Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Practical based on Data Visualization with Plotly	
Experiment No: 24	**Date:**	**Enrollment No: 92400133055**

GITHUB

Aim: Practical based on Data Visualization with Plotly
IDE: Visual Studio Code
Installation pip install
plotly pandas Creating a
Sample Dataset import
pandas as pd import
plotly.express as px
Creating a Sample Dataset

```
# Sample data
data = {
   'Product': ['A', 'B', 'C', 'D', 'E'],
   'Sales': [100, 200, 150, 300, 250],
   'Profit': [30, 70, 50, 120, 90]
}
```

df = pd.DataFrame(data) Creating
Basic Visualizations
Bar Chart
Bar chart for Sales
A bar chart is great for comparing quantities across categories.

```
fig = px.bar(df, x='Product', y='Sales', title='Sales by Product')
```

fig.show() Output:

Line Chart
A line chart can help visualize trends over time or categories.
Line chart for Profit

```
fig = px.line(df, x='Product', y='Profit', title='Profit by Product')
```

fig.show() Output:

Scatter Plot
A scatter plot is useful for examining the relationship between two numerical variables.

	Marwadi University **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Practical based on Data Visualization with Plotly	
Experiment No: 24	**Date:**	**Enrollment No: 92400133055**

```
# Scatter plot for Sales vs. Profit
fig = px.scatter(df, x='Sales', y='Profit', color='Product', title='Sales vs. Profit')
fig.show() Output:
```


Customizing Visualizations

Plotly allows for extensive customization. Let's enhance our bar chart with more features.

```
# Enhanced Bar chart
fig = px.bar(df, x='Product', y='Sales',
title='Sales by Product',              color='Profit',
# Color by Profit
          text='Sales')    # Show sales value on bars

# Customize layout
fig.update_layout(xaxis_title='Product',
yaxis_title='Sales',                  legend_title='Profit',
               template='plotly_dark')  # Change template

fig.show()
```

Exporting Visualizations

Plotly figures as static images or HTML files.

```
# Save the figure as an HTML file
fig.write_html('sales_by_product.html')

# Save the figure as a PNG file (you may need to install kaleido)
fig.write_image('sales_by_product.png') Output:
```