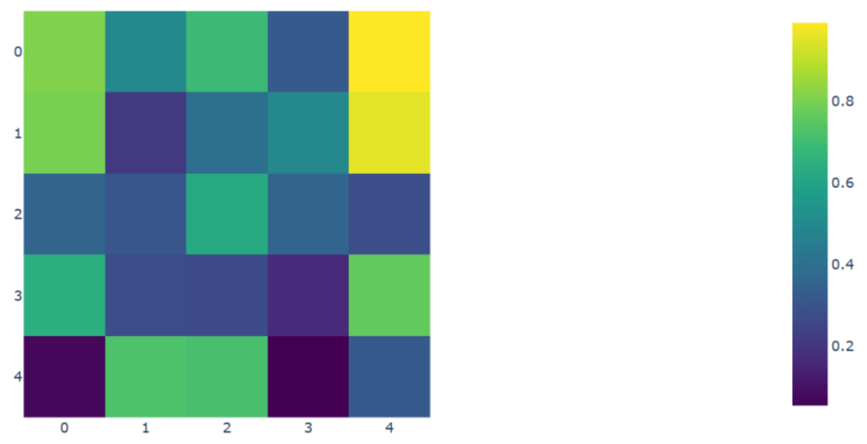


```
import plotly.express as px
import numpy as np

data = np.random.rand(5, 5)
fig = px.imshow(data, color_continuous_scale='Viridis', title="Sales Metrics Correlation Heatmap")
fig.show()
```

Output:

Sales Metrics Correlation Heatmap



4. Comparison Between Matplotlib and Plotly

Feature	**Matplotlib**	**Plotly**
Ease of Use	Requires more setup but offers detailed control.	Very simple syntax for quick plotting.
Customization	Fully customizable visuals.	Customization through parameters, limited deep control.
Interactivity	Static by default.	Interactive by default (hover, zoom).
Performance	Excellent for small and medium datasets.	Great for large and web-based dashboards.
Integration	Works with NumPy and Pandas.	Integrates with Dash and Pandas easily.

5.Conclusion

Both **Matplotlib** and **Plotly** are powerful tools for visualizing data.
Matplotlib is perfect for **static, detailed, publication-ready graphs**, while Plotly is ideal for **interactive dashboards** and **data exploration**.
When analyzing retail sales, use **Matplotlib** for reports and **Plotly** for live, dynamic presentations.