```
Start coding or generate with AI.
```

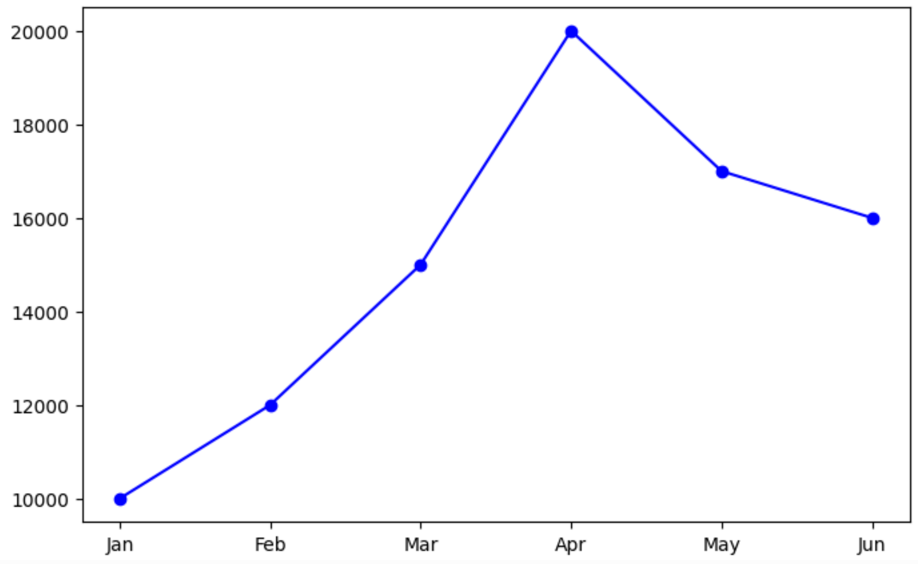
```
import pandas as pd
data={
    "Month":['Jan','Feb','Mar','Apr','May','Jun'],
    "Sales":[10000,12000,15000,20000,17000,16000],
    "Profit":[2000,1500,600,3000,3500,2500]
}
df = pd.DataFrame(data)
print(df)
```

```
  Month  Sales  Profit
0   Jan  10000    2000
1   Feb  12000    1500
2   Mar  15000     600
3   Apr  20000    3000
4   May  17000    3500
5   Jun  16000    2500
```

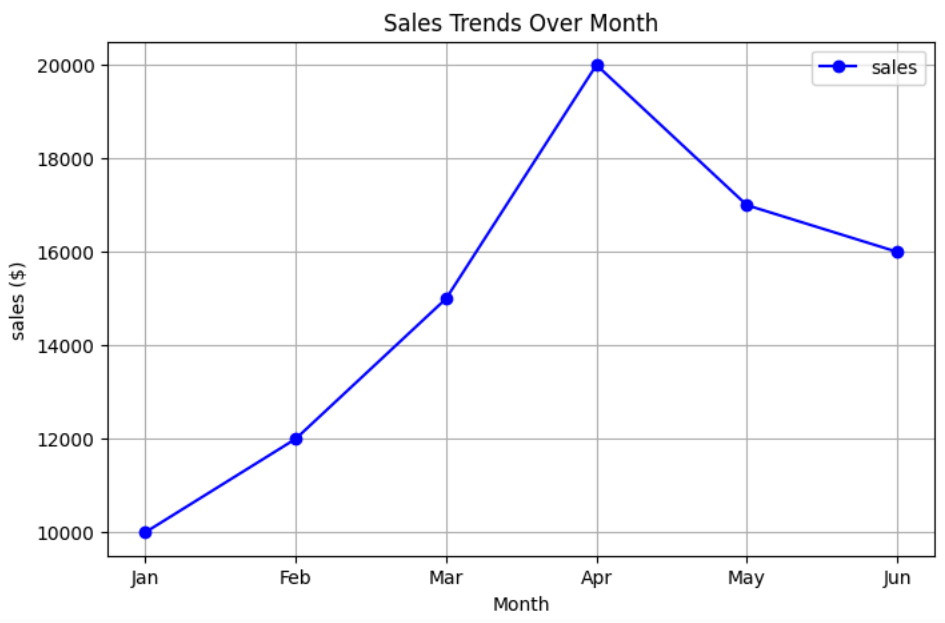
LINE PLOT MONTHLY SALES

```
import matplotlib.pyplot as plt
plt.figure(figsize=(8,5))
plt.plot(df['Month'],df['Sales'],color='blue',marker='o',linestyle='-',label='sales')
```

```
[<matplotlib.lines.Line2D at 0x7eb198fc9490>]
```

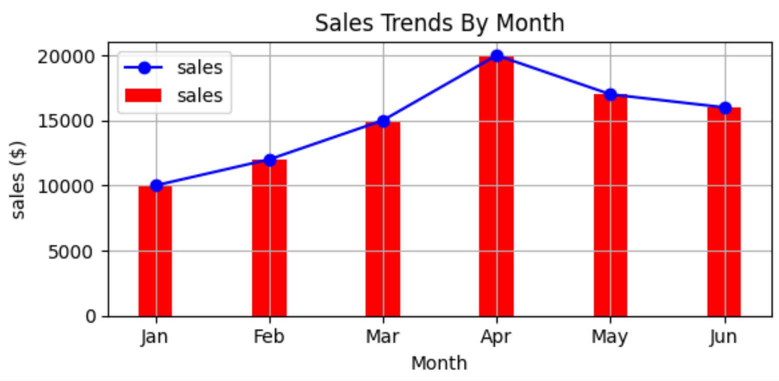


```
import matplotlib.pyplot as plt
plt.figure(figsize=(8,5))
plt.plot(df['Month'],df['Sales'],color='blue',marker='o',linestyle='-',label='sales')
plt.title('Sales Trends Over Month')
plt.xlabel('Month')
plt.ylabel('sales ($)')
plt.grid(True)
plt.legend()
plt.show()
```

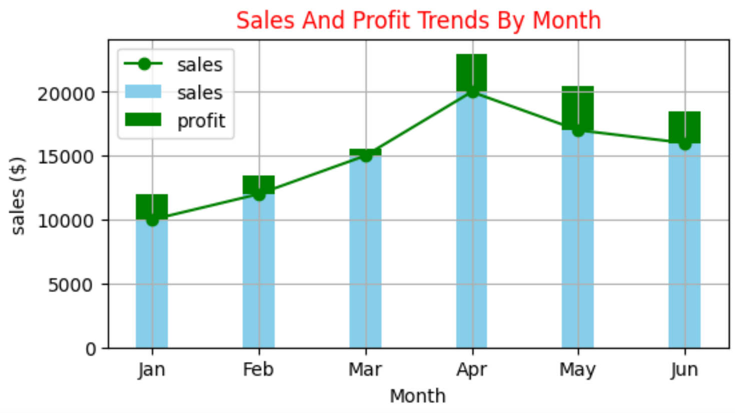


BARPLOT MONTH VS PROFIT

```
import matplotlib.pyplot as plt
plt.figure(figsize=(6,3))
width=0.3
plt.bar(df['Month'],df['Sales'], width=width,color='red',label='sales')
plt.plot(df['Month'],df['Sales'],color='blue',marker='o',linestyle='-',label='sales')
plt.title('Sales Trends By Month')
plt.xlabel('Month')
plt.ylabel('sales ($)')
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```



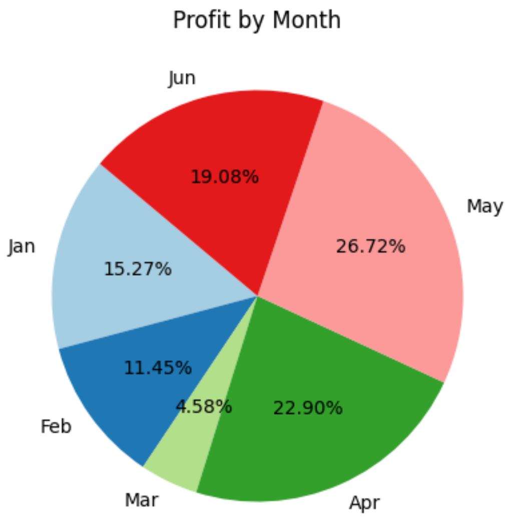
```
import matplotlib.pyplot as plt
plt.figure(figsize=(6,3))
width=0.3
plt.bar(df['Month'],df['Sales'], width=width,color='skyblue',label='sales')
plt.bar(df['Month'],df['Profit'], width=width,color='green',label='profit',bottom=df['Sales'])
plt.plot(df['Month'],df['Sales'],color='green',marker='o',linestyle='-',label='sales')
plt.title('Sales And Profit Trends By Month',color='red')
plt.xlabel('Month')
plt.ylabel('sales ($)')
plt.grid(True)
plt.legend()
plt.show()
```



PIE CHART PROFIT VS MONTH

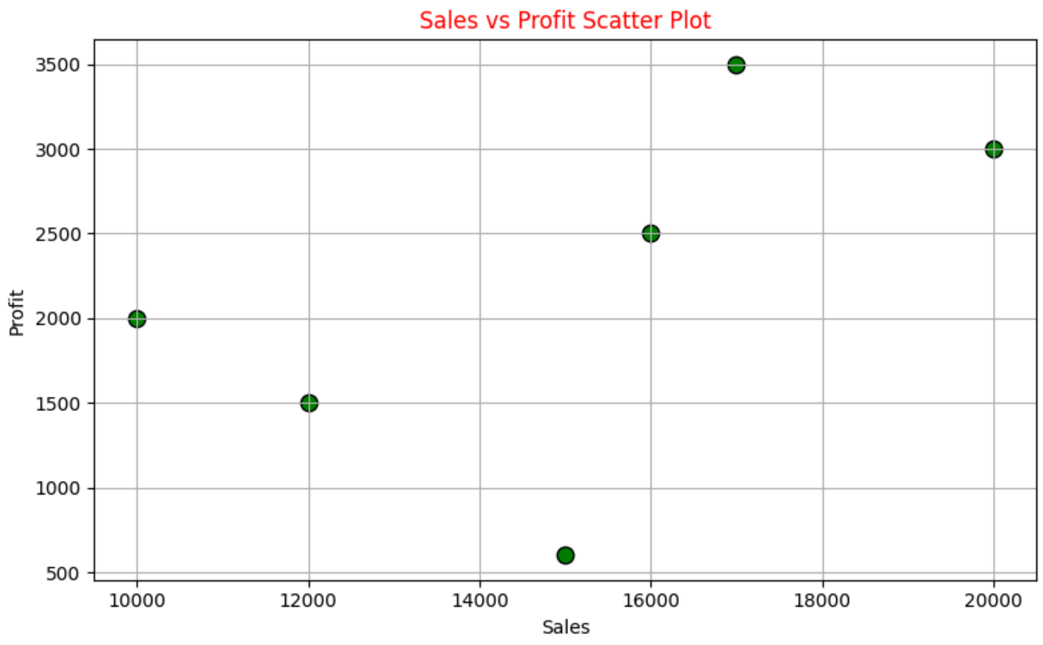
```
from enum import auto
plt.figure(figsize=(10,5))
plt.pie(df['Profit'],labels=df['Month'],autopct='%1.2f%%',startangle=140,colors=plt.cm.Paired.colors)
plt.title('Profit by Month')
```

```
Text(0.5, 1.0, 'Profit by Month')
```



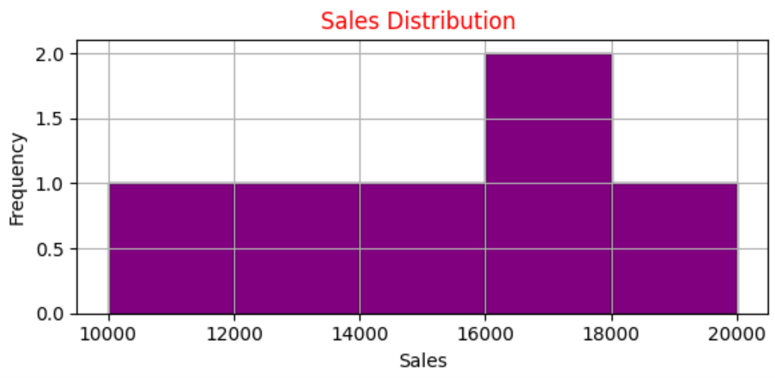
SCATTER **PLOT**

```
plt.figure(figsize=(8,5))
plt.scatter(df['Sales'],df['Profit'],color='green',s=80,edgecolors='black')
plt.title('Sales vs Profit Scatter Plot',color='red')
plt.xlabel('Sales')
plt.ylabel('Profit')
plt.grid(True)
plt.tight_layout()
plt.show()
```



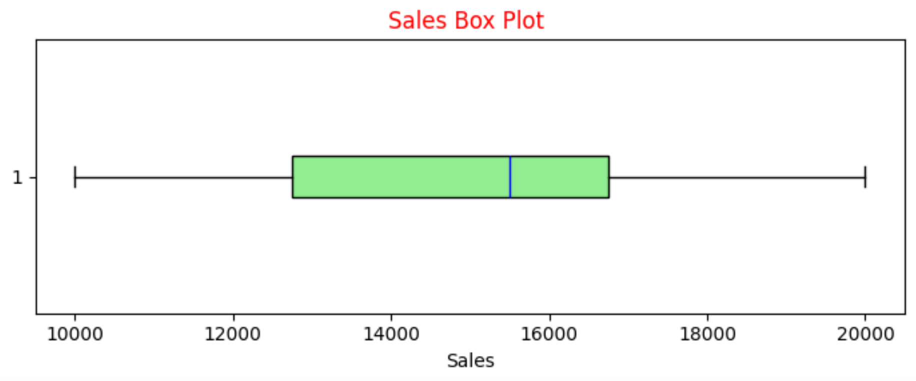
HISTOGRAM

```
plt.figure(figsize=(6,3))
plt.hist(df['Sales'],bins=5,color='purple',edgecolor='black')
plt.title('Sales Distribution ',color='red')
plt.xlabel('Sales')
plt.ylabel('Frequency')
plt.grid(True)
plt.tight_layout()
plt.show()
```

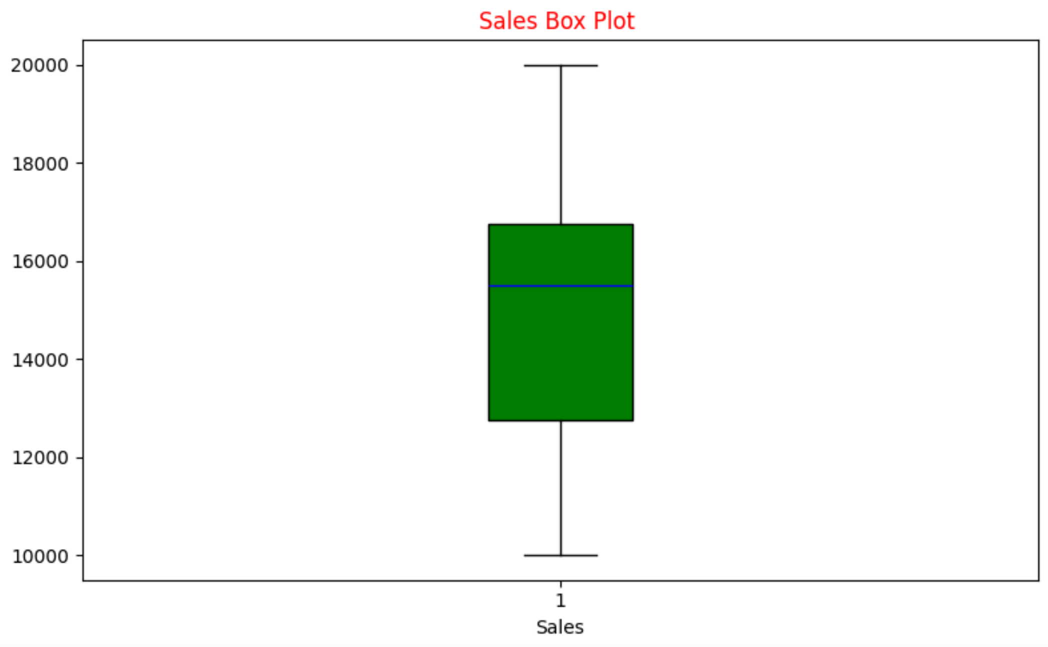


BOX **PLOT**

```
plt.figure(figsize=(7,3))
plt.boxplot(df['Sales'],vert=False,patch_artist=True,boxprops=dict(facecolor='lightgreen'),medianprops=dict(color='blue')),
plt.title('Sales Box Plot ',color='red')
plt.xlabel('Sales')
plt.tight_layout()
plt.show()
```



```
plt.figure(figsize=(8,5))
plt.boxplot(df['Sales'],patch_artist=True,boxprops=dict(facecolor='green'),medianprops=dict(color='blue'))
plt.title('Sales Box Plot ',color='red')
plt.xlabel('Sales')
#plt.grid(True)
plt.tight_layout()
plt.show()
```



```
#!pip install gradio
```

```
Requirement already satisfied: gradio in /usr/local/lib/python3.11/dist-packages (5.29.0)
Requirement already satisfied: aiofiles<25.0,>=22.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (24.1.0)
Requirement already satisfied: anyio<5.0,>=3.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (4.9.0)
Requirement already satisfied: fastapi<1.0,>=0.115.2 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.115.12)
Requirement already satisfied: ffmpy in /usr/local/lib/python3.11/dist-packages (from gradio) (0.5.0)
Requirement already satisfied: gradio-client==1.10.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (1.10.0)
Requirement already satisfied: groovy~=0.1 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.1.2)
Requirement already satisfied: httpx>=0.24.1 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.28.1)
Requirement already satisfied: huggingface-hub>=0.28.1 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.31.1)
Requirement already satisfied: jinja2<4.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (3.1.6)
Requirement already satisfied: markupsafe<4.0,>=2.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (3.0.2)
Requirement already satisfied: numpy<3.0,>=1.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (2.0.2)
Requirement already satisfied: orjson~=3.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (3.10.18)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from gradio) (24.2)
Requirement already satisfied: pandas<3.0,>=1.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (2.2.2)
Requirement already satisfied: pillow<12.0,>=8.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (11.2.1)
Requirement already satisfied: pydantic<2.12,>=2.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (2.11.4)
Requirement already satisfied: pydub in /usr/local/lib/python3.11/dist-packages (from gradio) (0.25.1)
Requirement already satisfied: python-multipart>=0.0.18 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.0.20)
Requirement already satisfied: pyyaml<7.0,>=5.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (6.0.2)
Requirement already satisfied: ruff>=0.9.3 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.11.9)
Requirement already satisfied: safehttpx<0.2.0,>=0.1.6 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.1.6)
Requirement already satisfied: semantic-version~=2.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (2.10.0)
Requirement already satisfied: starlette<1.0,>=0.40.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.46.2)
Requirement already satisfied: tomlkit<0.14.0,>=0.12.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.13.2)
Requirement already satisfied: typer<1.0,>=0.12 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.15.3)
Requirement already satisfied: typing-extensions~=4.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (4.13.2)
Requirement already satisfied: uvicorn>=0.14.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.34.2)
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-packages (from gradio-client==1.10.0->gradio) (2025.3.2)
Requirement already satisfied: websockets<16.0,>=10.0 in /usr/local/lib/python3.11/dist-packages (from gradio-client==1.10.0->gradic
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.11/dist-packages (from anyio<5.0,>=3.0->gradio) (3.10)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.11/dist-packages (from anyio<5.0,>=3.0->gradio) (1.3.1)
Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-packages (from httpx>=0.24.1->gradio) (2025.4.26)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.11/dist-packages (from httpx>=0.24.1->gradio) (1.0.9)
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.11/dist-packages (from httpcore==1.*->httpx>=0.24.1->gradio) (0.1
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.28.1->gradio) (3.18.0)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.28.1->gradio) (2.32.3)
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.28.1->gradio) (4.67
Requirement already satisfied: hf-xet<2.0.0,>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.28.1->gradic
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (2
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (2025.2)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.11/dist-packages (from pydantic<2.12,>=2.0->gradio)
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/python3.11/dist-packages (from pydantic<2.12,>=2.0->gradio)
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from pydantic<2.12,>=2.0->gradic
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.11/dist-packages (from typer<1.0,>=0.12->gradio) (8.1.8)
Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3.11/dist-packages (from typer<1.0,>=0.12->gradio) (1.5.4
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.11/dist-packages (from typer<1.0,>=0.12->gradio) (13.9.4)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas<3.0,>=1.0->g
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/dist-packages (from rich>=10.11.0->typer<1.0,>=0.1
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from rich>=10.11.0->typer<1.0,>=0
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->huggingface-hub>=
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->huggingface-hub>=0.28.1
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0->rich>=10.11.0->typ
```

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

data = {
    "Month": ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun'],
    "Sales": [10000, 12000, 15000, 20000, 17000, 16000],
    "Profit": [2000, 1500, 600, 3000, 3500, 2500]
}
df = pd.DataFrame(data)

def generate_plot(plot_type):
    fig = plt.figure(figsize=(8, 5))

    if plot_type == 'Line Plot':
        plt.plot(df['Month'], df['Sales'], color='blue', marker='o', linestyle='-', label='sales')
        plt.title('Sales Trends Over Month')
        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='skyblue', label='sales')
        plt.bar(df['Month'], df['Profit'], width=width, color='green', label='profit', bottom=df['Sales'])
        plt.plot(df['Month'], df['Sales'], color='green', marker='o', linestyle='-', label='sales')
        plt.title('Sales And Profit Trends By Month', color='red')
        plt.xlabel('Month')
        plt.ylabel('Amount ($)')
        plt.grid(True)
        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, colors=plt.cm.Paired.colors)
        plt.title("Profit By Month")
    elif plot_type == 'Scatter Plot':
        plt.scatter(df['Sales'], df['Profit'], color='green', s=80, edgecolors='black')
        plt.title('Sales vs Profit Scatter Plot', color='red')
        plt.xlabel('Sales')
        plt.ylabel('Profit')
        plt.grid(True)
    elif plot_type == 'Histogram':
        plt.hist(df['Sales'], bins=5, color='purple', edgecolor='black')
        plt.title('Sales Distribution ', color='red')
        plt.xlabel('Sales')
        plt.ylabel('Frequency')

    elif plot_type == 'Box Plot':
        plt.boxplot(df['Sales'], patch_artist=True, boxprops=dict(facecolor='green'), medianprops=dict(color='blue'))
        plt.title('Sales Box Plot ', color='red')
        plt.xlabel('Sales')
        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', 'Box Plot'],
                    label="Choose plot type"),
    outputs=gr.Plot(label='Sales data visualization'),
    title='Sales and Profit Visual Insight',
    description="Choose the type to visualize the data"
)
```