

 Marwadi University Marwadi Chandrana 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: 92400133181

Aim: Practical based on Data Visualization with Plotly

IDE:

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()
```

 Marwadi University Marwadi Chandrana 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: 92400133181

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()
```

Scatter Plot

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

```
# Scatter plot for Sales vs. Profit
```

```
fig = px.scatter(df, x='Sales', y='Profit', color='Product', title='Sales vs. Profit')
```

```
fig.show()
```

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
```

 Marwadi University Marwadi Chandrana 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: 92400133181

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')
```

Code:



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: 92400133181

```
1 import pandas as pd
2 import plotly.express as px
3 # Sample data
4 data = {
5     'Product': ['A', 'B', 'C', 'D', 'E'],
6     'Sales': [100, 200, 150, 300, 250],
7     'Profit': [30, 70, 50, 120, 90]
8 }
9 df = pd.DataFrame(data)
10 # Bar chart for Sales
11 fig = px.bar(df, x='Product', y='Sales', title='Sales by Product')
12 fig.show()
13
14 fig = px.scatter(df, x='Sales', y='Profit', color='Product', title='Sales vs. Profit')
15 fig.show()
16
17 # Enhanced Bar chart
18 fig = px.bar(df, x='Product', y='Sales',
19             title='Sales by Product',
20             color='Profit', # Color by Profit
21             text='Sales')    # Show sales value on bars
22
23 # Customize layout
24 fig.update_layout(xaxis_title='Product',
25                     yaxis_title='Sales',
26                     legend_title='Profit',
27                     template='plotly_dark') # Change template
28 fig.show()
29
30 # Save the figure as an HTML file
31 fig.write_html('sales_by_product.html')
32
33 # Save the figure as a PNG file (you may need to install kaleido)
34 fig.write_image('sales_by_product.png')
35
```

Output:



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: 92400133181

The screenshot shows a Jupyter Notebook environment. At the top, there's a file browser window titled 'Files' with columns for 'Name' and 'Date Modified'. The browser lists several files: 'Reverse of a number PWP Lab 6.7' (modified 8/9/2025), 'Reversing of signal PWP Lab 12.py' (modified 9/16/2025), 'sales_by_product.html' (modified 11/22/2025), 'sales_by_product.png' (modified 11/22/2025), and 'Scaling of signals PWP Lab 12.py' (modified 9/16/2025). Below the browser are tabs for 'Help', 'Variable Explorer', 'Debugger', 'Profiler', 'Plots', and 'Files', with 'Files' being the active tab. The main area contains a 'Console 1/A' tab with the command 'In [2]: %runfile C:/Users/devah/Documents/PWP/untitled0.py --wdir' entered.



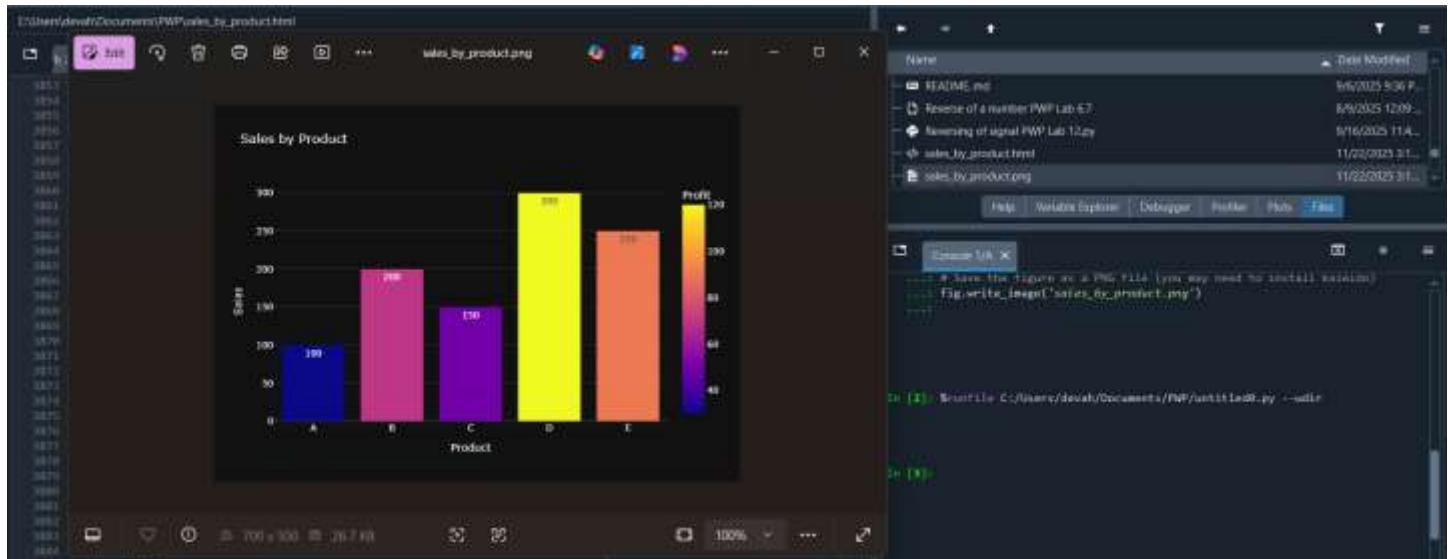
Subject: Programming With Python (01CT1309)

Aim: Practical based on Data Visualization with Plotly

Experiment No: 24

Date:

Enrollment No: 92400133181



GitHub:

<https://github.com/mallaadisrinivasu132035-code/python.git>