In [2]:

```
import gradio as gr
import pandas as pd
import matplotlib.pyplot as plt

# Sample data
data = {
    "Month": ["Jan","Feb","Mar","Apr","May","Jun"],
    "Sales": [10000,12000,15000,13000,17000,16000],
    "Profit": [2000,3000,4000,2500,3500,3000]
}
df = pd.DataFrame(data)

# Function to return selected plot
def generate_plot(plot_type):
    fig = plt.figure(figsize=(8,5))

    if plot_type == "Line Plot":
        plt.plot(df['Month'], df['Sales'],color='black',marker='o',linestyle='--',l
        plt.title('Sales Trends Over Months')
        plt.xlabel('Month')
        plt.ylabel('Sales ($)')
        plt.grid(True)
        plt.legend()


    elif plot_type == "Stacked Bar Chart":
        fig.set_size_inches(10, 6)
        width = 0.3
        plt.bar(df['Month'], df['Sales'], width=width, color='black', label='Sales'
        plt.bar(df['Month'], df['Profit'], width=width, color='green', label='Profi
        plt.title('Sales and Profit Comparison by Month')
        plt.xlabel('Month')
        plt.ylabel('Amount ($)')
        plt.legend()

    elif plot_type == "Pie Chart":
        fig.set_size_inches(7,7)
        plt.pie(df['Profit'], labels=df['Month'], autopct='%1.2f%%',startangle=140,
        plt.title('Profit Distribution by Month')

    elif plot_type == "Scatter Plot":
        plt.scatter(df['Sales'], df['Profit'], color='red', s=100, edgecolors='blac
        plt.title('Sales vs Profit Scatter Plot')
        plt.xlabel('Sales')
        plt.ylabel('Profit')
        plt.grid(True)

    elif plot_type == "Histogram":
        plt.hist(df['Sales'], bins=5,color='yellow',edgecolor='black'),
        plt.title('Sales Distribution')
        plt.xlabel('Sales')
        plt.ylabel('Frequency')

    elif plot_type == "Box Plot":
        plt.boxplot(df['Profit'], vert=False, patch_artist=True, boxprops=dict(face
        plt.title('Sales Box Plot')
        plt.xlabel('Profit ($)')

    plt.tight_layout()
    return fig

# Gradio UI
demo = gr.Interface(
    fn=generate_plot,
    inputs=gr.Radio(
        ["Line Plot", "Stacked Bar Chart", "Pie Chart", "Scatter Plot", "Histogram"
        label="Choose Plot Type"
    ),
    outputs=gr.Plot(label="Visualization"),
    title="Sales & Profit Visual Explorer",
    description="Choose a chart type to visualize the data."
)

demo.launch()
```

```
* Running on local URL:  http://127.0.0.1:7869
* To create a public link, set `share=True` in `launch()`.
```

Out[2]:

In [ ]: