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]:

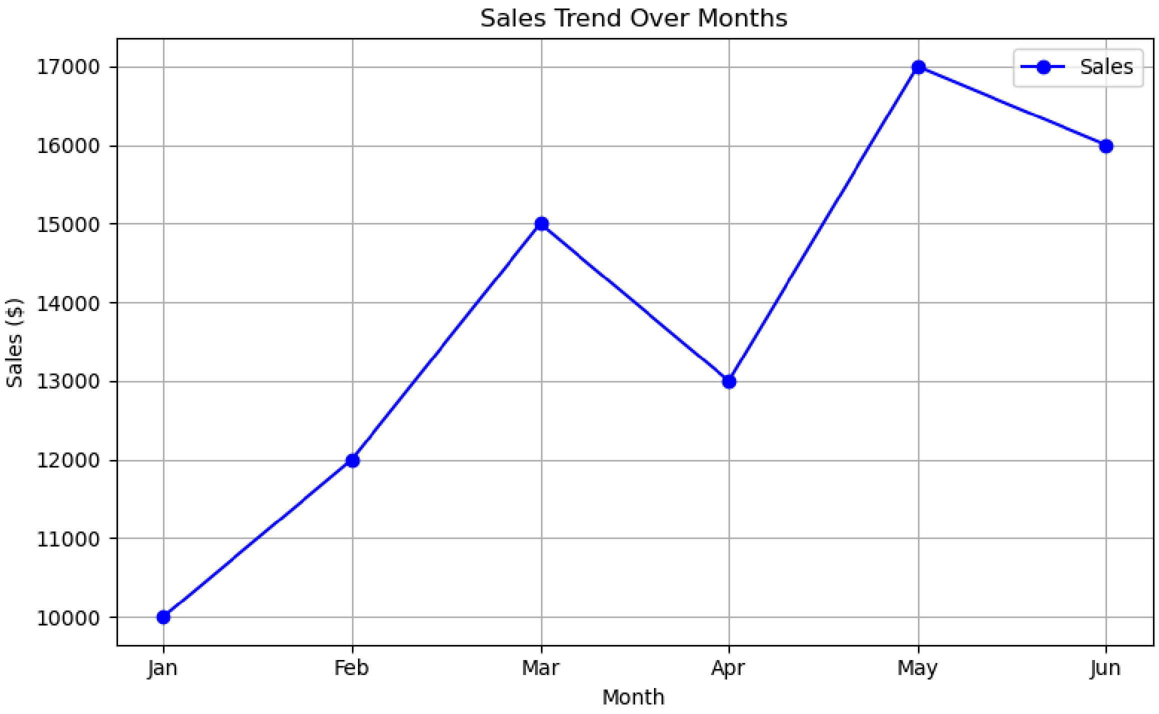
```
df
```

Out[2]:

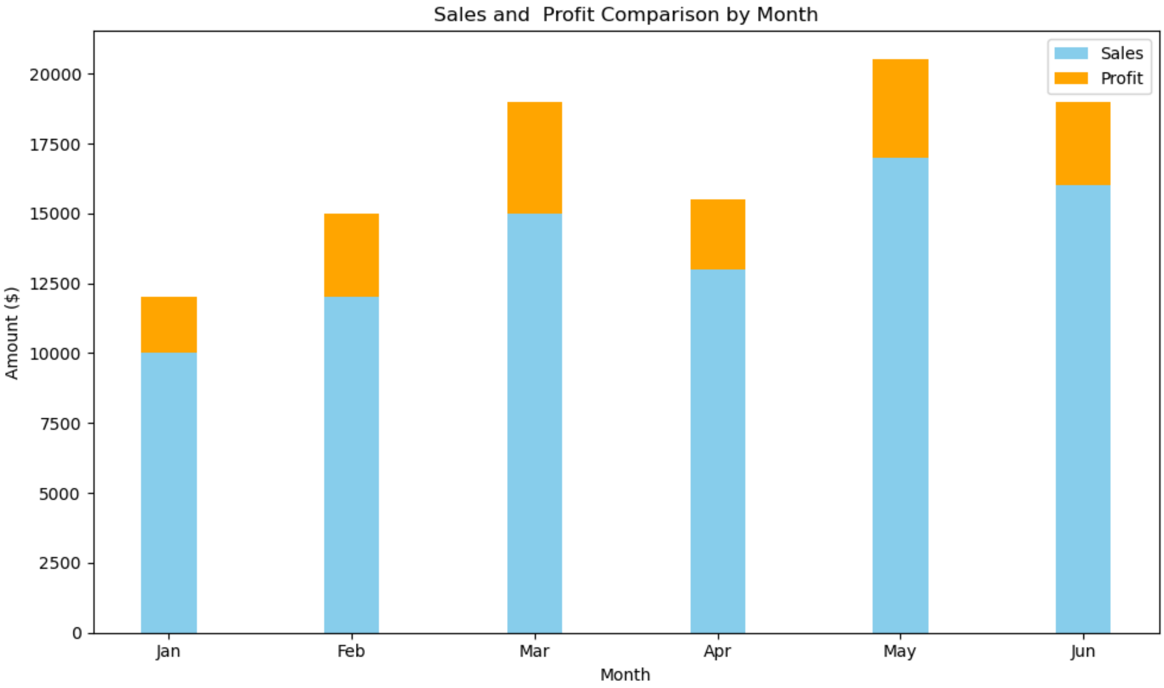
	Month	Sales	Profit
0	Jan	10000	2000
1	Feb	12000	3000
2	Mar	15000	4000
3	Apr	13000	2500
4	May	17000	3500
5	Jun	16000	3000

In [3]:

```
# 1- line plot sales over time

import matplotlib.pyplot as plt

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

# Output: A line plot showing trendz over months
```

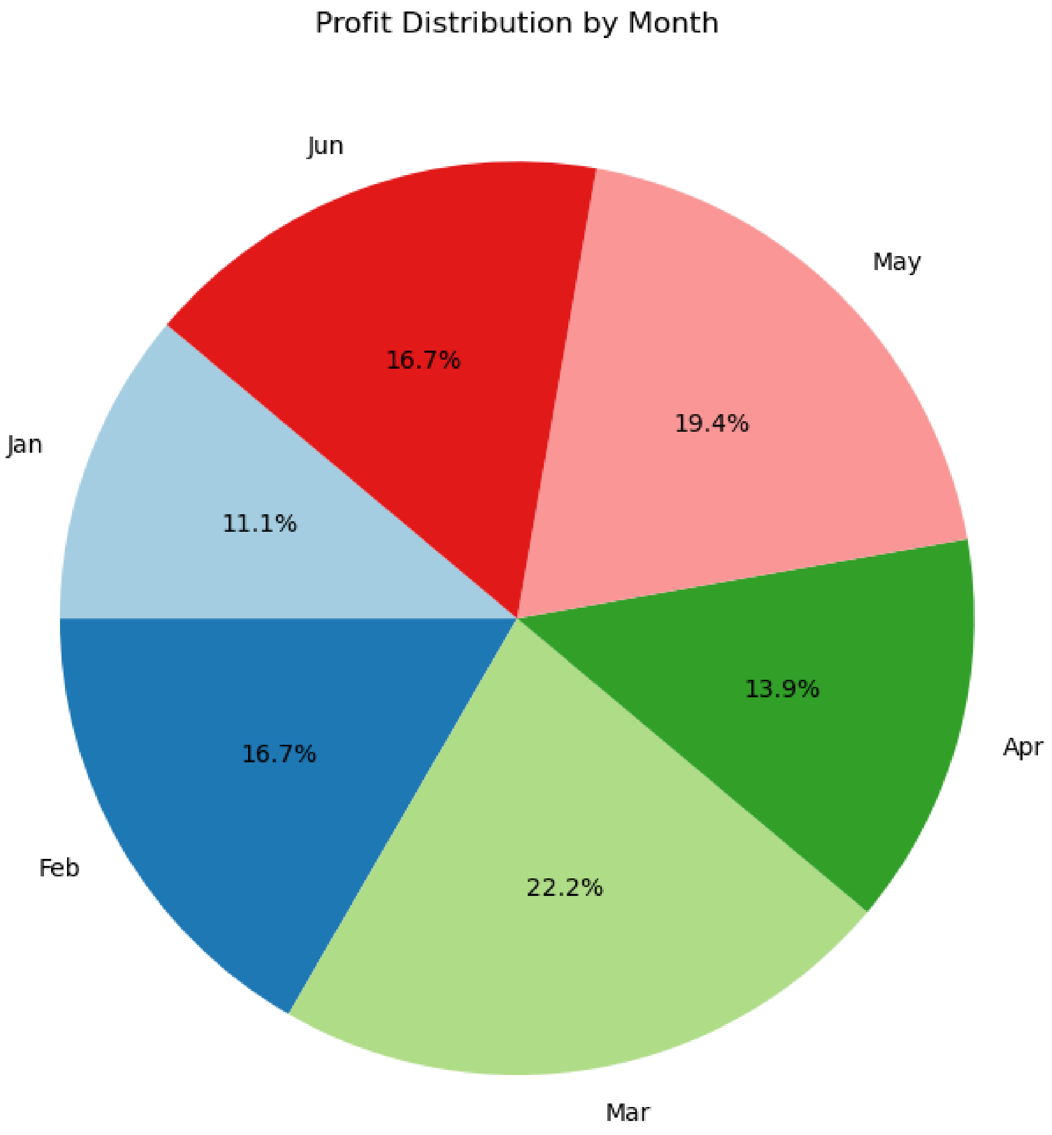


```
In [6]: # Bar Plt: Sales vs Profit by Month
        plt.figure(figsize=(10,6))
        width = 0.3
        plt.bar(df['Month'], df['Sales'],width=width, color= 'skyblue', label= 'Sales')
        plt.bar(df['Month'], df['Profit'],width=width, color= 'orange', label= 'Profit',
        plt.title('Sales 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.
```



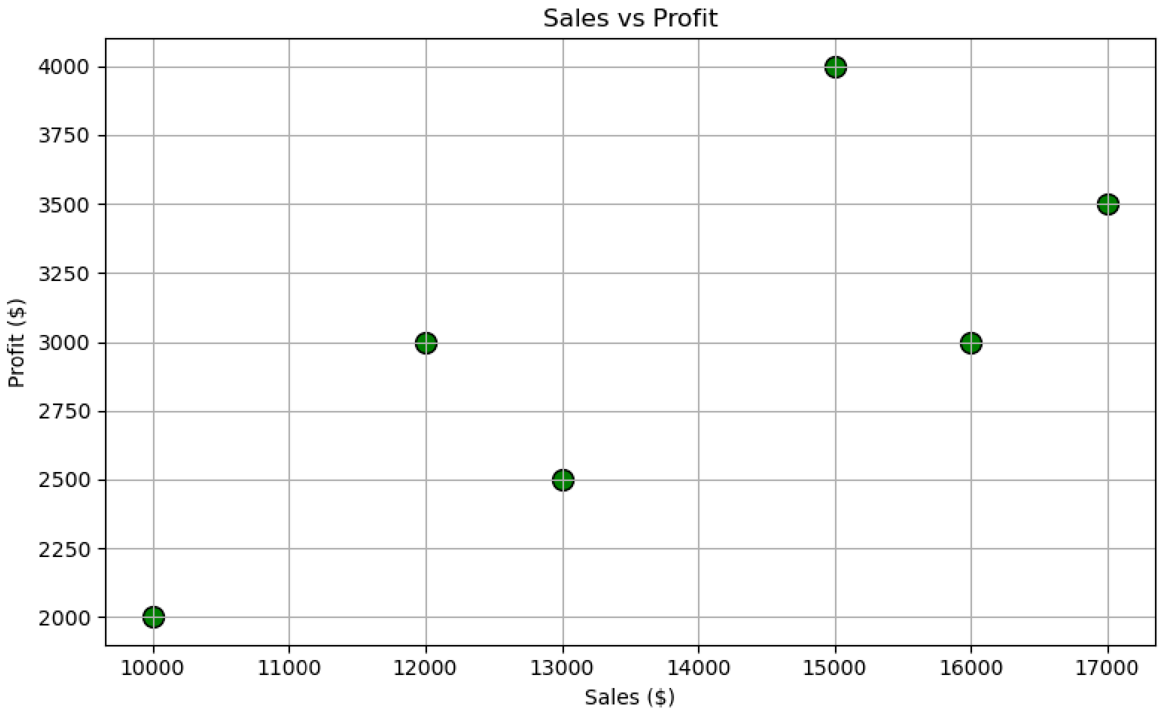
```
In [7]: # Pie Chart: Profit Distribution by Month
```

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

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



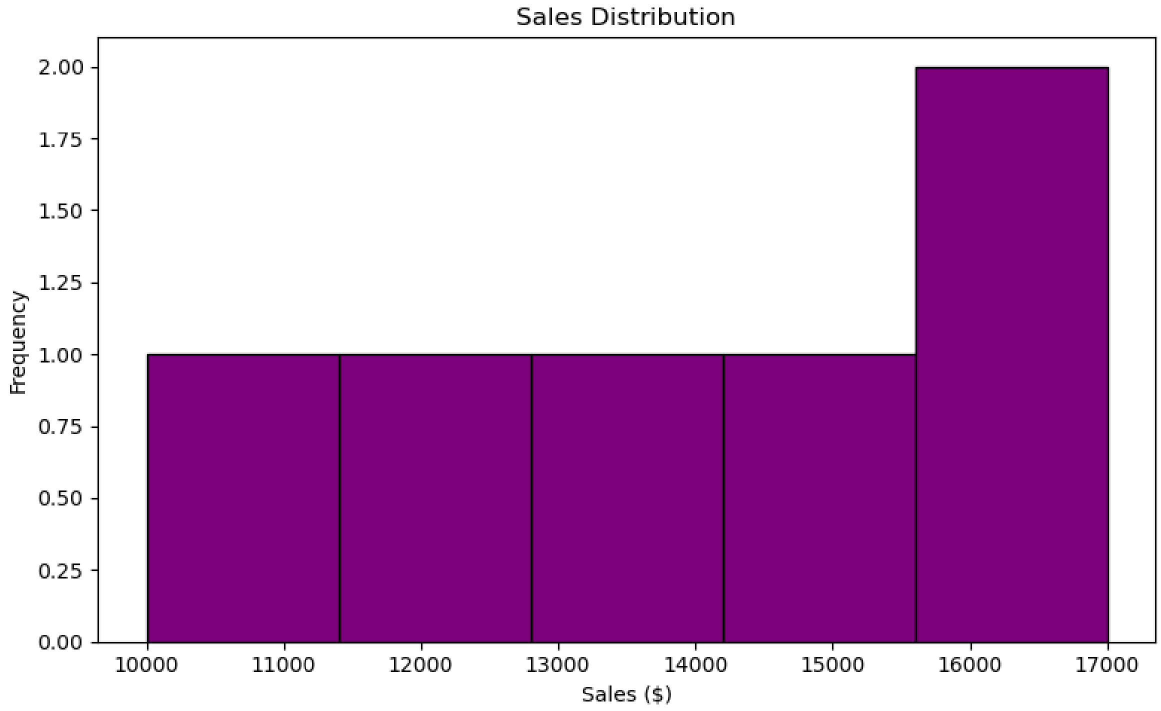
In [8]:

```
# 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.
```



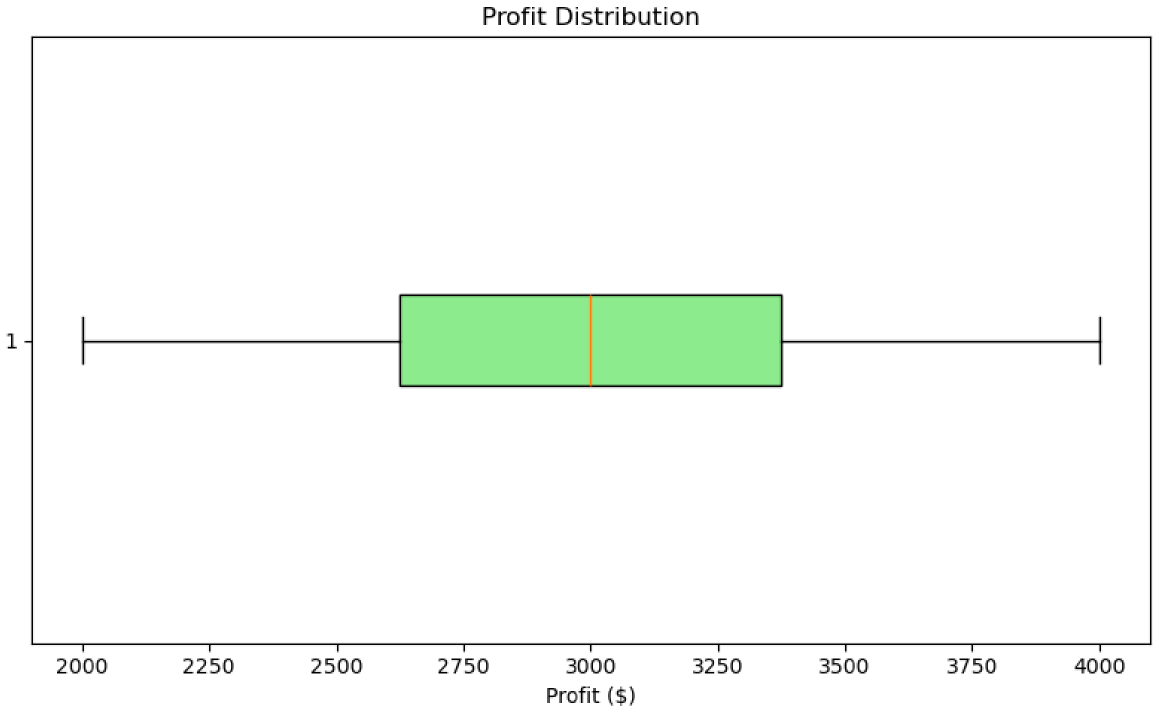
In [11]:

```
# Histogram : Distribution of Saales

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 [12]:

```
# 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 [1]:

```
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='Sale
        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='sky
        plt.bar(df['Month'], df['Profit'], width=width, label='Profit', color='c
        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=
        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", "Histogra
        label="Choose Plot Type"
    ),
    outputs=gr.Plot(label= "Visualization"),
    title= "Sales & Profit Visual Explorere",
    description="Choose a chart type to visualisze 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[1]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:
```