

In [1]: `import pandas as pd`

```
# 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)
```

In [2]: `df`

Out[2]:

	Month	Sales	Profit
0	Jan	10000	2000
1	Feb	12000	3000
2	Mar	15000	4000
3	Apr	13000	2500
4	May	17000	3500
5	Jun	16000	3000

In [3]: *# 1- Line plot sales over time*

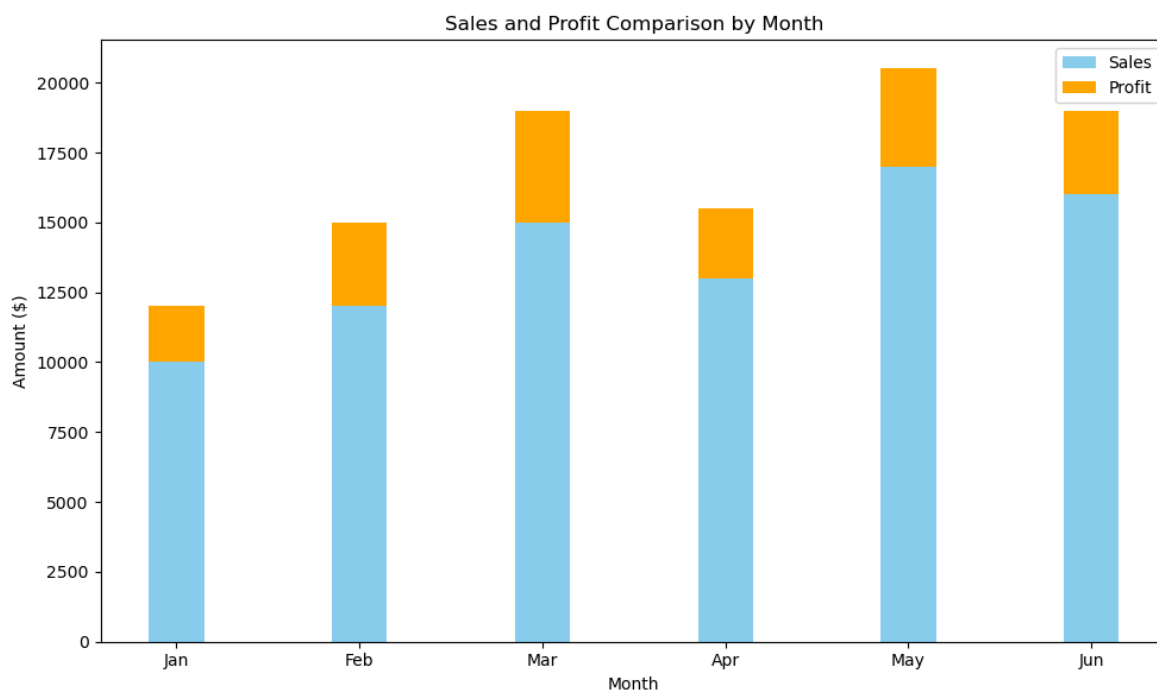
```
import matplotlib.pyplot as plt

# Line Plot: Sales over Time
plt.figure(figsize=(8, 5))
plt.plot(df['Month'], df['Sales'], color='blue', marker='o', linestyle='-', label='Sales')
plt.title('Sales Trend Over Months')
plt.xlabel('Month')
plt.ylabel('Sales ($)')
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```



```
In [4]: # Bar Plot: Sales vs Profit by Month
plt.figure(figsize=(10, 6))
width = 0.3
plt.bar(df['Month'], df['Sales'], width=width, label='Sales', color='skyblue')
plt.bar(df['Month'], df['Profit'], width=width, label='Profit', color='orange',
plt.title('Sales and Profit Comparison by Month')
plt.xlabel('Month')
plt.ylabel('Amount ($)')
plt.legend()
plt.tight_layout()
plt.show()

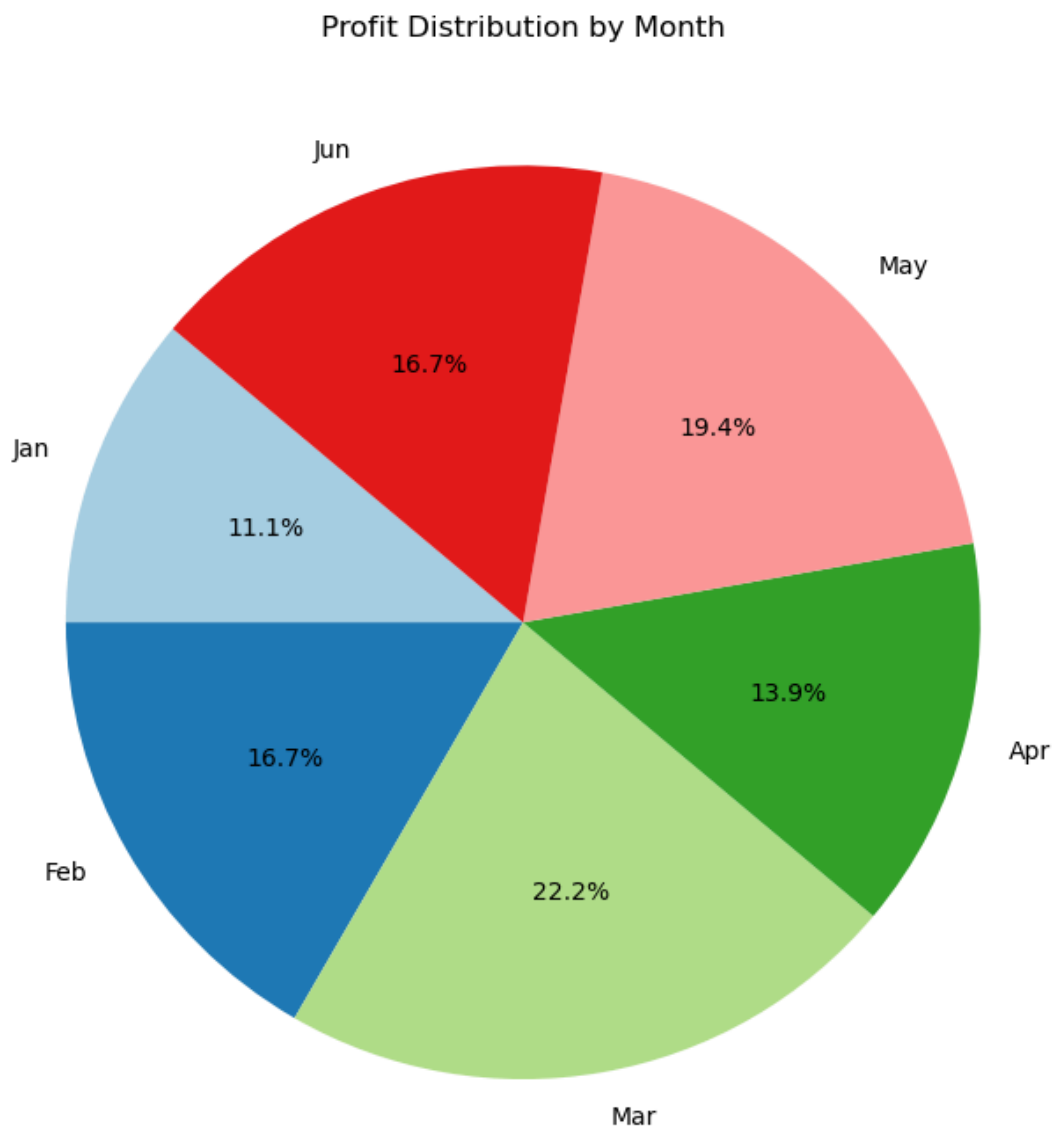
# A stacked bar plot where you can compare Sales and Profit for each month.
```



```
In [5]: # Pie Chart: Profit Distribution by Month
plt.figure(figsize=(7, 7))
```

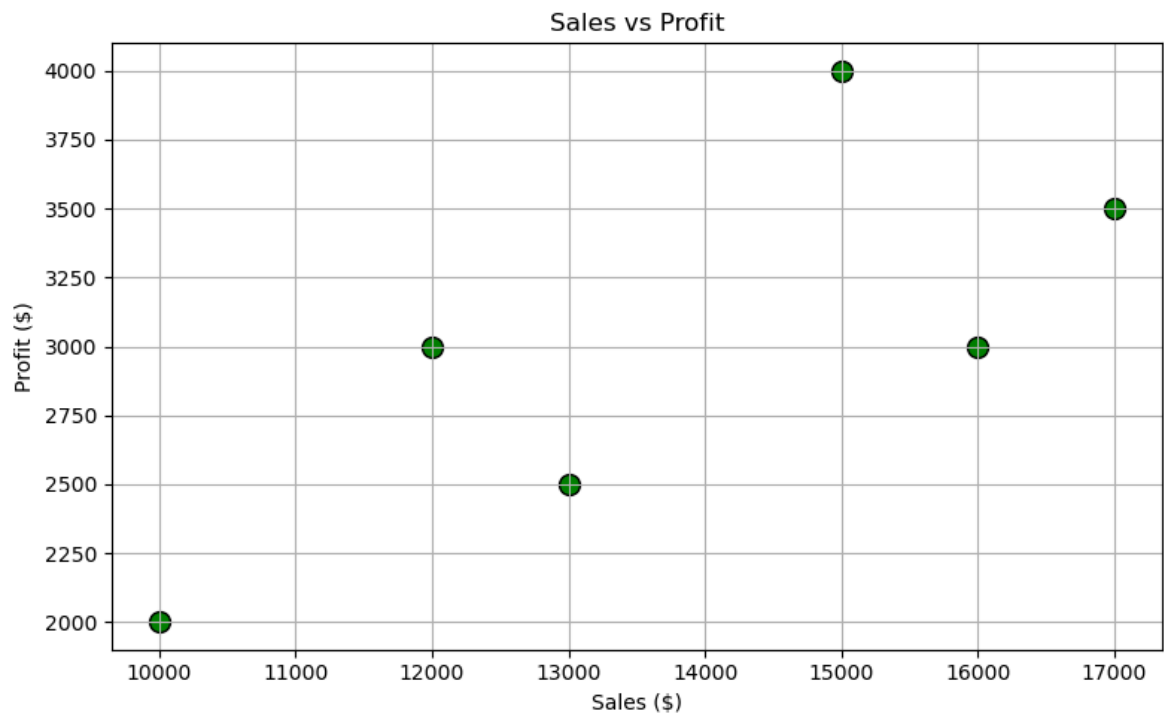
```
plt.pie(df['Profit'], labels=df['Month'], autopct='%1.1f%%', startangle=140, col
plt.title('Profit Distribution by Month')
plt.tight_layout()
plt.show()
```

A pie chart displaying the proportion of profit distribution for each month.



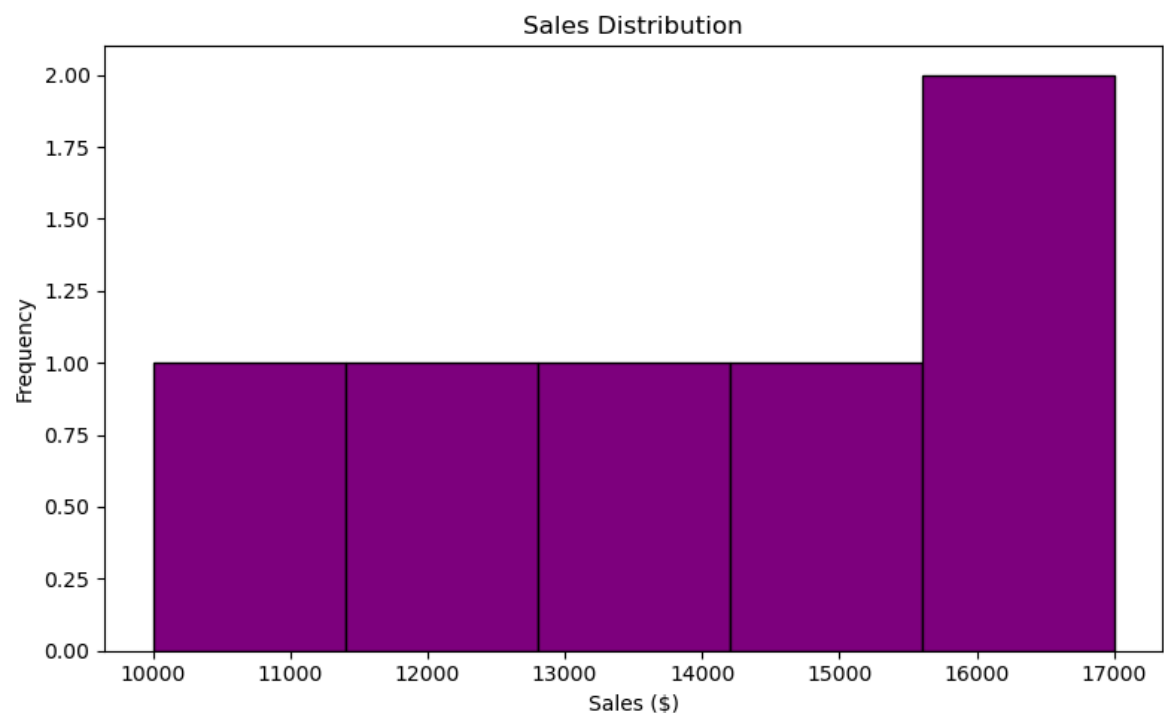
```
In [6]: # Scatter Plot: Sales vs Profit (Correlation)
plt.figure(figsize=(8, 5))
plt.scatter(df['Sales'], df['Profit'], color='green', s=100, edgecolors='black')
plt.title('Sales vs Profit')
plt.xlabel('Sales ($)')
plt.ylabel('Profit ($)')
plt.grid(True)
plt.tight_layout()
plt.show()
```

A scatter plot showing the correlation between Sales and Profit.



```
In [7]: # Histogram: Distribution of Sales
plt.figure(figsize=(8, 5))
plt.hist(df['Sales'], bins=5, color='purple', edgecolor='black')
plt.title('Sales Distribution')
plt.xlabel('Sales ($)')
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()

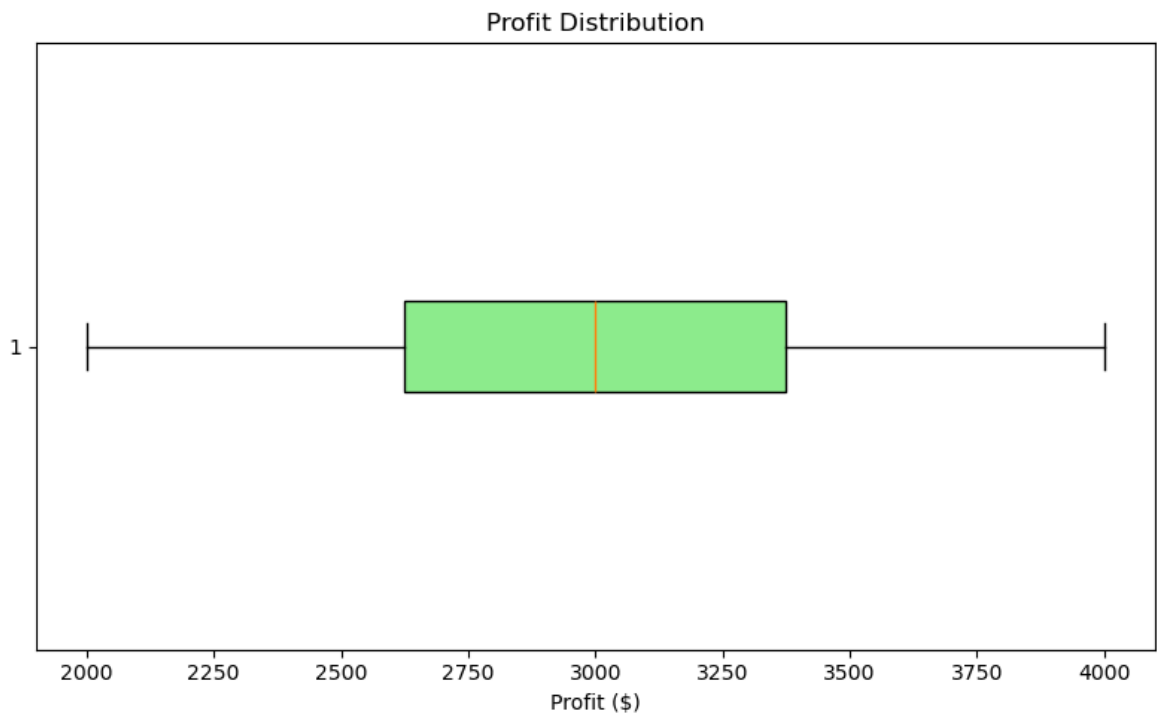
# A histogram showing the distribution of Sales values across months.
```



```
In [8]: # Box Plot: Profit Distribution
plt.figure(figsize=(8, 5))
plt.boxplot(df['Profit'], vert=False, patch_artist=True, boxprops=dict(facecolor
```

```
plt.title('Profit Distribution')  
plt.xlabel('Profit ($)')  
plt.tight_layout()  
plt.show()
```

A box plot showing the spread and outliers of Profit.



In [9]: `!pip install gradio`

Requirement already satisfied: gradio in c:\users\hai\anaconda3\lib\site-packages (5.36.2)

Requirement already satisfied: aiofiles<25.0,>=22.0 in c:\users\hai\anaconda3\lib\site-packages (from gradio) (24.1.0)

Requirement already satisfied: anyio<5.0,>=3.0 in c:\users\hai\anaconda3\lib\site-packages (from gradio) (4.2.0)

Requirement already satisfied: brotli>=1.1.0 in c:\users\hai\anaconda3\lib\site-packages (from gradio) (1.1.0)

Requirement already satisfied: fastapi<1.0,>=0.115.2 in c:\users\hai\anaconda3\lib\site-packages (from gradio) (0.116.1)

Requirement already satisfied: ffmpeg in c:\users\hai\anaconda3\lib\site-packages (from gradio) (0.6.0)

Requirement already satisfied: gradio-client==1.10.4 in c:\users\hai\anaconda3\lib\site-packages (from gradio) (1.10.4)

Requirement already satisfied: groovy~0.1 in c:\users\hai\anaconda3\lib\site-packages (from gradio) (0.1.2)

Requirement already satisfied: httpx<1.0,>=0.24.1 in c:\users\hai\anaconda3\lib\site-packages (from gradio) (0.27.0)

Requirement already satisfied: huggingface-hub>=0.28.1 in c:\users\hai\anaconda3\lib\site-packages (from gradio) (0.33.4)

Requirement already satisfied: jinja2<4.0 in c:\users\hai\anaconda3\lib\site-packages (from gradio) (3.1.4)

Requirement already satisfied: markupsafe<4.0,>=2.0 in c:\users\hai\anaconda3\lib\site-packages (from gradio) (2.1.3)

Requirement already satisfied: numpy<3.0,>=1.0 in c:\users\hai\anaconda3\lib\site-packages (from gradio) (1.26.4)

Requirement already satisfied: orjson~3.0 in c:\users\hai\anaconda3\lib\site-packages (from gradio) (3.10.18)

Requirement already satisfied: packaging in c:\users\hai\anaconda3\lib\site-packages (from gradio) (24.1)

Requirement already satisfied: pandas<3.0,>=1.0 in c:\users\hai\anaconda3\lib\site-packages (from gradio) (2.2.2)

Requirement already satisfied: pillow<12.0,>=8.0 in c:\users\hai\anaconda3\lib\site-packages (from gradio) (10.4.0)

Requirement already satisfied: pydantic<2.12,>=2.0 in c:\users\hai\anaconda3\lib\site-packages (from gradio) (2.8.2)

Requirement already satisfied: pydub in c:\users\hai\anaconda3\lib\site-packages (from gradio) (0.25.1)

Requirement already satisfied: python-multipart>=0.0.18 in c:\users\hai\anaconda3\lib\site-packages (from gradio) (0.0.20)

Requirement already satisfied: pyyaml<7.0,>=5.0 in c:\users\hai\anaconda3\lib\site-packages (from gradio) (6.0.1)

Requirement already satisfied: ruff>=0.9.3 in c:\users\hai\anaconda3\lib\site-packages (from gradio) (0.12.3)

Requirement already satisfied: safehttpx<0.2.0,>=0.1.6 in c:\users\hai\anaconda3\lib\site-packages (from gradio) (0.1.6)

Requirement already satisfied: semantic-version~2.0 in c:\users\hai\anaconda3\lib\site-packages (from gradio) (2.10.0)

Requirement already satisfied: starlette<1.0,>=0.40.0 in c:\users\hai\anaconda3\lib\site-packages (from gradio) (0.47.1)

Requirement already satisfied: tomlkit<0.14.0,>=0.12.0 in c:\users\hai\anaconda3\lib\site-packages (from gradio) (0.13.3)

Requirement already satisfied: typer<1.0,>=0.12 in c:\users\hai\anaconda3\lib\site-packages (from gradio) (0.16.0)

Requirement already satisfied: typing-extensions~4.0 in c:\users\hai\anaconda3\lib\site-packages (from gradio) (4.11.0)

Requirement already satisfied: uvicorn>=0.14.0 in c:\users\hai\anaconda3\lib\site-packages (from gradio) (0.35.0)

Requirement already satisfied: fsspec in c:\users\hai\anaconda3\lib\site-packages (from gradio-client==1.10.4->gradio) (2024.6.1)

Requirement already satisfied: websockets<16.0,>=10.0 in c:\users\hai\anaconda3\lib\site-packages (from gradio-client==1.10.4->gradio) (15.0.1)

Requirement already satisfied: idna>=2.8 in c:\users\hai\anaconda3\lib\site-packages (from anyio<5.0,>=3.0->gradio) (3.7)

Requirement already satisfied: sniffio>=1.1 in c:\users\hai\anaconda3\lib\site-packages (from anyio<5.0,>=3.0->gradio) (1.3.0)

Requirement already satisfied: certifi in c:\users\hai\anaconda3\lib\site-packages (from httpx<1.0,>=0.24.1->gradio) (2025.4.26)

Requirement already satisfied: httpcore==1.* in c:\users\hai\anaconda3\lib\site-packages (from httpx<1.0,>=0.24.1->gradio) (1.0.2)

Requirement already satisfied: h11<0.15,>=0.13 in c:\users\hai\anaconda3\lib\site-packages (from httpcore==1.*->httpx<1.0,>=0.24.1->gradio) (0.14.0)

Requirement already satisfied: filelock in c:\users\hai\anaconda3\lib\site-packages (from huggingface-hub>=0.28.1->gradio) (3.13.1)

Requirement already satisfied: requests in c:\users\hai\anaconda3\lib\site-packages (from huggingface-hub>=0.28.1->gradio) (2.32.3)

Requirement already satisfied: tqdm>=4.42.1 in c:\users\hai\anaconda3\lib\site-packages (from huggingface-hub>=0.28.1->gradio) (4.66.5)

Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\hai\anaconda3\lib\site-packages (from pandas<3.0,>=1.0->gradio) (2.9.0.post0)

Requirement already satisfied: pytz>=2020.1 in c:\users\hai\anaconda3\lib\site-packages (from pandas<3.0,>=1.0->gradio) (2024.1)

Requirement already satisfied: tzdata>=2022.7 in c:\users\hai\anaconda3\lib\site-packages (from pandas<3.0,>=1.0->gradio) (2023.3)

Requirement already satisfied: annotated-types>=0.4.0 in c:\users\hai\anaconda3\lib\site-packages (from pydantic<2.12,>=2.0->gradio) (0.6.0)

Requirement already satisfied: pydantic-core==2.20.1 in c:\users\hai\anaconda3\lib\site-packages (from pydantic<2.12,>=2.0->gradio) (2.20.1)

Requirement already satisfied: click>=8.0.0 in c:\users\hai\anaconda3\lib\site-packages (from typer<1.0,>=0.12->gradio) (8.1.7)

Requirement already satisfied: shellingham>=1.3.0 in c:\users\hai\anaconda3\lib\site-packages (from typer<1.0,>=0.12->gradio) (1.5.0)

Requirement already satisfied: rich>=10.11.0 in c:\users\hai\anaconda3\lib\site-packages (from typer<1.0,>=0.12->gradio) (13.7.1)

Requirement already satisfied: colorama in c:\users\hai\anaconda3\lib\site-packages (from click>=8.0.0->typer<1.0,>=0.12->gradio) (0.4.6)

Requirement already satisfied: six>=1.5 in c:\users\hai\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pandas<3.0,>=1.0->gradio) (1.16.0)

Requirement already satisfied: markdown-it-py>=2.2.0 in c:\users\hai\anaconda3\lib\site-packages (from rich>=10.11.0->typer<1.0,>=0.12->gradio) (2.2.0)

Requirement already satisfied: pygments<3.0.0,>=2.13.0 in c:\users\hai\anaconda3\lib\site-packages (from rich>=10.11.0->typer<1.0,>=0.12->gradio) (2.15.1)

Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\hai\anaconda3\lib\site-packages (from requests->huggingface-hub>=0.28.1->gradio) (3.3.2)

Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\hai\anaconda3\lib\site-packages (from requests->huggingface-hub>=0.28.1->gradio) (2.2.3)

Requirement already satisfied: mdurl~=0.1 in c:\users\hai\anaconda3\lib\site-packages (from markdown-it-py>=2.2.0->rich>=10.11.0->typer<1.0,>=0.12->gradio) (0.1.0)

```
In [10]: import gradio as gr
import pandas as pd
import matplotlib.pyplot as plt
```

```
In [11]: # Sample data
data = {
    "Month": ["Jan", "Feb", "Mar", "Apr", "May", "Jun"],
    "Sales": [10000, 12000, 15000, 13000, 17000, 16000],
```

```
"Profit": [2000, 3000, 4000, 2500, 3500, 3000]
}
```

```
In [12]: df = pd.DataFrame(data)
```

```
In [13]: df
```

```
Out[13]:
```

	Month	Sales	Profit
0	Jan	10000	2000
1	Feb	12000	3000
2	Mar	15000	4000
3	Apr	13000	2500
4	May	17000	3500
5	Jun	16000	3000

```
In [18]: 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='blue', marker='o', label='Sales')
        plt.title('Sales Trend 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, label='Sales', color='skyblue')
        plt.bar(df['Month'], df['Profit'], width=width, label='Profit', color='coral')
        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='%1.1f%%', startangle=90)
        plt.title('Profit Distribution by Month')
```



```

elif plot_type == "Scatter Plot":
    plt.scatter(df['Sales'], df['Profit'], color='green', s=100, edgecolors=
    plt.title('Sales vs Profit')
    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')
    plt.xlabel('Sales ($)')
    plt.ylabel('Frequency')

elif plot_type == "Box Plot":
    plt.boxplot(df['Profit'], vert=False, patch_artist=True, boxprops=dict(f
    plt.title('Profit Distribution')
    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://127.0.0.1:7860>

* To create a public link, set `share=True` in `launch()`.

Sales & Profit Visual Explorer

Choose a chart type to visualize the data.

Choose Plot Type

☐ Line Plot

☐ Stacked Bar Chart

☐ Pie Chart

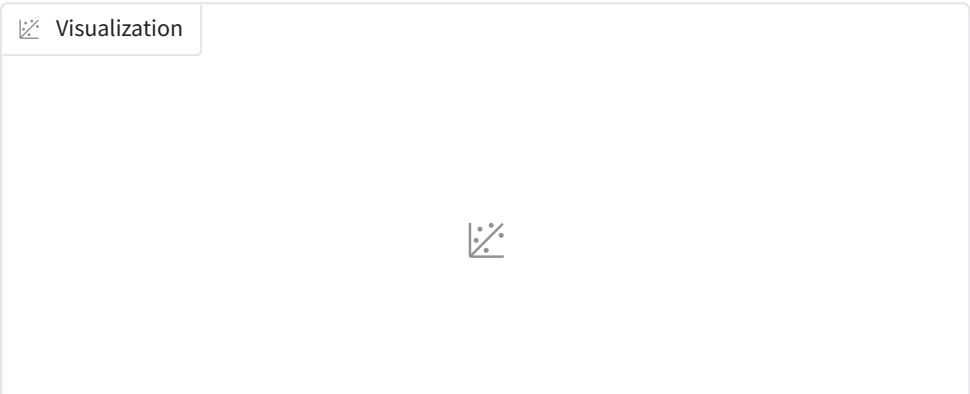
☐ Scatter Plot

☐ Histogram

☐ Box Plot

Clear

Submit



Out[18]:

In []: