Visualizing Data with Matplotlib and Create Front end by using Gradio

In [1]:

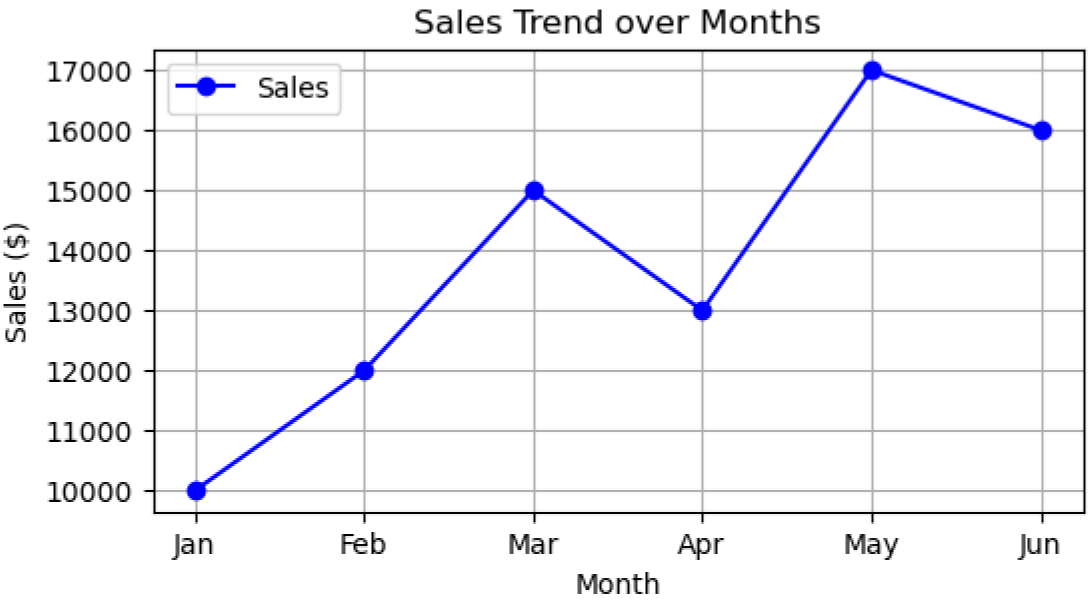
```
#Create the dataframe
import pandas as pd

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

```
  Month   Sale  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
```

Line Plot ---- Monthly Sales

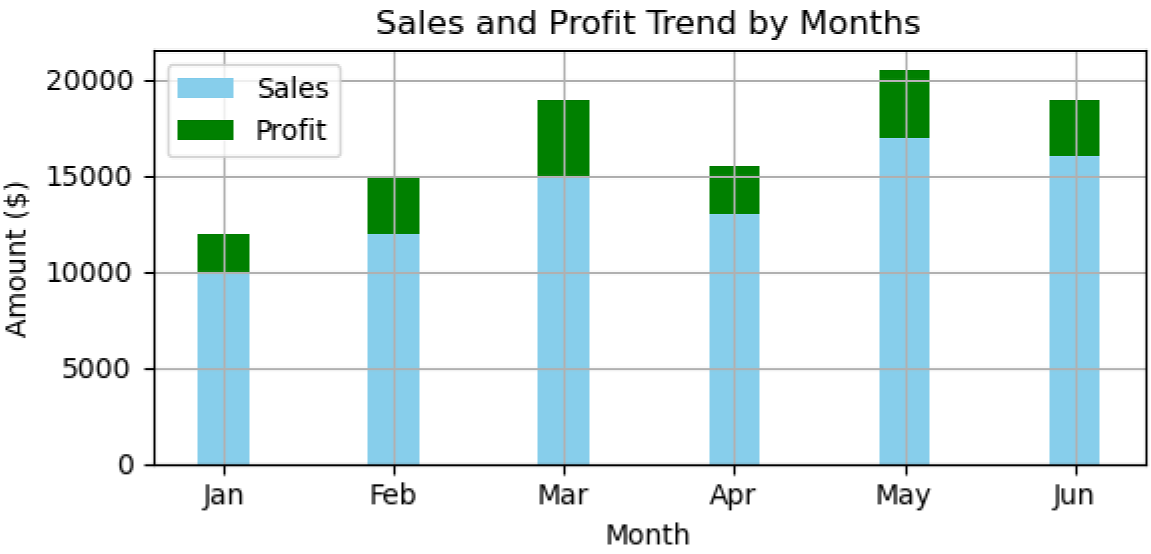
In [2]:

```
import matplotlib.pyplot as plt
#figure width=8 and height=5
plt.figure(figsize=(6,3))
# plot the graph
plt.plot(df['Month'],df['Sale'], color='blue', marker='o', linestyle='-', label='Sa
plt.title('Sales Trend over Months')
plt.xlabel('Month')
plt.ylabel('Sales ($)')
plt.grid(True)
plt.legend()
plt.show()
```



Stacked Bar Chart

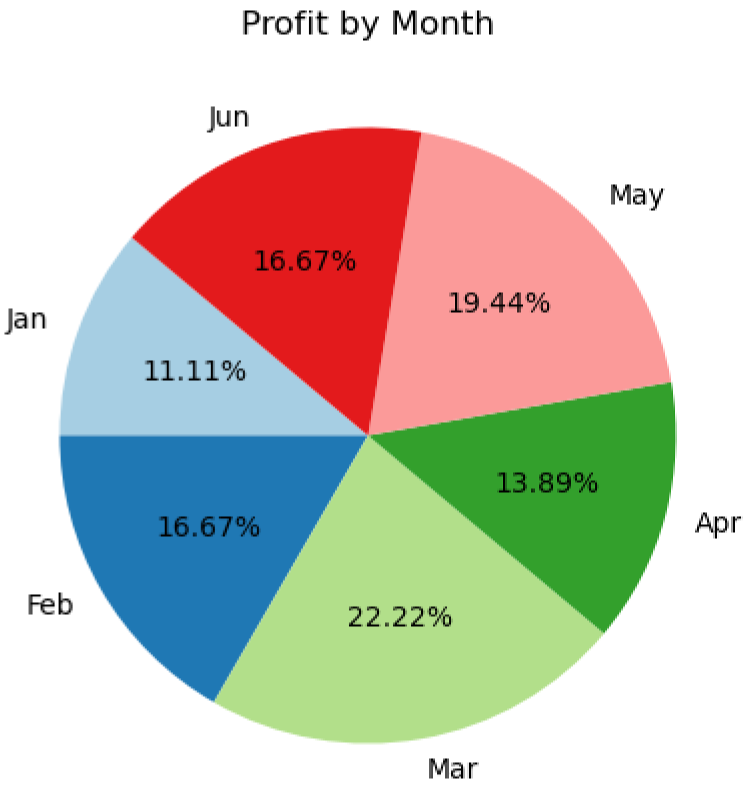
In [3]:
```
plt.figure(figsize=(6,3))
width=0.3
plt.bar(df['Month'],df['Sale'], width=width, color='skyblue', label='Sales')
plt.bar(df['Month'],df['Profit'], width=width, color='green', label='Profit', botto
plt.title('Sales and Profit Trend by Months')
plt.xlabel('Month')
plt.ylabel('Amount ($)')
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```



Pie Chart --- Prifit by Month

In [4]:

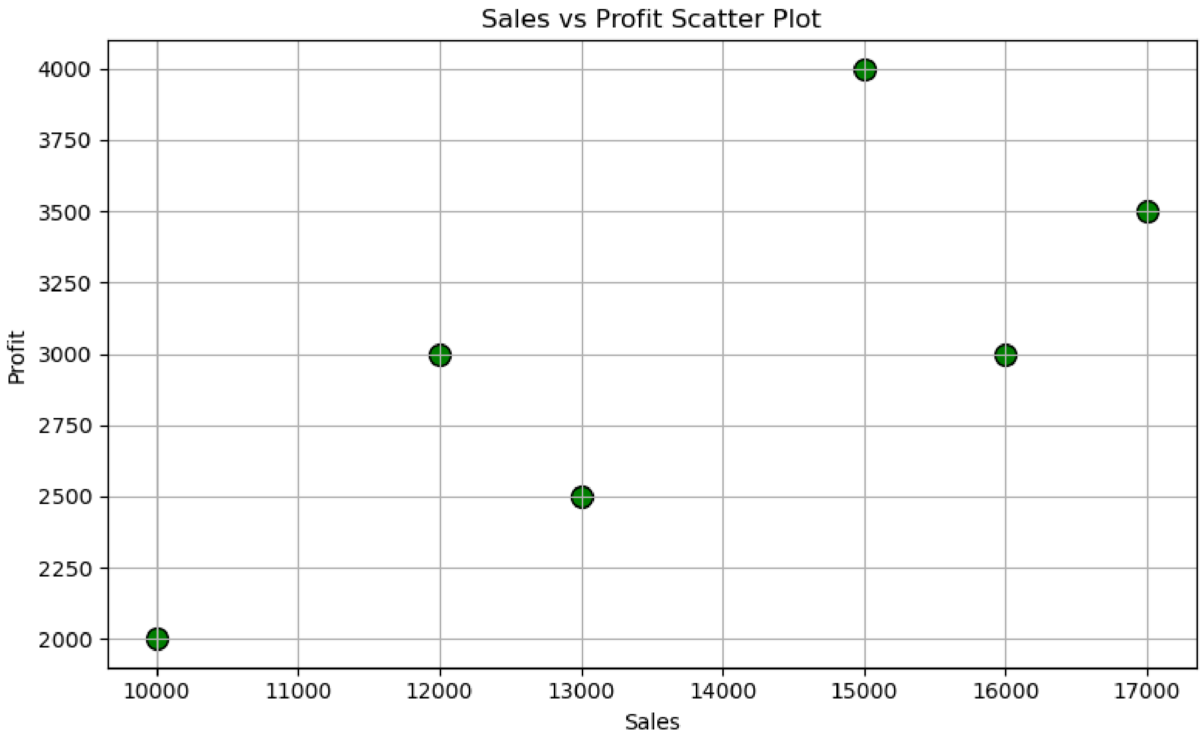
```
plt.figure(figsize=(10,5))
plt.pie(df['Profit'], labels=df['Month'], autopct='%1.2f%%', startangle=140, colors
plt.title('Profit by Month')
plt.show()
```



Scatter plot

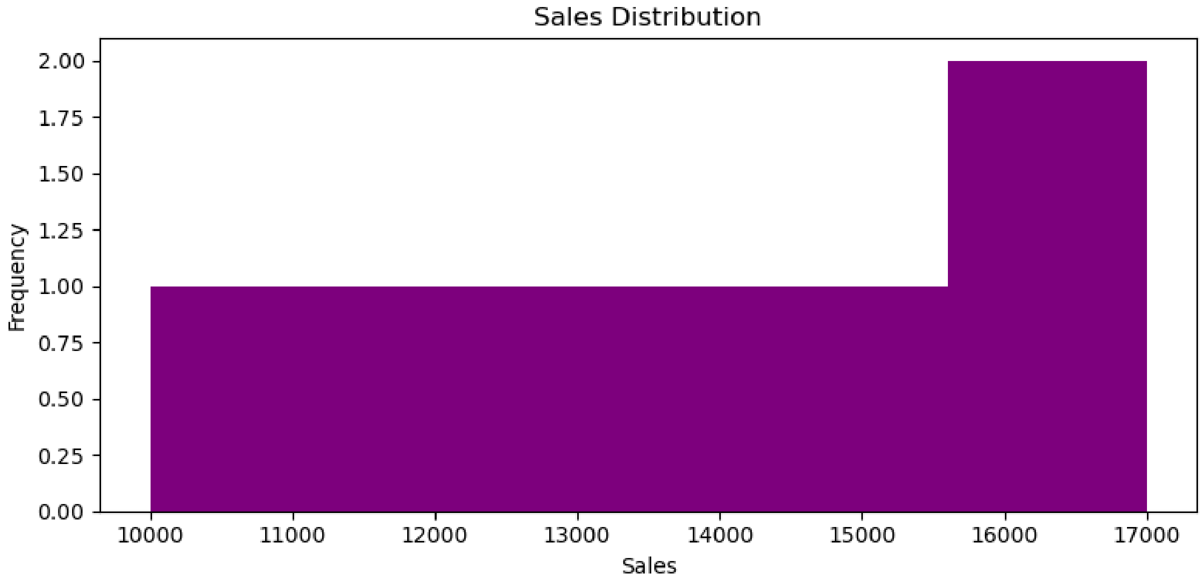
In [5]:

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



histogram

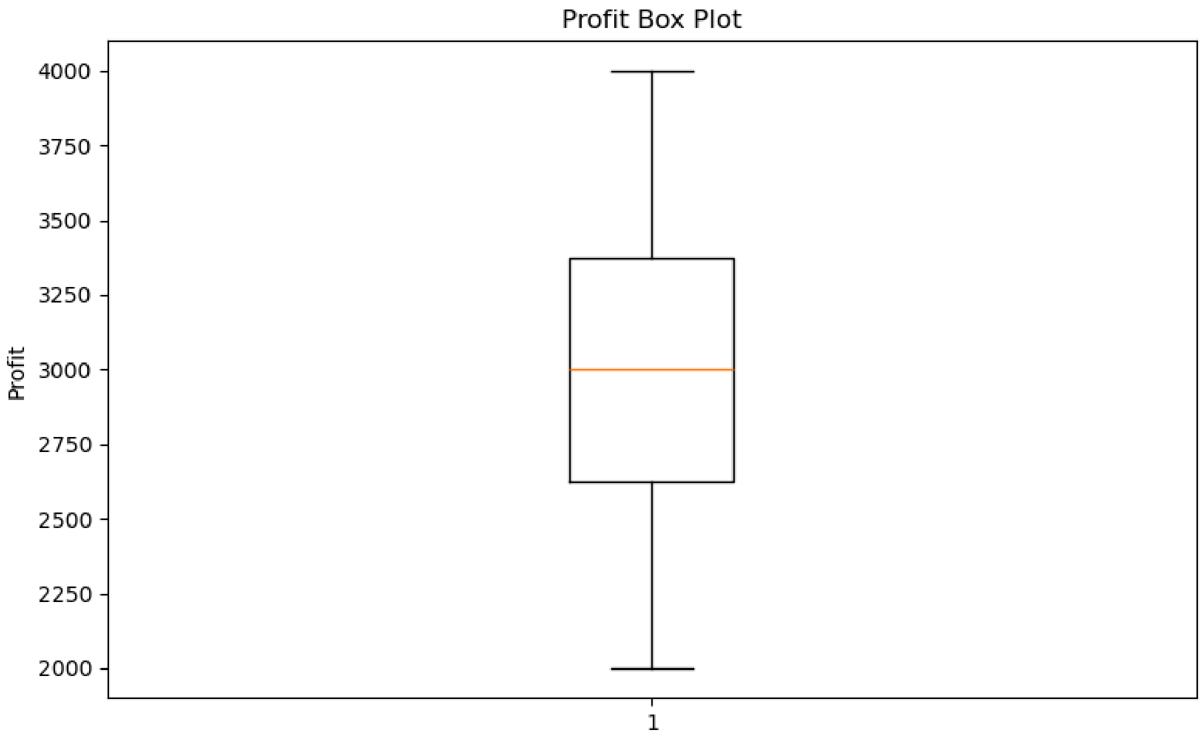
In [6]:
```
plt.figure(figsize=(8,4))
plt.hist(df['Sale'], bins=5, color='purple')
plt.title('Sales Distribution')
plt.xlabel('Sales')
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```



Box Plot

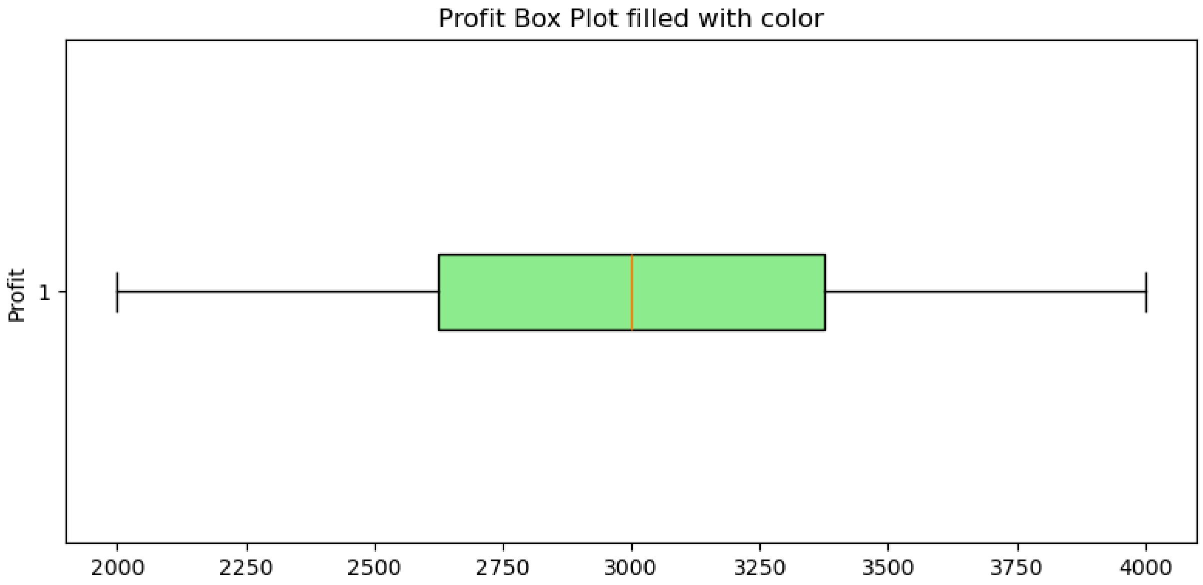
In [7]:

```
plt.figure(figsize=(8,5))
plt.boxplot(df['Profit'])
plt.title('Profit Box Plot')
plt.ylabel('Profit')
plt.tight_layout()
plt.show()
```



In [8]:

```
plt.figure(figsize=(8,4))
plt.boxplot(df['Profit'],vert=False, patch_artist=True, boxprops=dict(facecolor='li
plt.title('Profit Box Plot filled with color')
plt.ylabel('Profit')
plt.tight_layout()
plt.show()
```



install gradio

Gradio is an open-source Python library that allows developers to quickly build user interfaces (UIs) for machine learning models, APIs, and other Python functions

In [9]:
```
#!pip install gradio
```

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

data = {
    "Month": ['Jan','Feb','Mar','Apr','May','Jun'],
    "Sale" : [10000, 12000, 15000, 13000,17000, 16000],
    "Profit":[2000,3000,4000,2500,3500,3000]
}
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['Sale'], color='blue', marker='o', linestyle='--',
        plt.title('Sales Trend over Months')
        plt.xlabel('Month')
        plt.ylabel('Sales ($)')
        plt.grid(True)
        plt.legend()
    elif plot_type=='stacked bar chart':
        fig.set_size_inches(10,6)
        width=0.3
        plt.bar(df['Month'],df['Sale'], width=width, color='skyblue', label='Sales'
        plt.bar(df['Month'],df['Profit'], width=width, color='green', label='Profit
        plt.title('Sales and Profit Trend by Months')
        plt.xlabel('Month')
        plt.ylabel('Amount ($)')
        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
        plt.title('Profit by Month')
    elif plot_type=='Scatter Plot':
        plt.scatter(df['Sale'], df['Profit'],color='green', s=100, edgecolors='blac
        plt.title('Sales vs Profit Scatter Plot')
        plt.xlabel('Sales')
        plt.ylabel('Profit')
        plt.grid(True)
    elif plot_type=='Histogram':
        plt.hist(df['Sale'], bins=5, color='purple')
        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(facec
        plt.title('Profit Box Plot filled with color')
        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',
                    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')

demo.launch()
```

```
* Running on local URL:  http://127.0.0.1:7860
* To create a public link, set `share=True` in `launch()`.
```

Out[10]:

In [ ]: