

In [16]: `import pandas as pd`

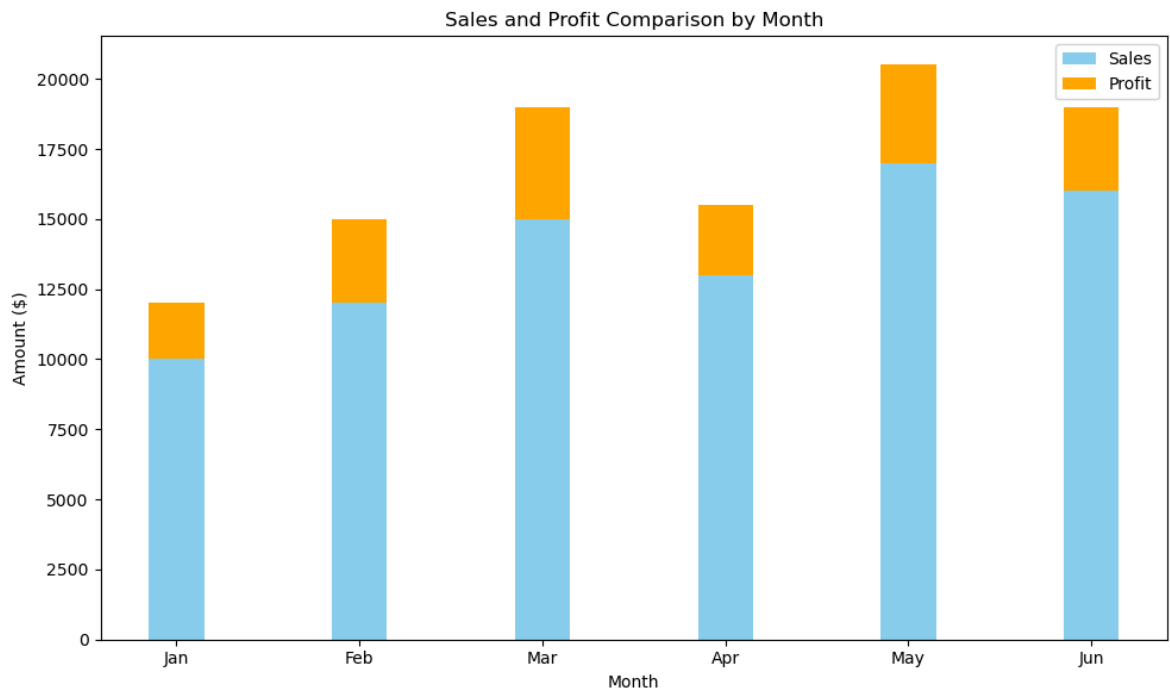
```
data = {  
    "Month": ["Jan", "Feb", "Mar", "Apr", "May", "Jun"],  
    "Sales": [10000, 12000, 15000, 13000, 17000, 16000],  
    "Profit": [2000, 3000, 4000, 2500, 3500, 3000]  
}  
df = pd.DataFrame(data)
```

In [17]: `import matplotlib.pyplot as plt`

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

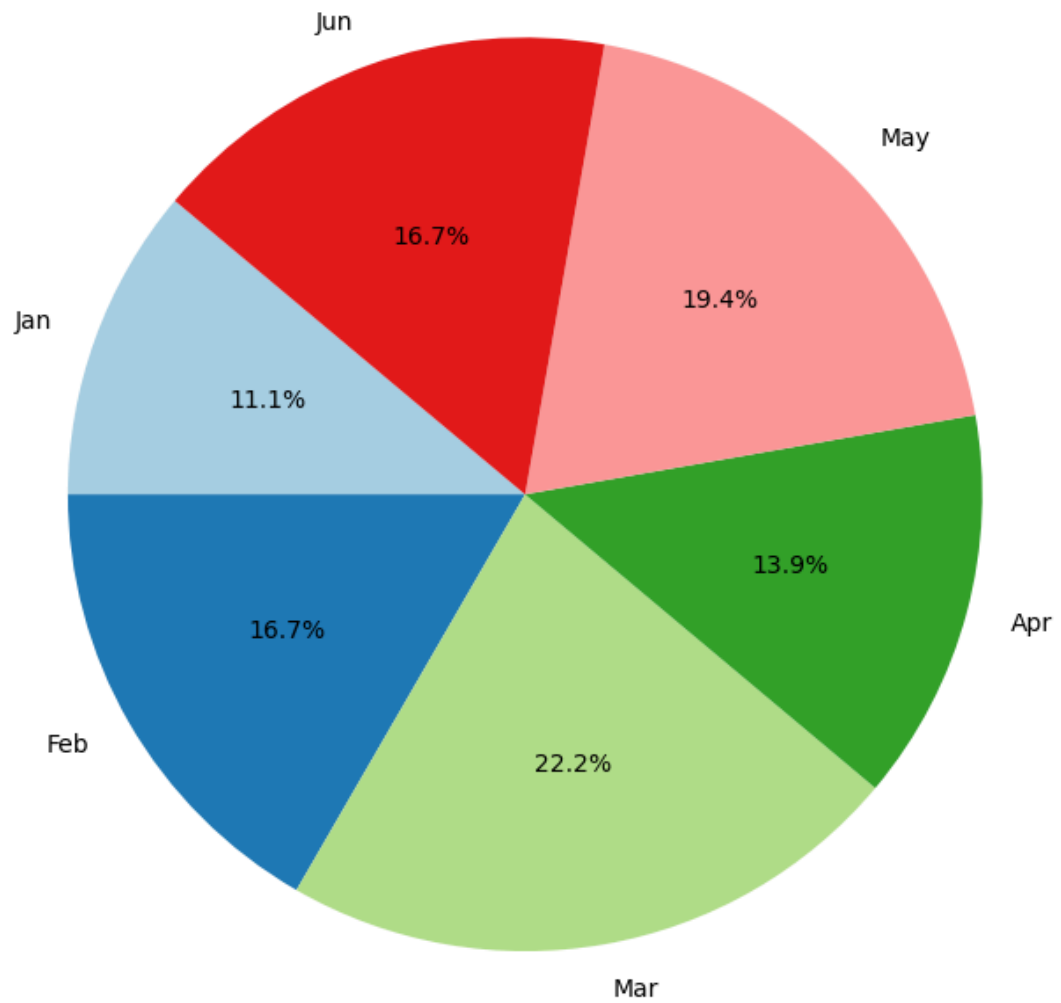


```
In [18]: plt.figure(figsize=(10, 6))  
width = 0.3  
plt.bar(df['Month'], df['Sales'], width=width, label='Sales', color='skyblue')  
plt.bar(df['Month'], df['Profit'], width=width, label='Profit', color='orange',  
plt.title('Sales and Profit Comparison by Month')  
plt.xlabel('Month')  
plt.ylabel('Amount ($)')  
plt.legend()  
plt.tight_layout()  
plt.show()
```

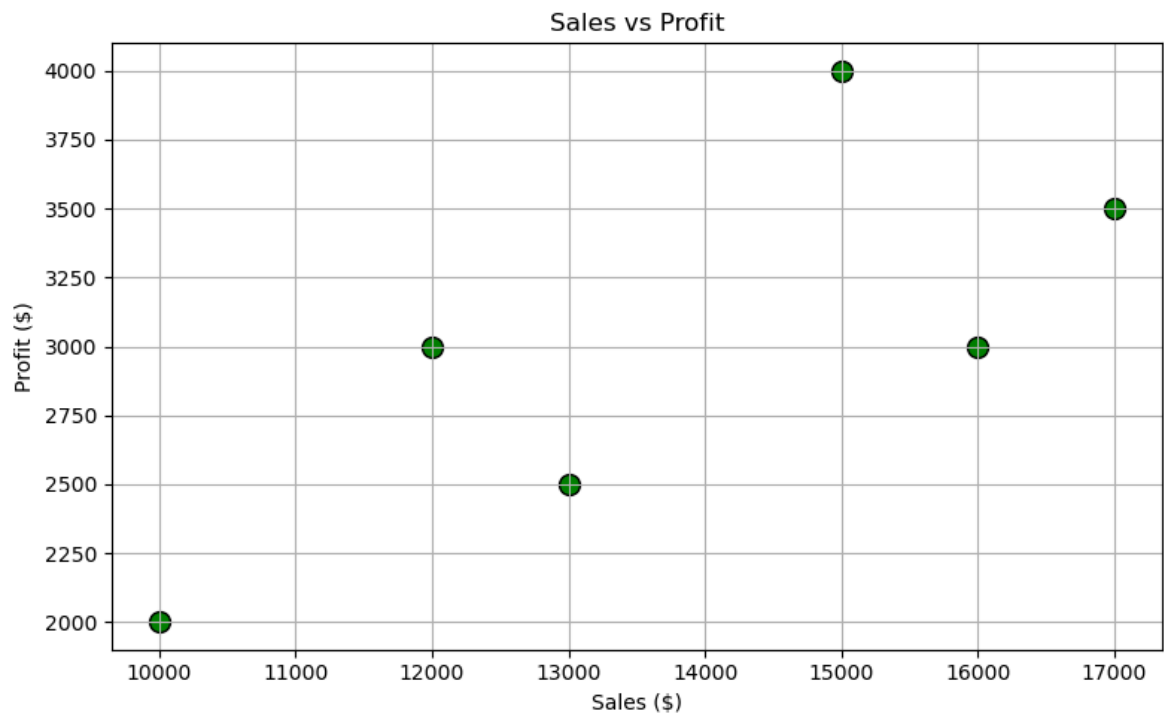


```
In [20]: plt.figure(figsize=(7, 7))
plt.pie(df['Profit'], labels=df['Month'], autopct='%1.1f%%', startangle=140, col
plt.title('Profit Distribution by Month')
plt.tight_layout()
plt.show()
```

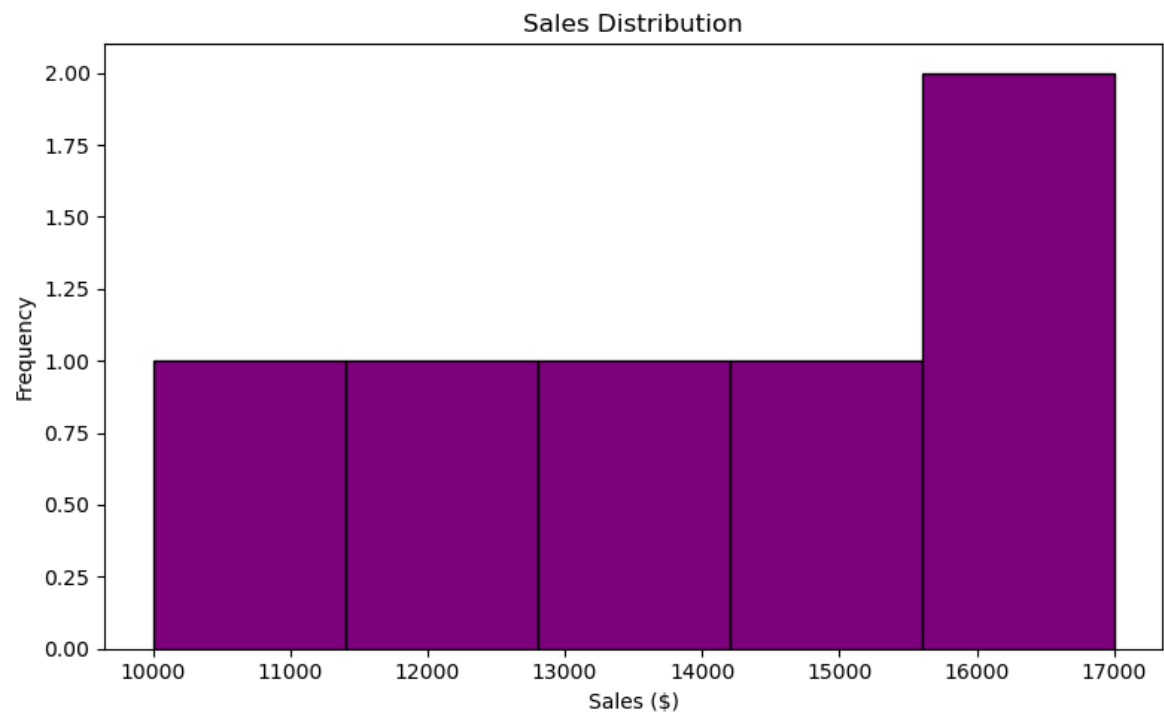
Profit Distribution by Month



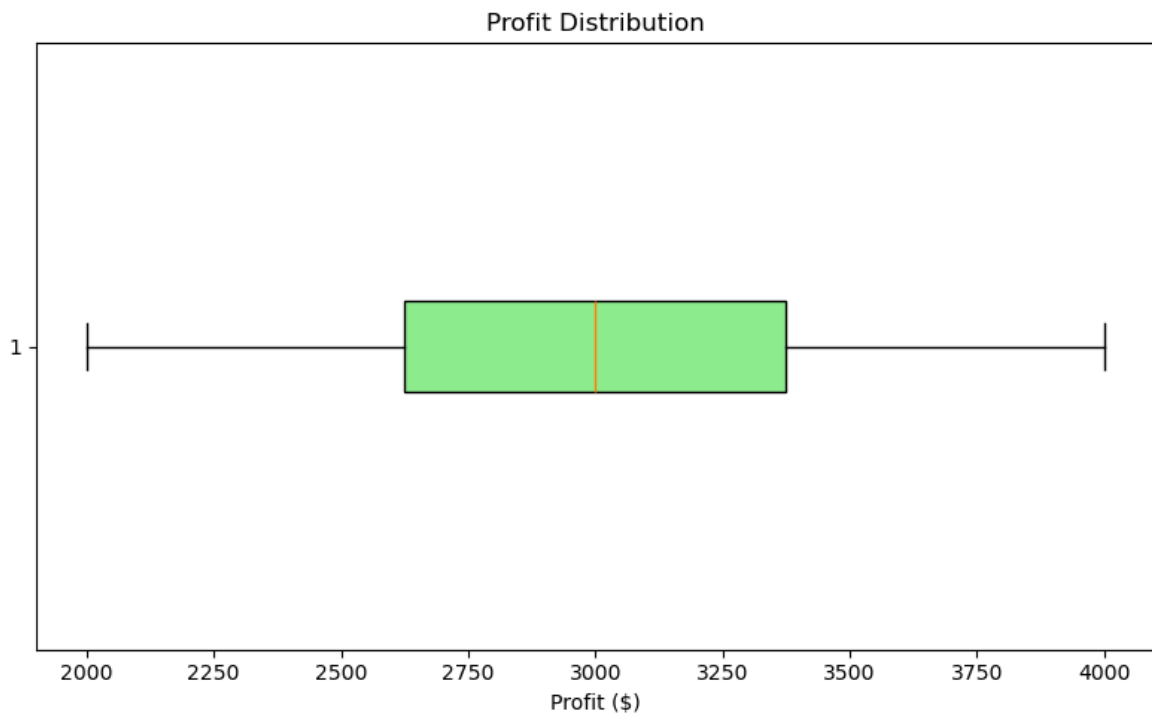
```
In [21]: plt.figure(figsize=(8, 5))
plt.scatter(df['Sales'], df['Profit'], color='green', s=100, edgecolors='black')
plt.title('Sales vs Profit')
plt.xlabel('Sales ($)')
plt.ylabel('Profit ($)')
plt.grid(True)
plt.tight_layout()
plt.show()
```



```
In [23]: 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 [24]: 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()
```



```
In [25]: 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=0)
```

```

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()`.

Out[25]:

In []:

In []: