

Matplotlib Visualization Using Gradio

```
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)
```

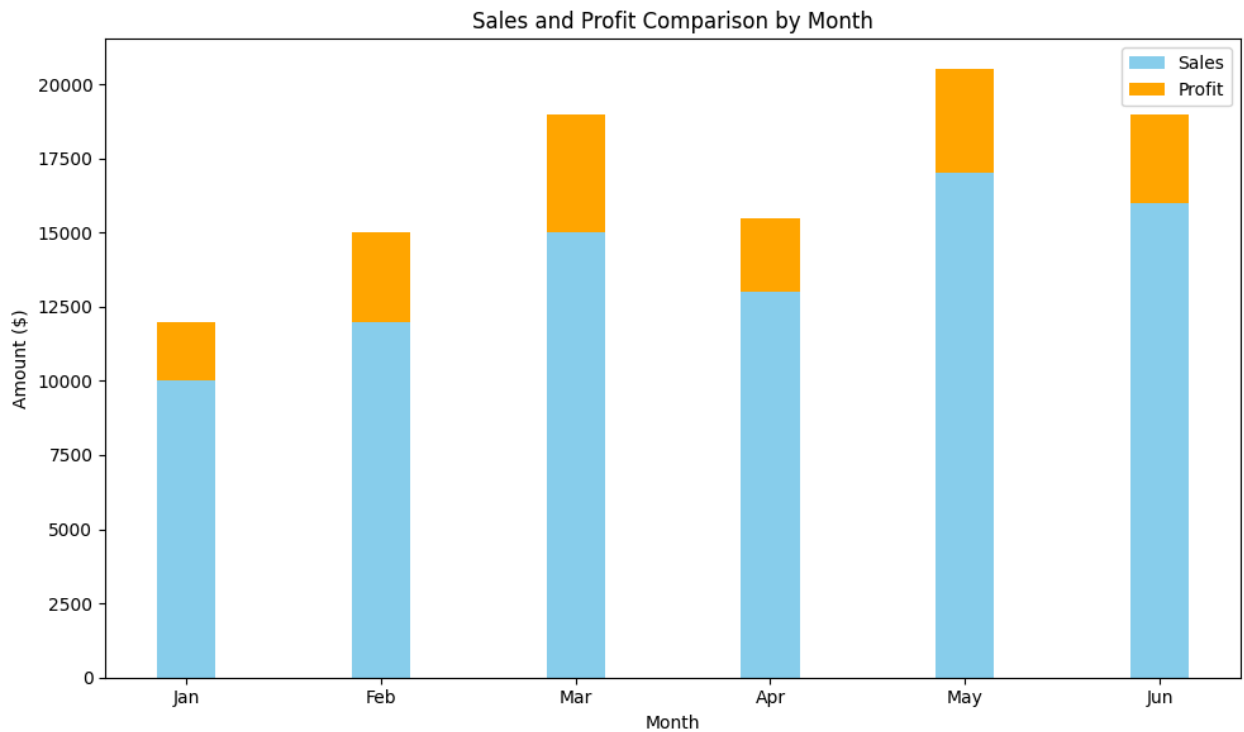
1- line plot sales over time

```
In [2]: 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()
```

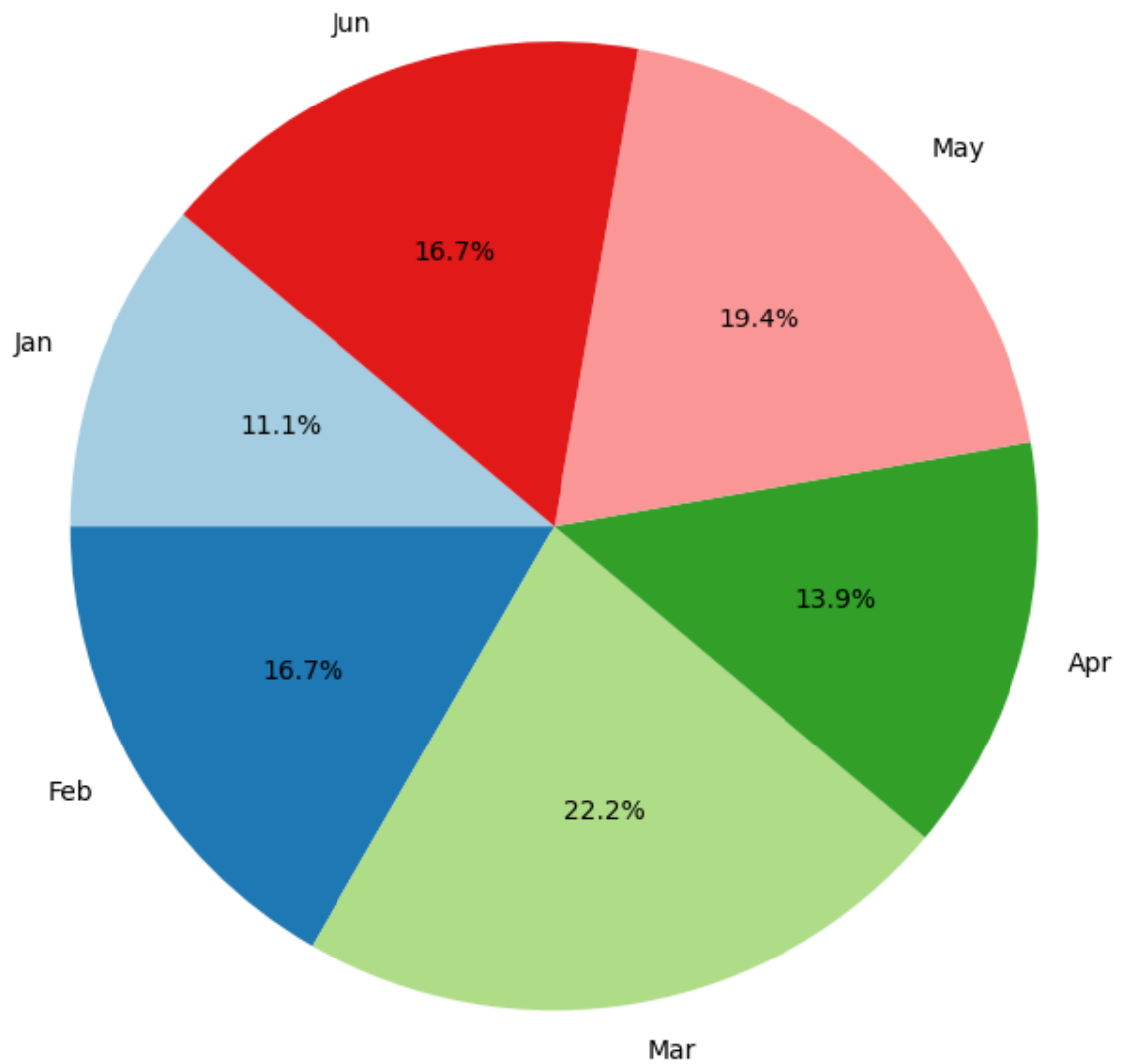


```
In [3]: # Bar Plot: Sales vs Profit by Month
plt.figure(figsize=(10, 6))
width = 0.3
plt.bar(df['Month'], df['Sales'], width=width, label='Sales', color='skyblue')
plt.bar(df['Month'], df['Profit'], width=width, label='Profit', color='orange')
plt.title('Sales and Profit Comparison by Month')
plt.xlabel('Month')
plt.ylabel('Amount ($)')
plt.legend()
plt.tight_layout()
plt.show()
```

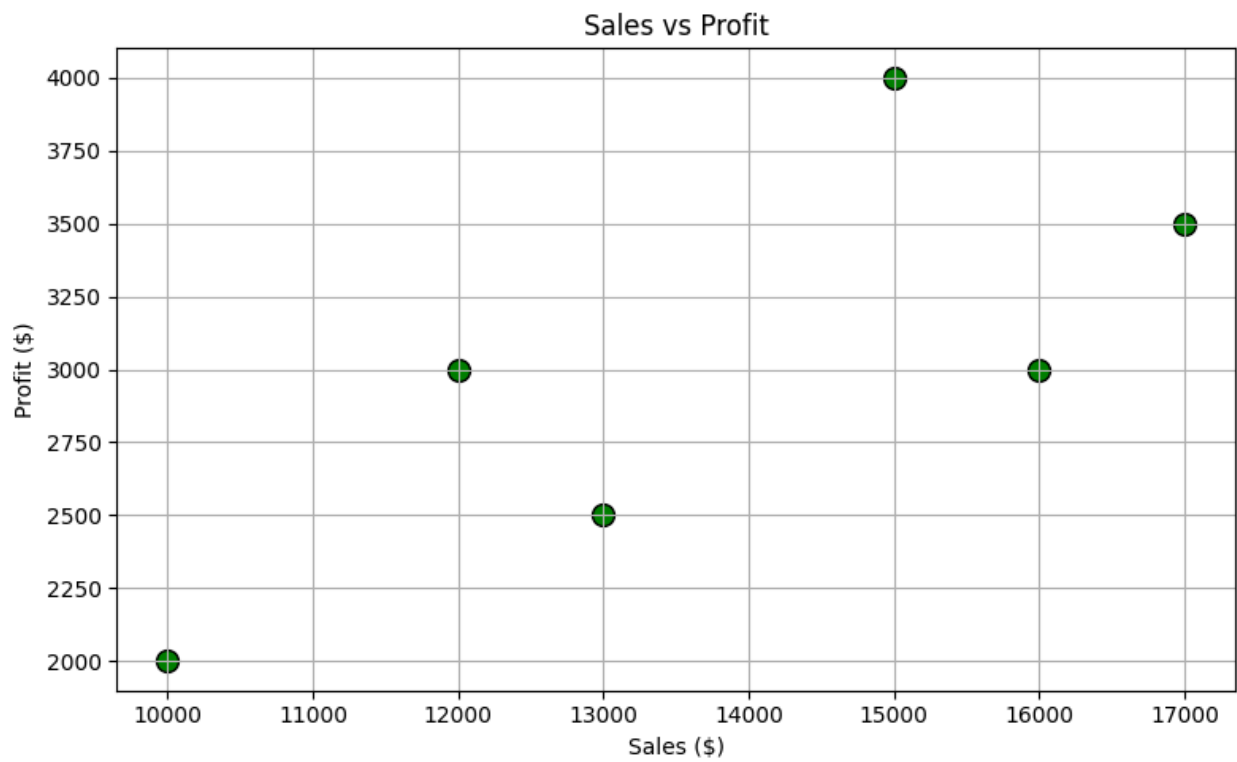


```
In [4]: # Pie Chart: Profit Distribution by Month
plt.figure(figsize=(7, 7))
plt.pie(df['Profit'], labels=df['Month'], autopct='%1.1f%%', startangle=140, c
plt.title('Profit Distribution by Month')
plt.tight_layout()
plt.show()
```

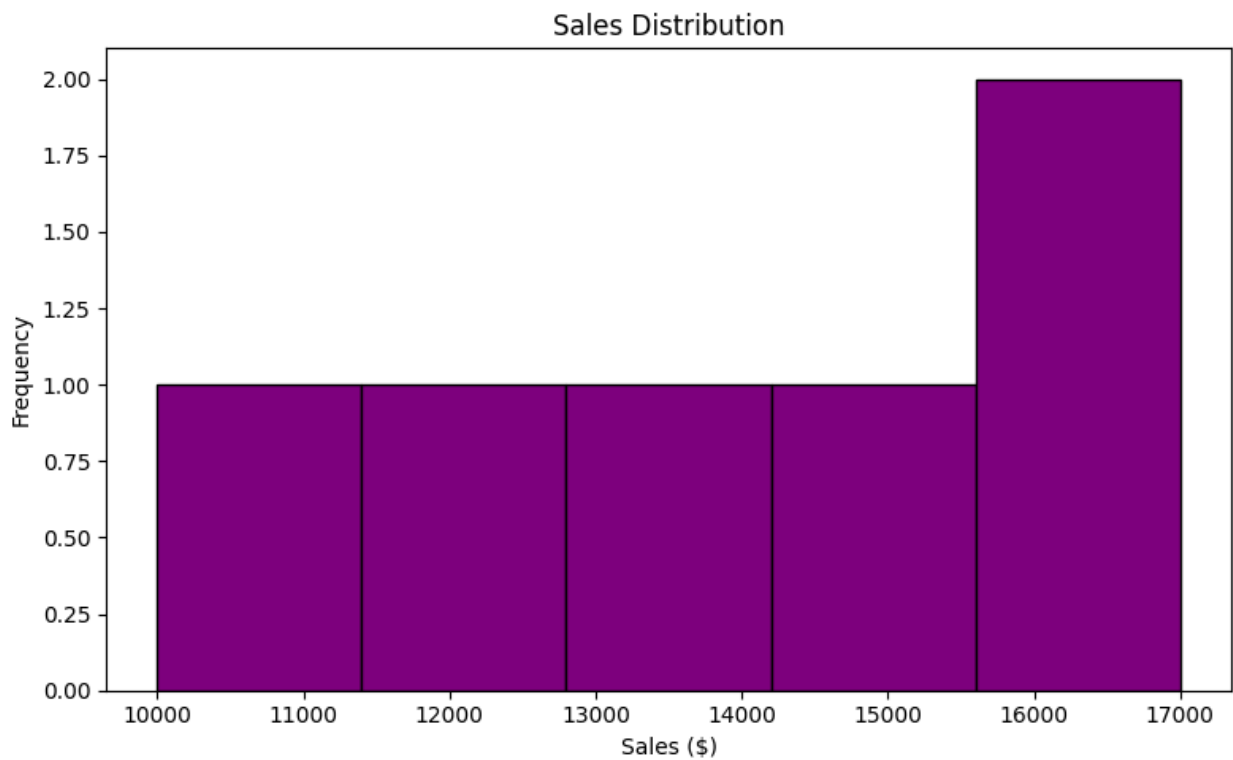
Profit Distribution by Month



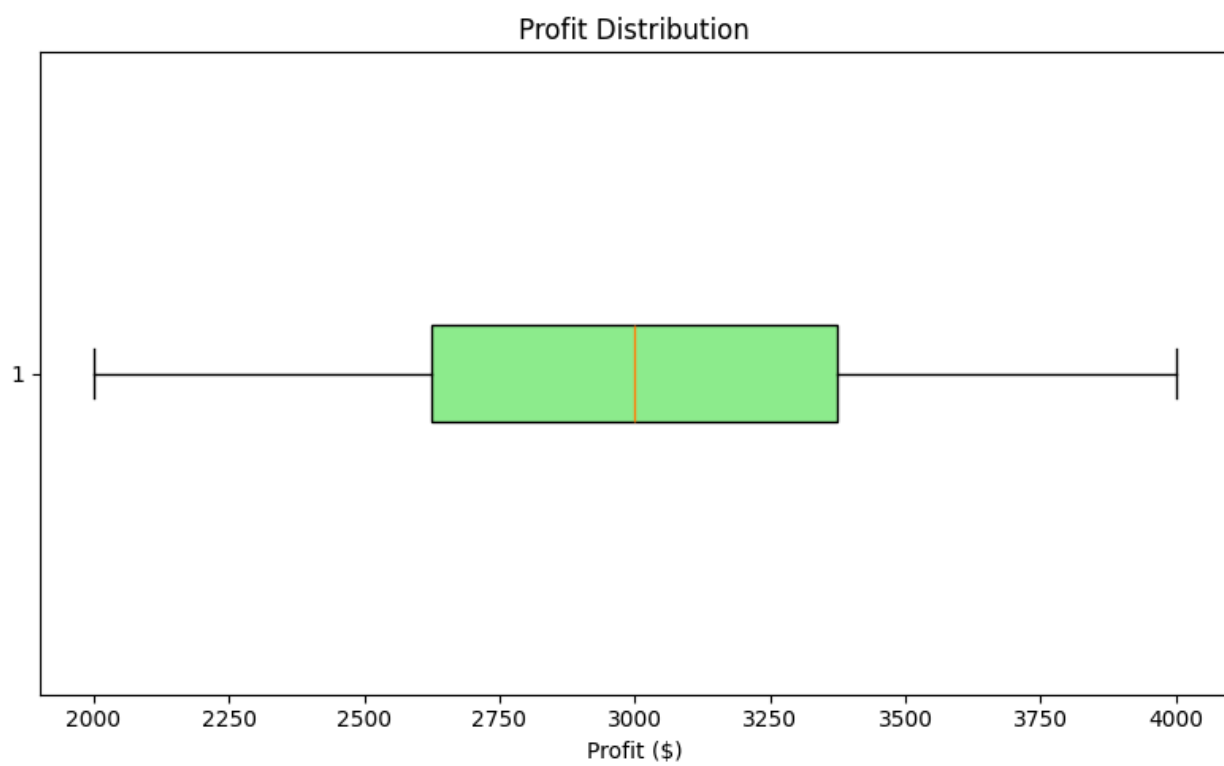
```
In [5]: # Scatter Plot: Sales vs Profit (Correlation)
plt.figure(figsize=(8, 5))
plt.scatter(df['Sales'], df['Profit'], color='green', s=100, edgecolors='black')
plt.title('Sales vs Profit')
plt.xlabel('Sales ($)')
plt.ylabel('Profit ($)')
plt.grid(True)
plt.tight_layout()
plt.show()
```



```
In [6]: # 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()
```



```
In [7]: # Box Plot: Profit Distribution
plt.figure(figsize=(8, 5))
plt.boxplot(df['Profit'], vert=False, patch_artist=True, boxprops=dict(facecol
plt.title('Profit Distribution')
plt.xlabel('Profit ($)')
plt.tight_layout()
plt.show()
```



In [8]: `!pip install gradio`

Requirement already satisfied: gradio in /usr/local/lib/python3.12/dist-packages (5.44.1)

Requirement already satisfied: aiofiles<25.0,>=22.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (24.1.0)

Requirement already satisfied: anyio<5.0,>=3.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (4.10.0)

Requirement already satisfied: brotli>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (1.1.0)

Requirement already satisfied: fastapi<1.0,>=0.115.2 in /usr/local/lib/python3.12/dist-packages (from gradio) (0.116.1)

Requirement already satisfied: ffmpeg in /usr/local/lib/python3.12/dist-packages (from gradio) (0.6.1)

Requirement already satisfied: gradio-client==1.12.1 in /usr/local/lib/python3.12/dist-packages (from gradio) (1.12.1)

Requirement already satisfied: groovy~=0.1 in /usr/local/lib/python3.12/dist-packages (from gradio) (0.1.2)

Requirement already satisfied: httpx<1.0,>=0.24.1 in /usr/local/lib/python3.12/dist-packages (from gradio) (0.28.1)

Requirement already satisfied: huggingface-hub<1.0,>=0.33.5 in /usr/local/lib/python3.12/dist-packages (from gradio) (0.34.4)

Requirement already satisfied: jinja2<4.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (3.1.6)

Requirement already satisfied: markupsafe<4.0,>=2.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (3.0.2)

Requirement already satisfied: numpy<3.0,>=1.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (2.0.2)

Requirement already satisfied: orjson~=3.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (3.11.3)

Requirement already satisfied: packaging in /usr/local/lib/python3.12/dist-packages (from gradio) (25.0)

Requirement already satisfied: pandas<3.0,>=1.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (2.2.2)

Requirement already satisfied: pillow<12.0,>=8.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (11.3.0)

Requirement already satisfied: pydantic<2.12,>=2.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (2.11.7)

Requirement already satisfied: pydub in /usr/local/lib/python3.12/dist-packages (from gradio) (0.25.1)

Requirement already satisfied: python-multipart>=0.0.18 in /usr/local/lib/python3.12/dist-packages (from gradio) (0.0.20)

Requirement already satisfied: pyyaml<7.0,>=5.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (6.0.2)

Requirement already satisfied: ruff>=0.9.3 in /usr/local/lib/python3.12/dist-packages (from gradio) (0.12.11)

Requirement already satisfied: safehttpx<0.2.0,>=0.1.6 in /usr/local/lib/python3.12/dist-packages (from gradio) (0.1.6)

Requirement already satisfied: semantic-version~=2.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (2.10.0)

Requirement already satisfied: starlette<1.0,>=0.40.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (0.47.3)

Requirement already satisfied: tomlkit<0.14.0,>=0.12.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (0.13.3)

Requirement already satisfied: typer<1.0,>=0.12 in /usr/local/lib/python3.12/dist-packages (from gradio) (0.17.3)

Requirement already satisfied: typing-extensions~=4.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (4.15.0)

Requirement already satisfied: uvicorn>=0.14.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (0.35.0)

Requirement already satisfied: fsspec in /usr/local/lib/python3.12/dist-packages (from gradio-client==1.12.1->gradio) (2025.3.0)

Requirement already satisfied: websockets<16.0,>=10.0 in /usr/local/lib/python3.12/dist-packages (from gradio-client==1.12.1->gradio) (15.0.1)

Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.12/dist-packages (from anyio<5.0,>=3.0->gradio) (3.10)

Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.12/dist-packages (from anyio<5.0,>=3.0->gradio) (1.3.1)

Requirement already satisfied: certifi in /usr/local/lib/python3.12/dist-packages (from httpx<1.0,>=0.24.1->gradio) (2025.8.3)

Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.12/dist-packages (from httpx<1.0,>=0.24.1->gradio) (1.0.9)

Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.12/dist-packages (from httpcore==1.*->httpx<1.0,>=0.24.1->gradio) (0.16.0)

Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<1.0,>=0.33.5->gradio) (3.19.1)

Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<1.0,>=0.33.5->gradio) (2.32.4)

Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<1.0,>=0.33.5->gradio) (4.67.1)

Requirement already satisfied: hf-xet<2.0.0,>=1.1.3 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<1.0,>=0.33.5->gradio) (1.1.9)

Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas<3.0,>=1.0->gradio) (2.9.0.post0)

Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas<3.0,>=1.0->gradio) (2025.2)

Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas<3.0,>=1.0->gradio) (2025.2)

Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from pydantic<2.12,>=2.0->gradio) (0.7.0)

Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/python3.12/dist-packages (from pydantic<2.12,>=2.0->gradio) (2.33.2)

Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.12/dist-packages (from pydantic<2.12,>=2.0->gradio) (0.4.1)

Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.12/dist-packages (from typer<1.0,>=0.12->gradio) (8.2.1)

Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3.12/dist-packages (from typer<1.0,>=0.12->gradio) (1.5.4)

Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.12/dist-packages (from typer<1.0,>=0.12->gradio) (13.9.4)

Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas<3.0,>=1.0->gradio) (1.17.0)

Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.12/dist-packages (from rich>=10.11.0->typer<1.0,>=0.12->gradio) (4.0.0)

Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.12/dist-packages (from rich>=10.11.0->typer<1.0,>=0.12->gradio) (2.19.2)

Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests->huggingface-hub<1.0,>=0.33.5->gradio) (3.4.3)

Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/

dist-packages (from requests->huggingface-hub<1.0,>=0.33.5->gradio) (2.5.0)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.12/dist-packages (from markdown-it-py>=2.2.0->rich>=10.11.0->typer<1.0,>=0.12->gradio) (0.1.2)

```
In [9]: 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='blue')
        plt.bar(df['Month'], df['Profit'], width=width, label='Profit', color='green')
        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, edgecolor='black')
        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
    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()

```

It looks like you are running Gradio on a hosted Jupyter notebook, which requires `share=True`. Automatically setting `share=True` (you can turn this off by setting `share=False` in `launch()` explicitly).

Colab notebook detected. To show errors in colab notebook, set debug=True in launch()

* Running on public URL: <https://3ab2704b4aa906b10e.gradio.live>

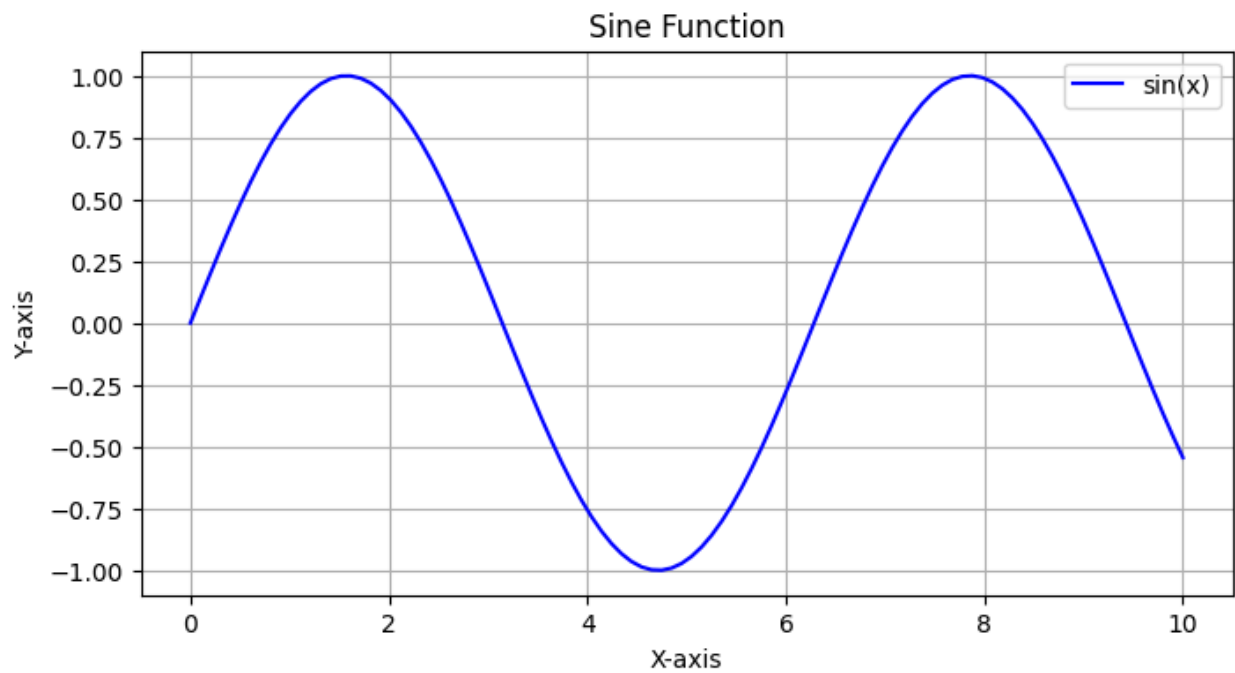
This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the working directory to deploy to Hugging Face Spaces (<https://huggingface.co/spaces>)

Out[9]:

```
In [10]: import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [14]: #line plot
x = np.linspace(0, 10, 100)
y = np.sin(x)

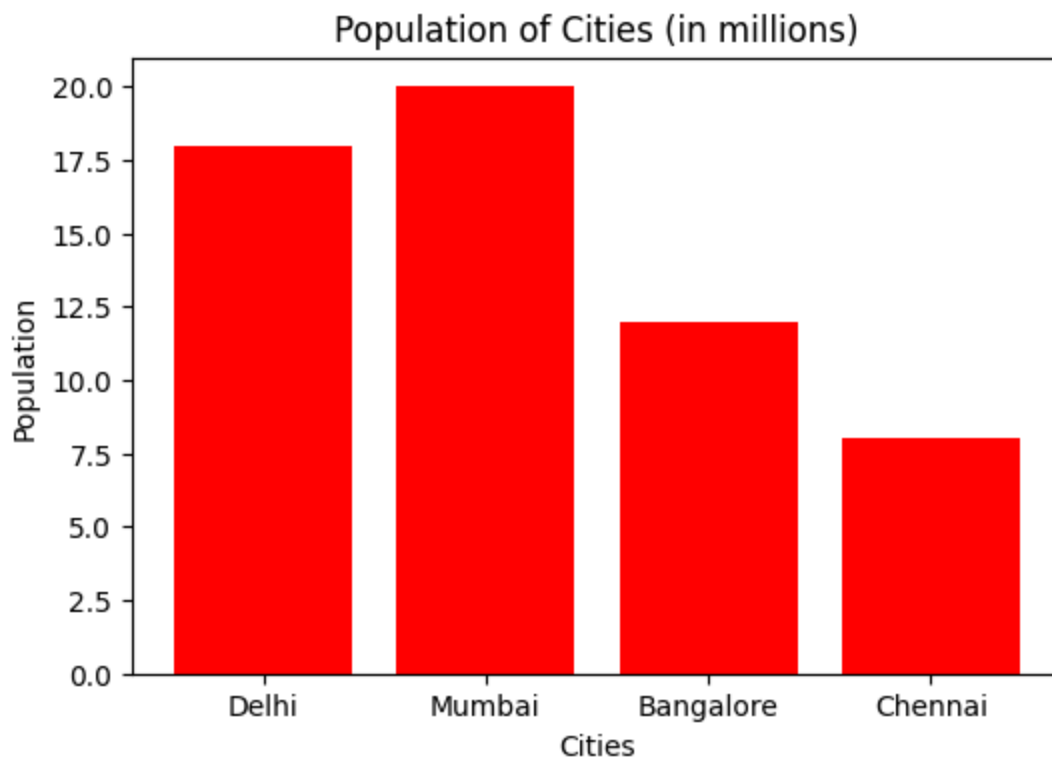
plt.figure(figsize=(8, 4))
plt.plot(x, y, label="sin(x)", color='blue')
plt.title("Sine Function")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.grid(True)
plt.legend()
plt.show()
```



In [13]: *#bar chat*

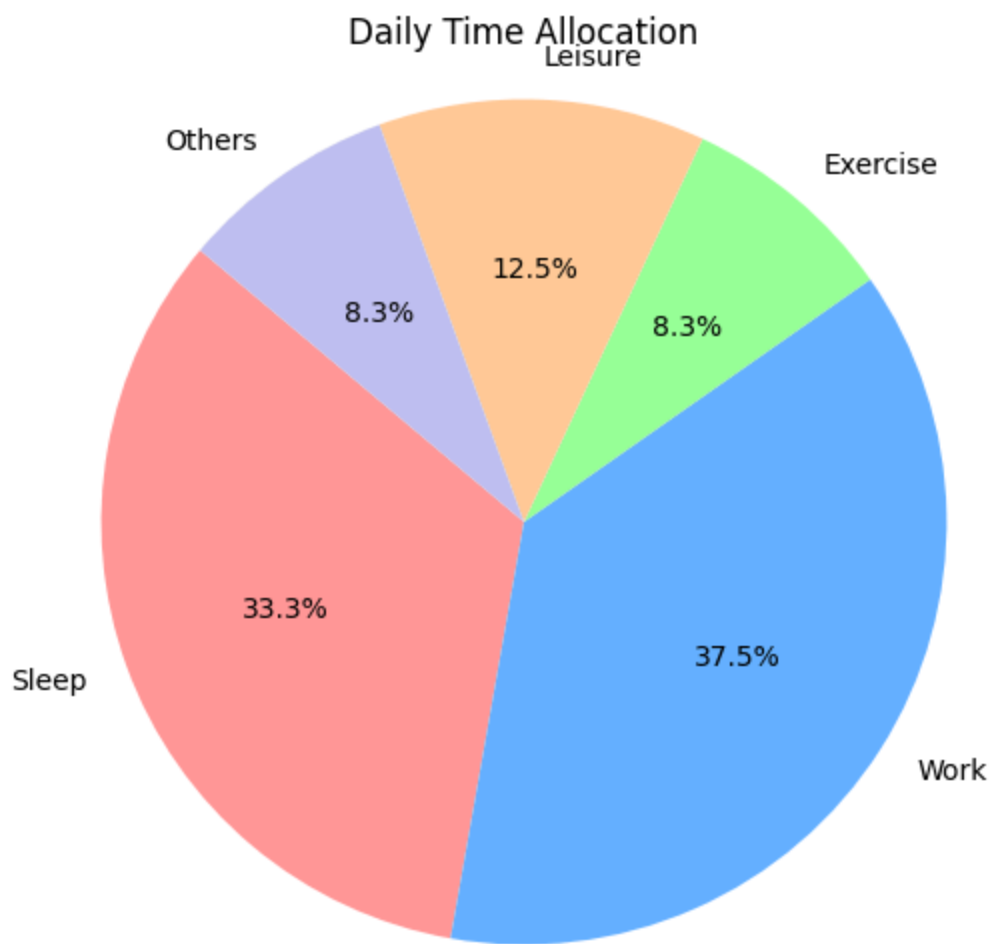
```
cities = ['Delhi', 'Mumbai', 'Bangalore', 'Chennai']
population = [18, 20, 12, 8]

plt.figure(figsize=(6,4))
plt.bar(cities, population, color='red')
plt.title("Population of Cities (in millions)")
plt.xlabel("Cities")
plt.ylabel("Population")
plt.show()
```



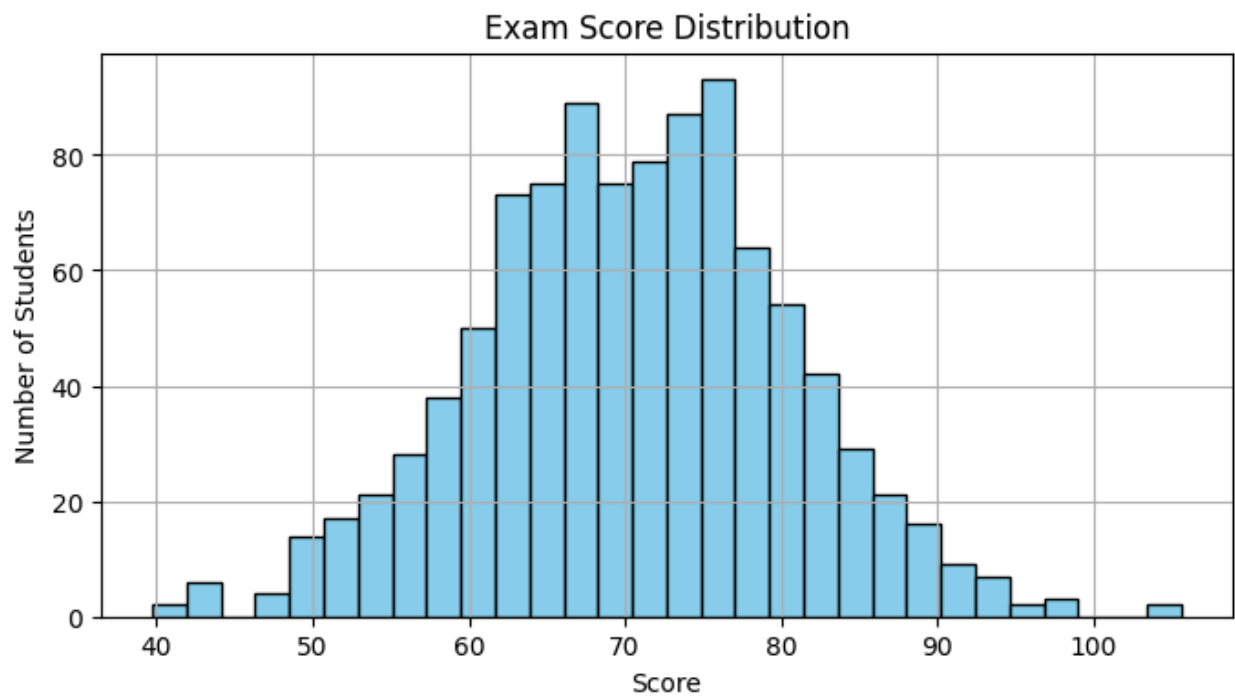
```
In [15]: #pie chart
labels = ['Sleep', 'Work', 'Exercise', 'Leisure', 'Others']
sizes = [8, 9, 2, 3, 2]
colors = ['#ff9999', '#66b3ff', '#99ff99', '#ffcc99', '#c2c2f0']

plt.figure(figsize=(6,6))
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=140)
plt.title("Daily Time Allocation")
plt.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()
```



```
In [16]: #histogarm
scores = np.random.normal(loc=70, scale=10, size=1000)

plt.figure(figsize=(8,4))
plt.hist(scores, bins=30, color='skyblue', edgecolor='black')
plt.title("Exam Score Distribution")
plt.xlabel("Score")
plt.ylabel("Number of Students")
plt.grid(True)
plt.show()
```



```
In [17]: #Heatmap

data = np.random.rand(10,10)
plt.figure(figsize=(6,5))
plt.imshow(data, cmap='hot', interpolation='nearest')
plt.colorbar(label='value')
plt.title("Random Heatmap")
plt.show()
```

