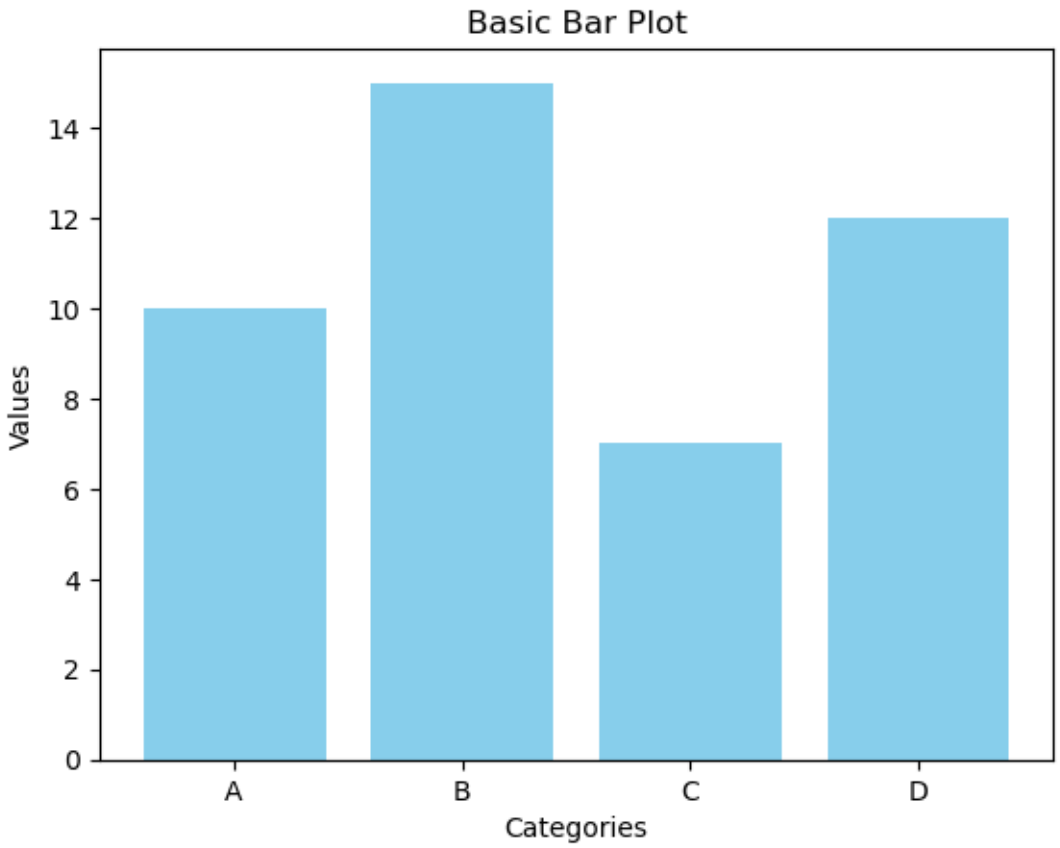
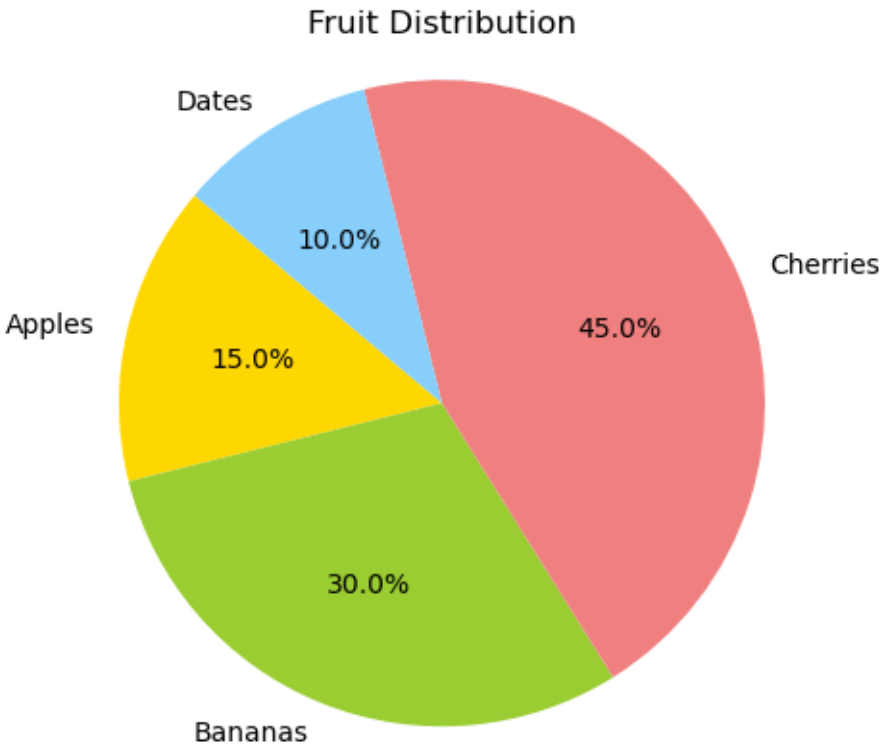
matplotlib

September 16, 2024

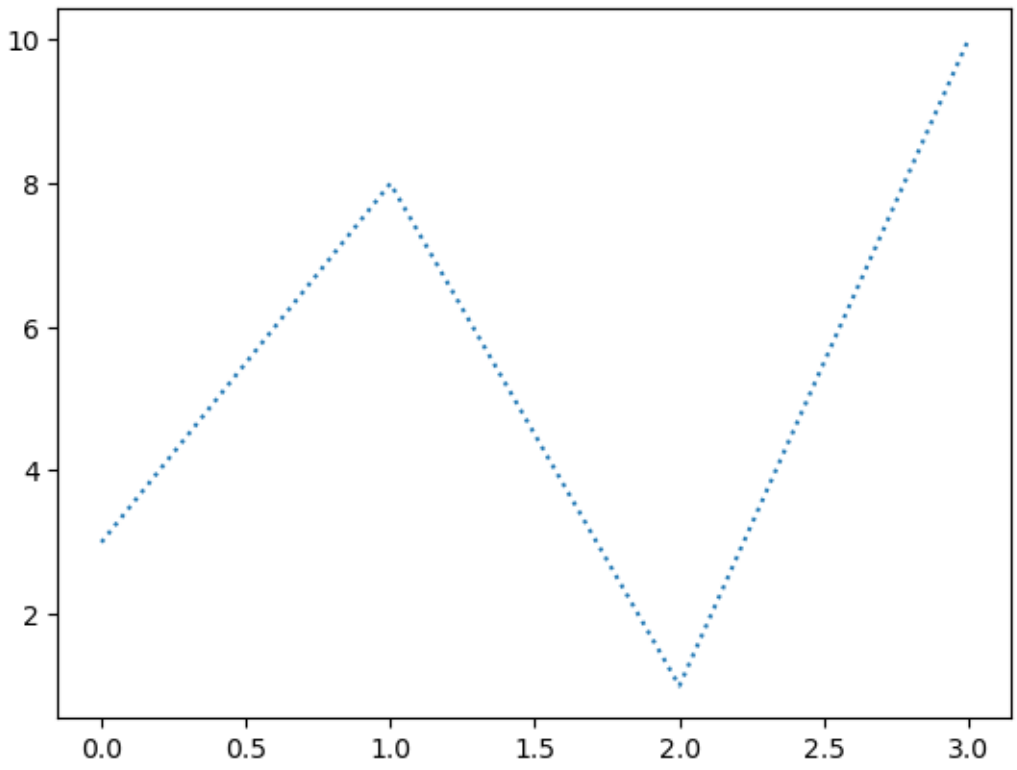
[1]:
```
import matplotlib.pyplot as plt

# Data
x = ['A', 'B', 'C', 'D']
y = [10, 15, 7, 12]

# Create a bar plot
plt.bar(x, y, color='skyblue')

# Add title and labels
plt.title('Basic Bar Plot')
plt.xlabel('Categories')
plt.ylabel('Values')

# Show the plot
plt.show()
```



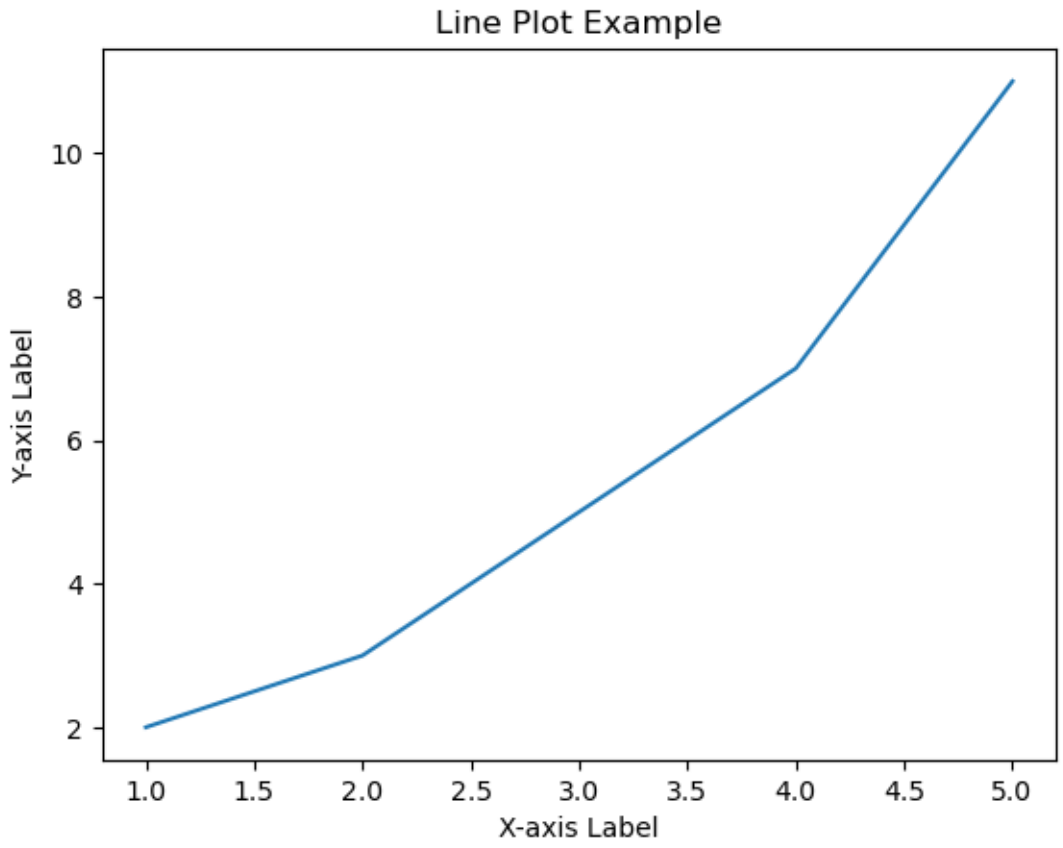
[2]:
```
import matplotlib.pyplot as plt

# Data
sizes = [15, 30, 45, 10]
labels = ['Apples', 'Bananas', 'Cherries', 'Dates']
colors = ['gold', 'yellowgreen', 'lightcoral', 'lightskyblue']

# Create a pie chart
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=140)

# Add title
plt.title('Fruit Distribution')

# Equal aspect ratio ensures that pie is drawn as a circle.
plt.axis('equal')

# Show the plot
plt.show()
```



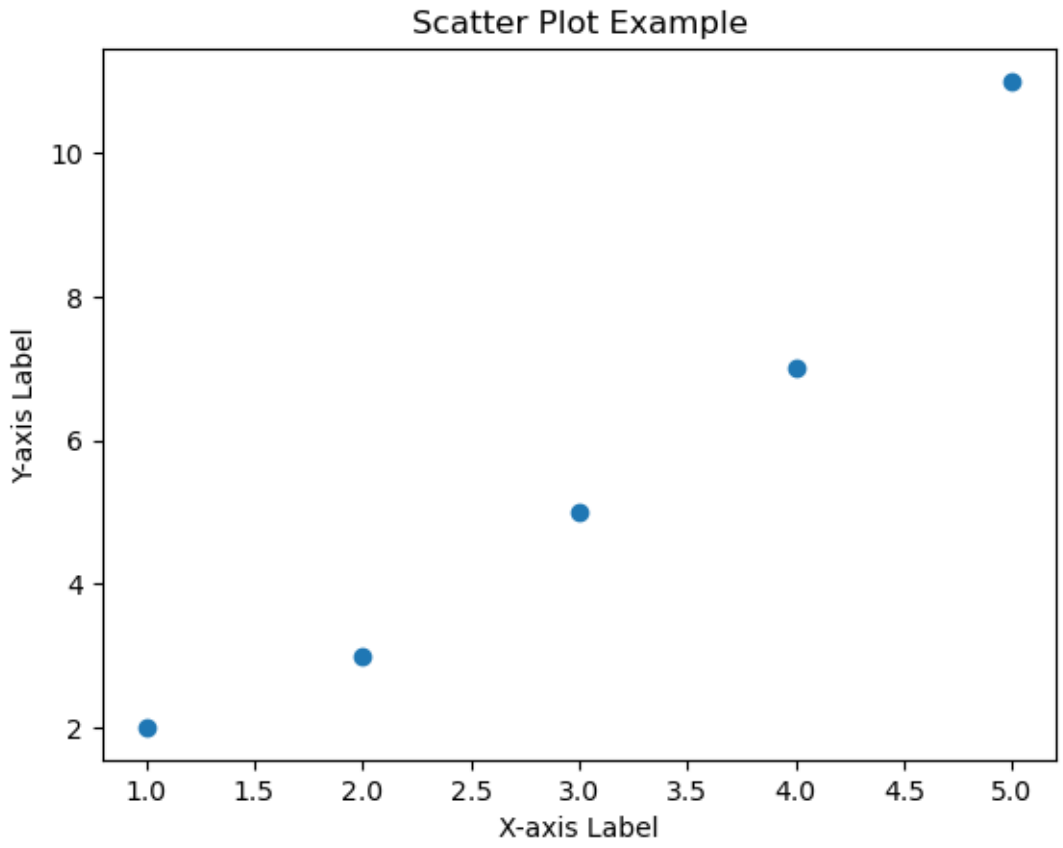
[3]:
```
import matplotlib.pyplot as plt
import numpy as np

ypoints = np.array([3, 8, 1, 10])

plt.plot(ypoints, linestyle = 'dotted')
plt.show()
```



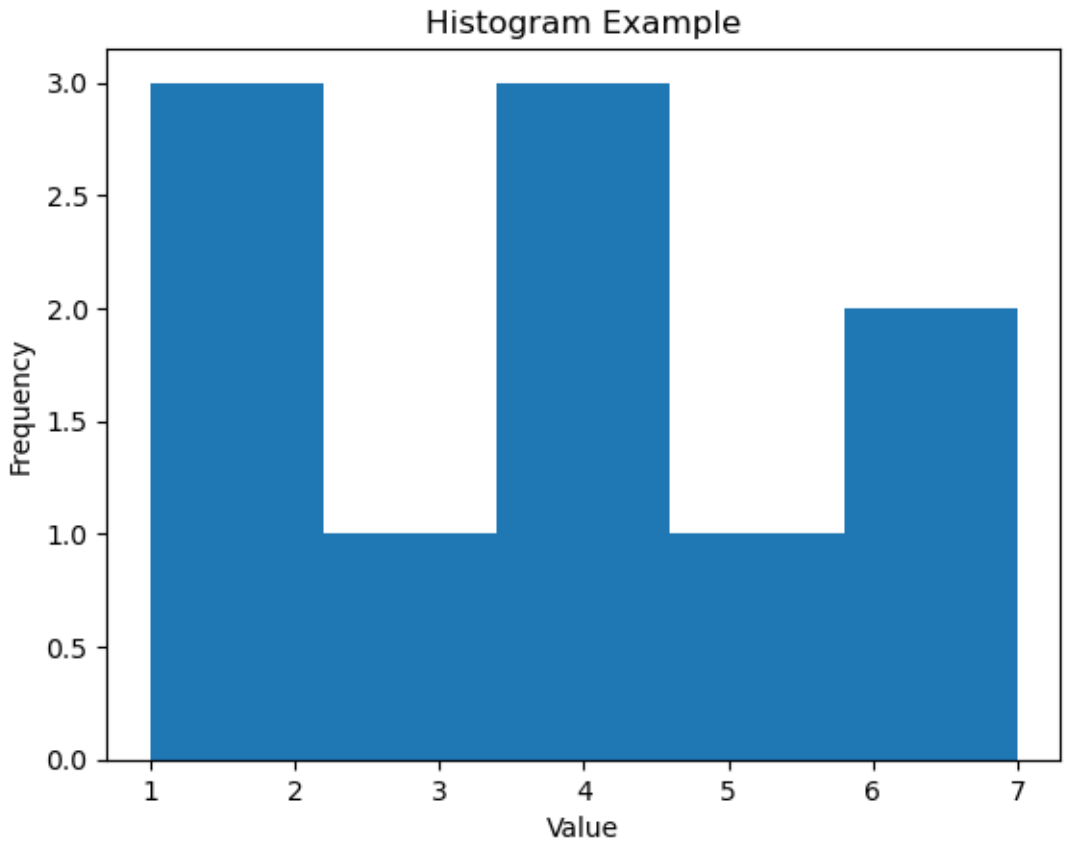
[4]:
```
import matplotlib.pyplot as plt

# Sample data
x = [1, 2, 3, 4, 5]
y = [2, 3, 5, 7, 11]

# Create a line plot
plt.plot(x, y)

# Add title and labels
plt.title('Line Plot Example')
plt.xlabel('X-axis Label')
plt.ylabel('Y-axis Label')

# Show the plot
plt.show()
```



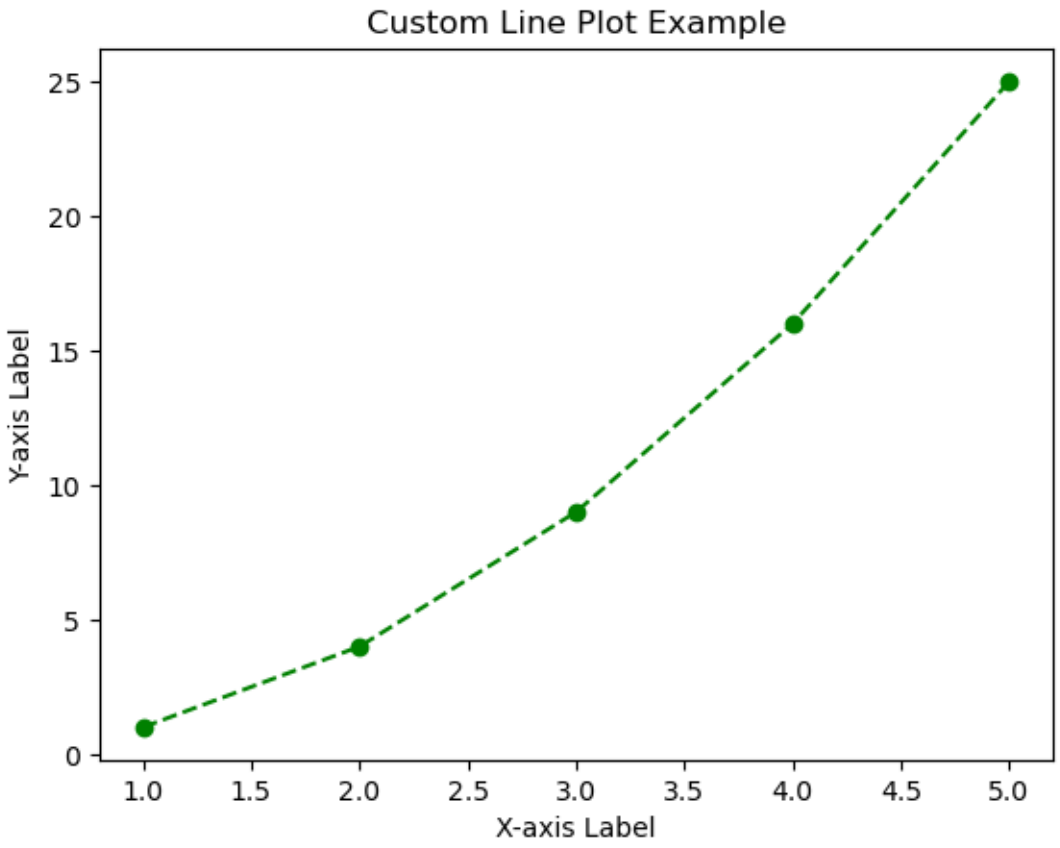
[5]:
```
import matplotlib.pyplot as plt

# Sample data
x = [1, 2, 3, 4, 5]
y = [2, 3, 5, 7, 11]

# Create a scatter plot
plt.scatter(x, y)

# Add title and labels
plt.title('Scatter Plot Example')
plt.xlabel('X-axis Label')
plt.ylabel('Y-axis Label')

# Show the plot
plt.show()
```



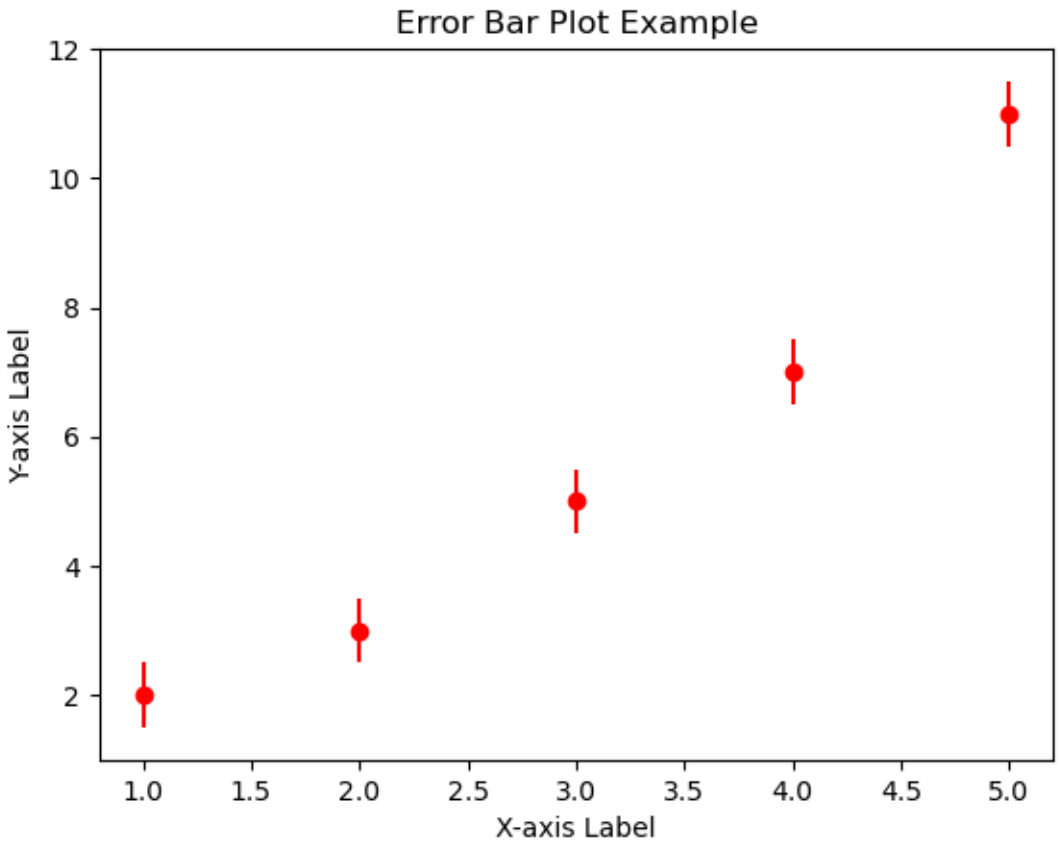
[6]:
```python
import matplotlib.pyplot as plt

# Sample data
data = [1, 2, 2, 3, 4, 4, 4, 5, 6, 7]

# Create a histogram
plt.hist(data, bins=5)

# Add title and labels
plt.title('Histogram Example')
plt.xlabel('Value')
plt.ylabel('Frequency')

# Show the plot
plt.show()
```



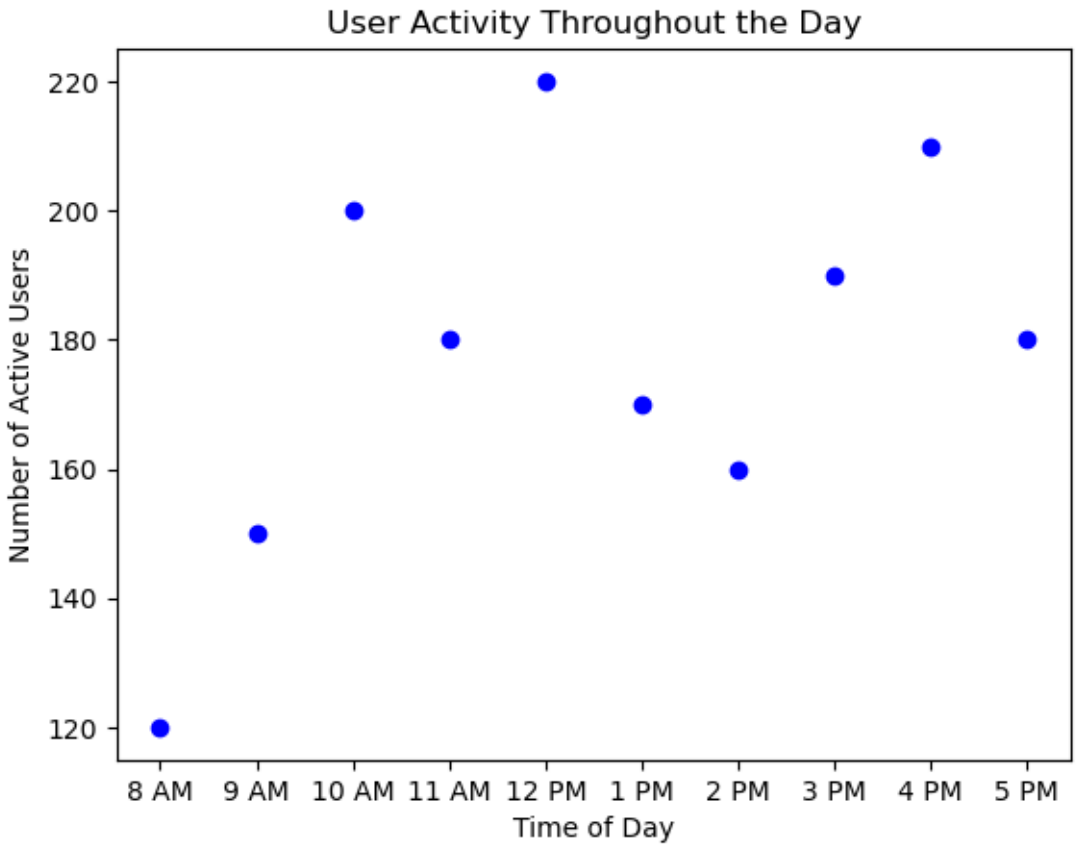
[7]:
```python
import matplotlib.pyplot as plt

# Sample data
x = [1, 2, 3, 4, 5]
y = [1, 4, 9, 16, 25]

# Create a plot with custom color and line style
plt.plot(x, y, color='green', linestyle='--', marker='o')

# Add title and labels
plt.title('Custom Line Plot Example')
plt.xlabel('X-axis Label')
plt.ylabel('Y-axis Label')

# Show the plot
plt.show()
```



[8]:
```python
import matplotlib.pyplot as plt

# Sample data
x = [1, 2, 3, 4, 5]
y = [2, 3, 5, 7, 11]
errors = [0.5, 0.5, 0.5, 0.5, 0.5]

# Create a plot with error bars
plt.errorbar(x, y, yerr=errors, fmt='o', color='red')

# Add title and labels
plt.title('Error Bar Plot Example')
plt.xlabel('X-axis Label')
plt.ylabel('Y-axis Label')

# Show the plot
plt.show()
```

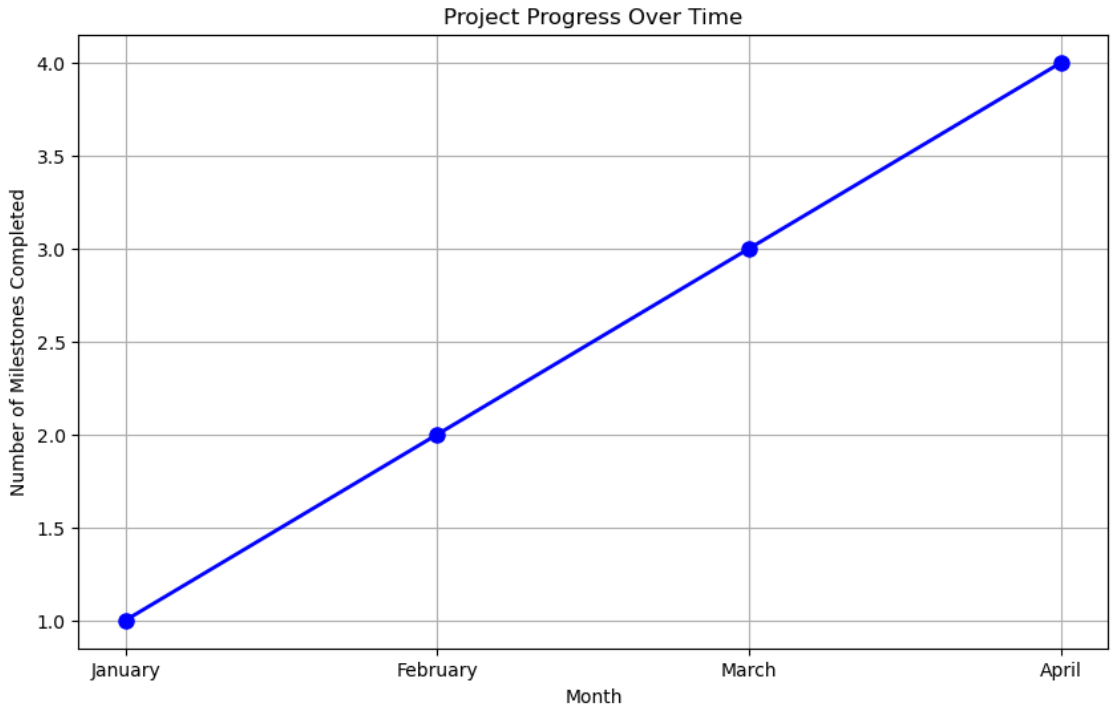


[9]:
```
import matplotlib.pyplot as plt

# Sample data: Time of day vs. number of users active
times_of_day = ['8 AM', '9 AM', '10 AM', '11 AM', '12 PM', '1 PM', '2 PM', '3␣
 ↪PM', '4 PM', '5 PM']
active_users = [120, 150, 200, 180, 220, 170, 160, 190, 210, 180]

# Create a scatter plot
plt.scatter(times_of_day, active_users, color='blue', marker='o')

# Add title and labels
plt.title('User Activity Throughout the Day')
plt.xlabel('Time of Day')
plt.ylabel('Number of Active Users')



# Show the plot
plt.show()
```



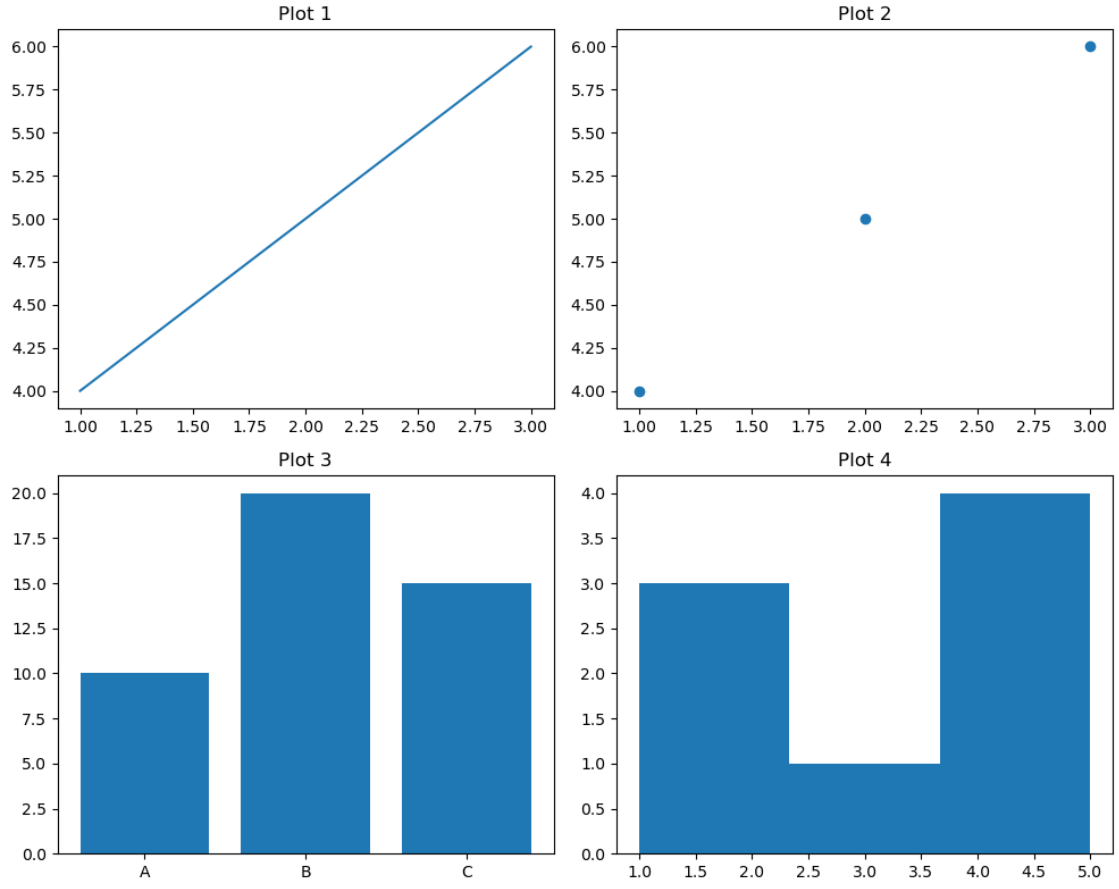
[10]:
```
import matplotlib.pyplot as plt

# Sample data
months = ['January', 'February', 'March', 'April']
milestones_completed = [1, 2, 3, 4]  # Assume one milestone completed each month

# Create the line chart
plt.figure(figsize=(10, 6))
plt.plot(months, milestones_completed, marker='o', color='b', linestyle='-',␣
 ↪linewidth=2, markersize=8)

# Add title and labels
plt.title('Project Progress Over Time')
plt.xlabel('Month')
plt.ylabel('Number of Milestones Completed')

# Display the grid
plt.grid(True)

# Show the plot
```

```
plt.show()
```



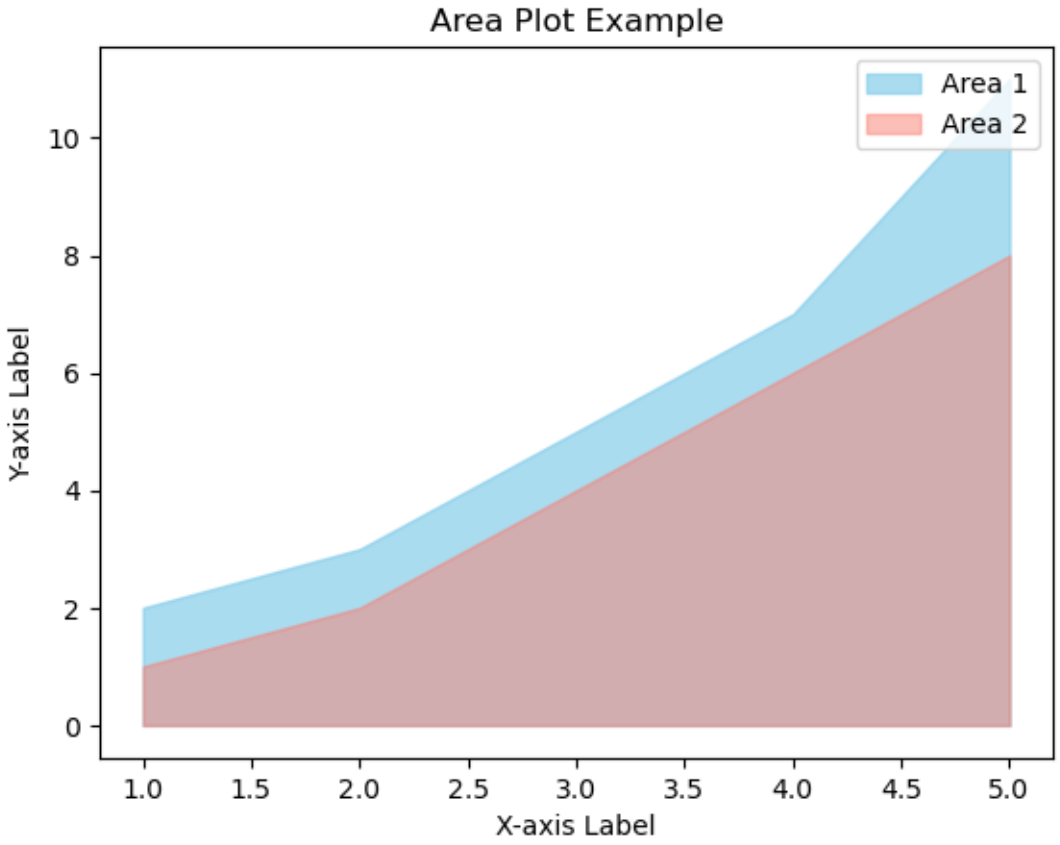
[31]:
```
import matplotlib.pyplot as plt

# Create subplots
fig, axs = plt.subplots(2, 2, figsize=(10, 8))

# Plot in the first subplot
axs[0, 0].plot([1, 2, 3], [4, 5, 6])
axs[0, 0].set_title('Plot 1')

# Plot in the second subplot
axs[0, 1].scatter([1, 2, 3], [4, 5, 6])
axs[0, 1].set_title('Plot 2')

# Plot in the third subplot
axs[1, 0].bar(['A', 'B', 'C'], [10, 20, 15])
axs[1, 0].set_title('Plot 3')

# Plot in the fourth subplot
axs[1, 1].hist([1, 2, 2, 3, 4, 4, 4, 5], bins=3)
axs[1, 1].set_title('Plot 4')

# Adjust layout
```

```
plt.tight_layout()

# Show the plot
plt.show()
```



[16]:
```
import matplotlib.pyplot as plt

# Sample data
x = [1, 2, 3, 4, 5]
y1 = [2, 3, 5, 7, 11]
y2 = [1, 2, 4, 6, 8]

# Create an area plot
plt.fill_between(x, y1, color='skyblue', alpha=0.7, label='Area 1')
plt.fill_between(x, y2, color='salmon', alpha=0.5, label='Area 2')

# Add title and labels
plt.title('Area Plot Example')
plt.xlabel('X-axis Label')
```

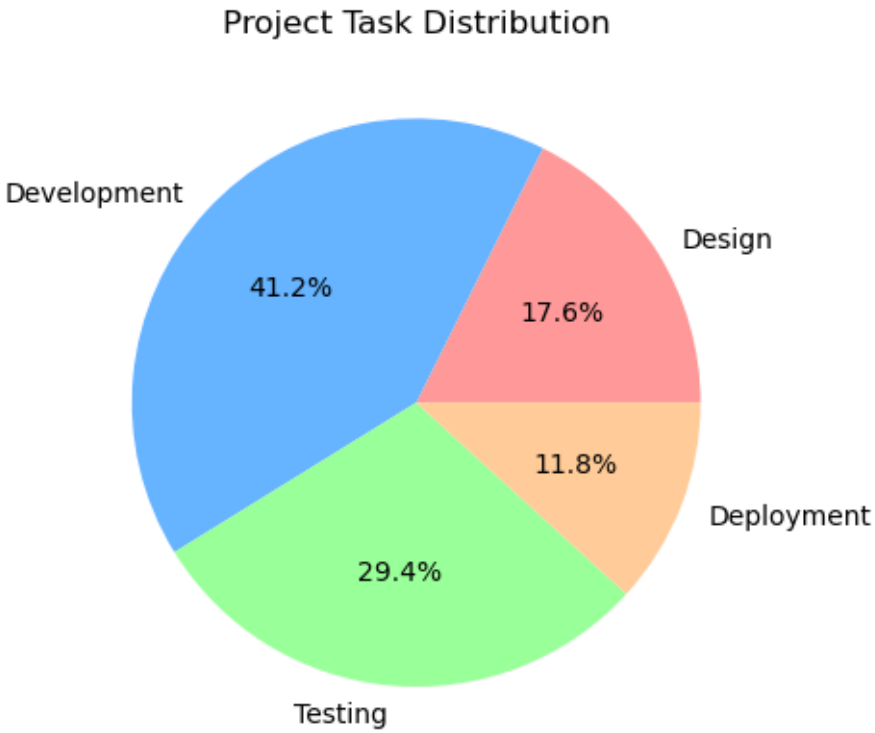
```
plt.ylabel('Y-axis Label')
plt.legend()

# Show the plot
plt.show()
```



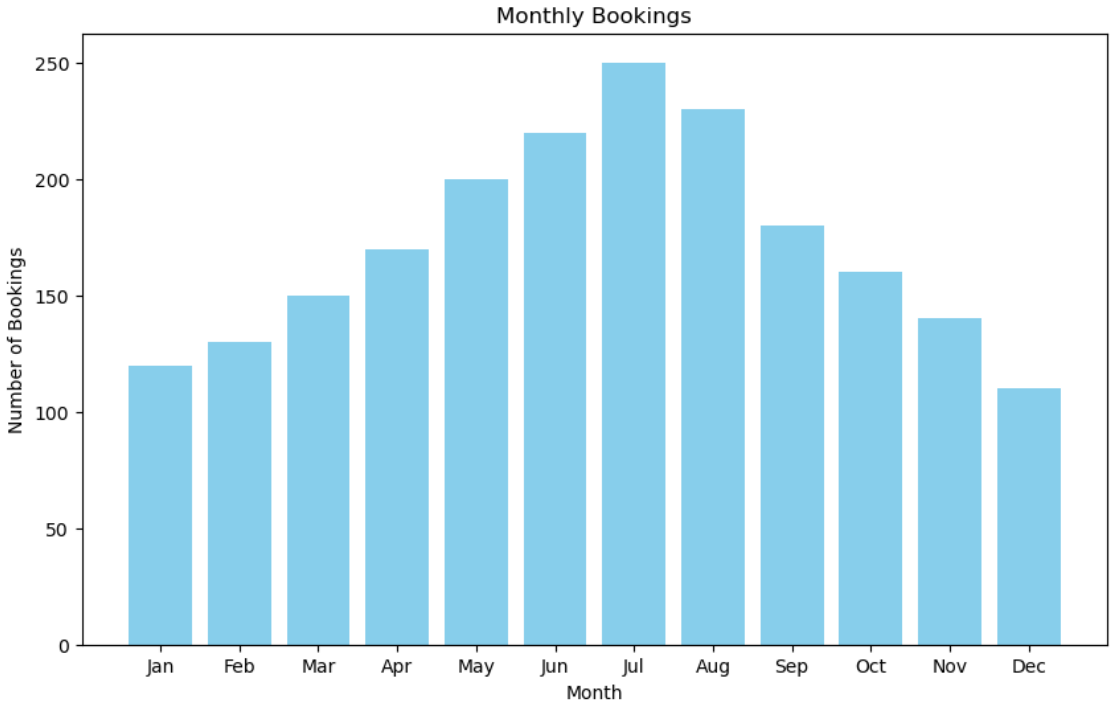
[ ]:

[19]:
```
import matplotlib.pyplot as plt

# Sample data: Task distribution in a project
tasks = ['Design', 'Development', 'Testing', 'Deployment']
hours_spent = [30, 70, 50, 20]

# Create a pie chart
plt.pie(hours_spent, labels=tasks, autopct='%1.1f%%',␣
  ↪colors=['#ff9999','#66b3ff','#99ff99','#ffcc99'])

# Add title
plt.title('Project Task Distribution')
```

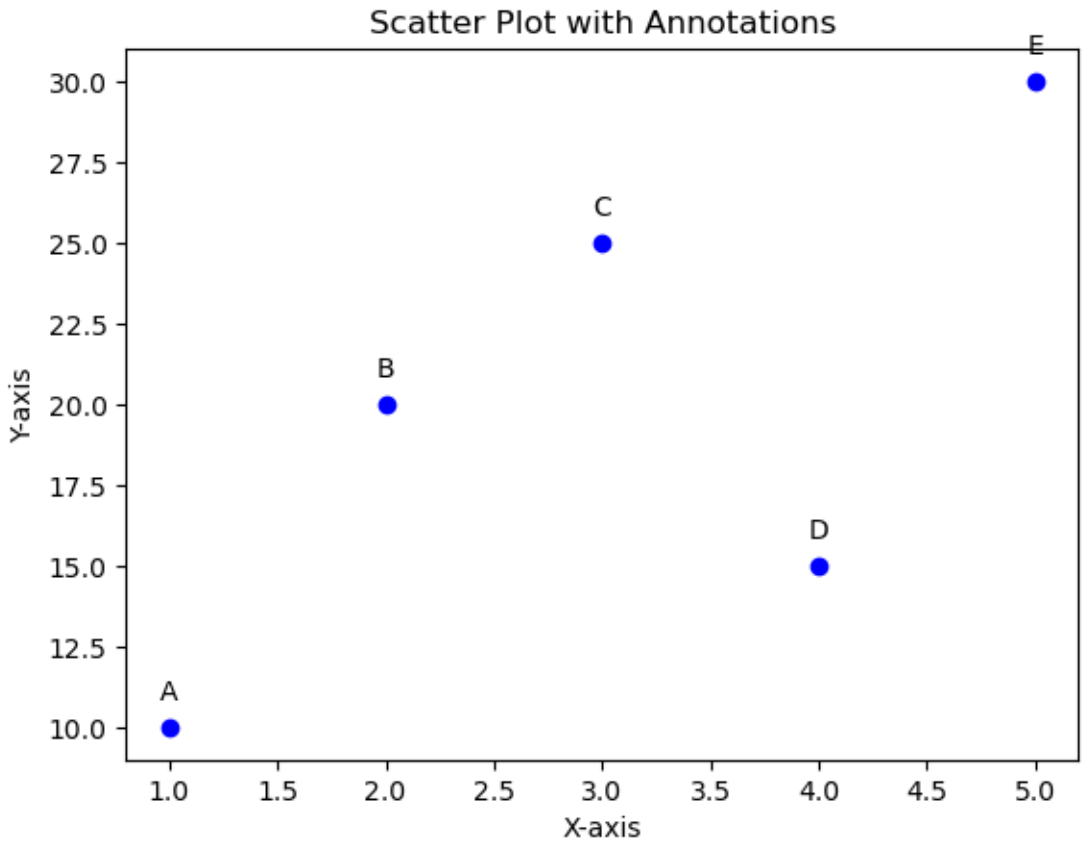
```
# Show the plot
plt.show()
```



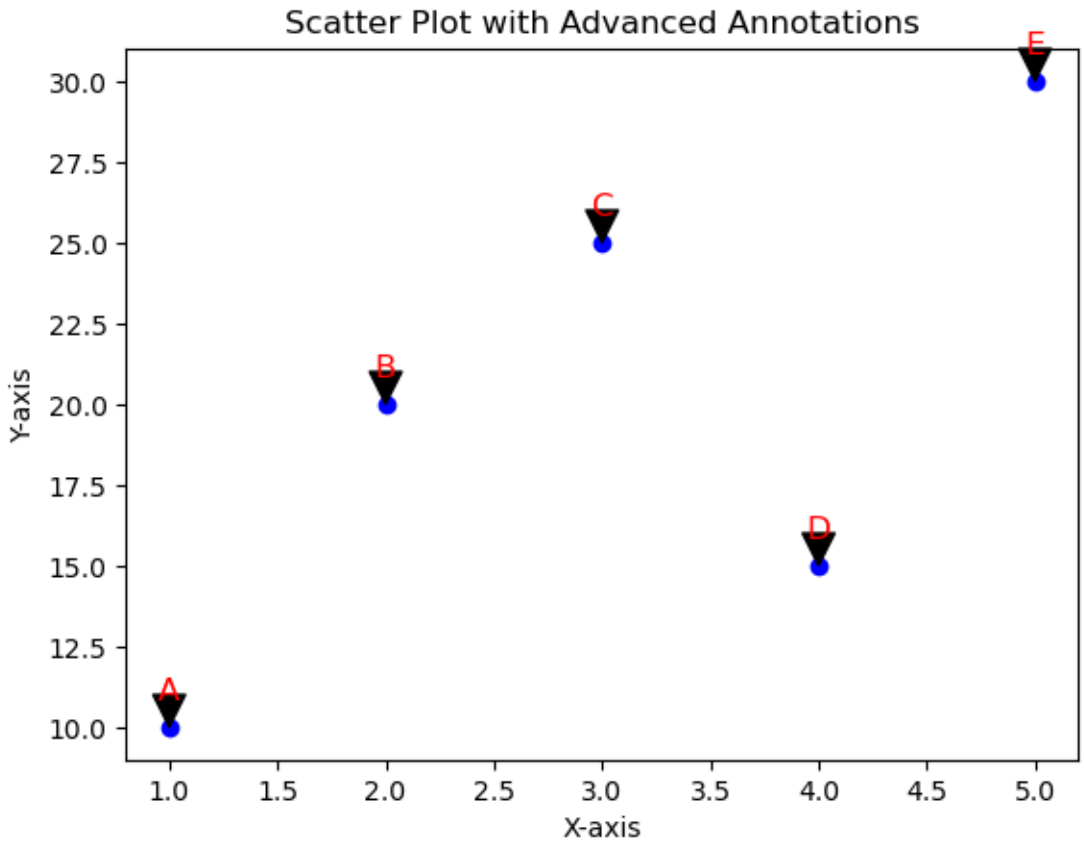
[20]:
```
import matplotlib.pyplot as plt

# Sample data: Number of bookings per month
months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct',␣
 ↪'Nov', 'Dec']
bookings = [120, 130, 150, 170, 200, 220, 250, 230, 180, 160, 140, 110]

# Create a bar chart
plt.figure(figsize=(10, 6))
plt.bar(months, bookings, color='skyblue')
plt.title('Monthly Bookings')
plt.xlabel('Month')
plt.ylabel('Number of Bookings')
plt.show()
```



[21]:
```
import matplotlib.pyplot as plt

# Sample data
x = [1, 2, 3, 4, 5]
y = [10, 20, 25, 15, 30]
labels = ['A', 'B', 'C', 'D', 'E']

# Create a scatter plot
plt.scatter(x, y, color='blue')

# Annotate each point with a label
for i in range(len(x)):
    plt.annotate(labels[i], (x[i], y[i]), textcoords="offset points",␣
 ↪xytext=(0,10), ha='center')

# Add title and labels
plt.title('Scatter Plot with Annotations')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')

# Show the plot
plt.show()
```

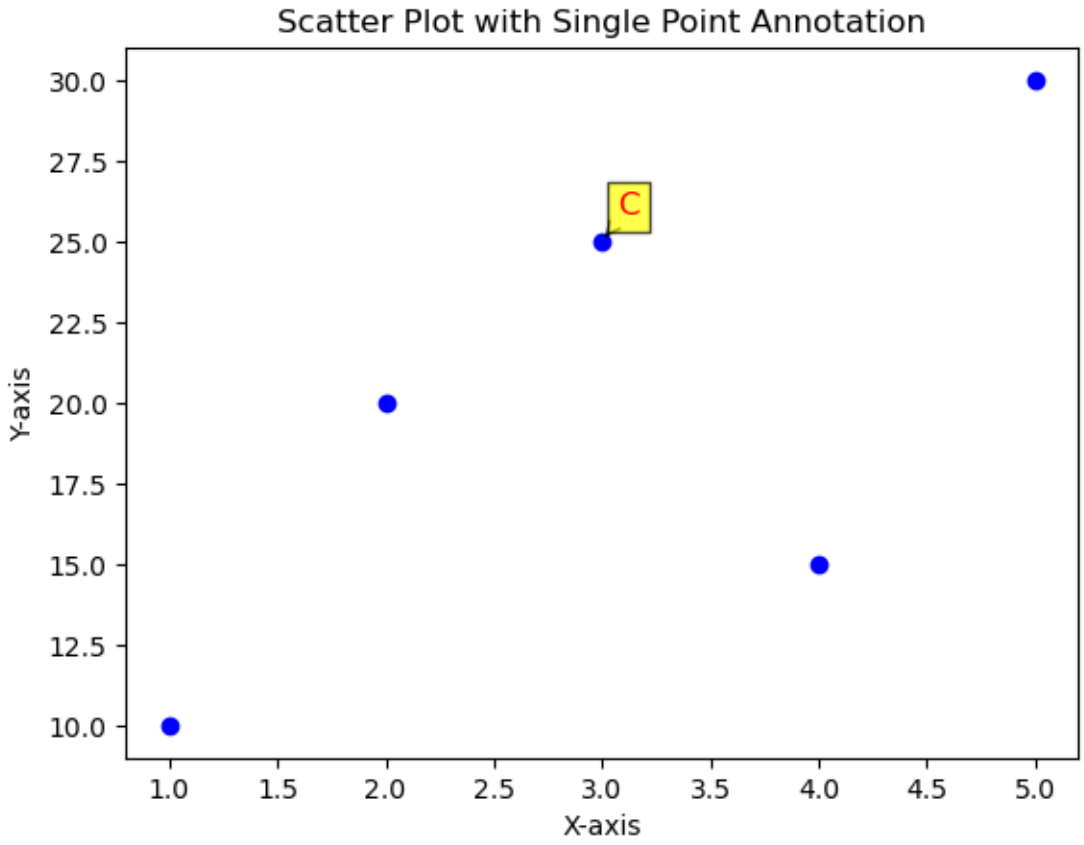


[23]:
```
plt.scatter(x, y, color='blue')

for i in range(len(x)):
    plt.annotate(labels[i],
                 (x[i], y[i]),
                 textcoords="offset points",
                 xytext=(0,10),
                 ha='center',
                 fontsize=12,
                 color='red',
                 arrowprops=dict(facecolor='black', shrink=0.02))

plt.title('Scatter Plot with Advanced Annotations')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.show()
```



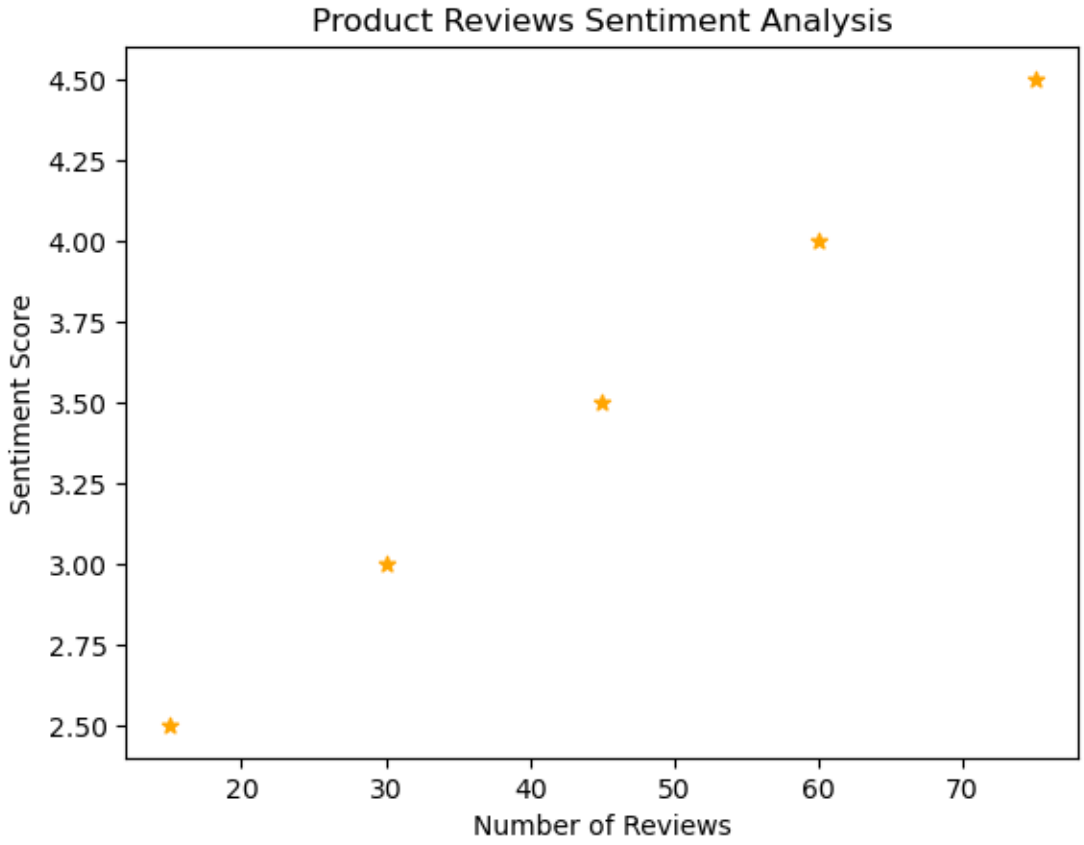
[26]:
```python
import matplotlib.pyplot as plt

# Sample data
x = [1, 2, 3, 4, 5]
y = [10, 20, 25, 15, 30]
labels = ['A', 'B', 'C', 'D', 'E']

# Create a scatter plot
plt.scatter(x, y, color='blue')

# Annotate the point with the label 'C'
point_index = 2  # Index of the point to annotate
plt.annotate(labels[point_index],
             (x[point_index], y[point_index]),
             textcoords="offset points",
             xytext=(10,10),
             ha='center',
             fontsize=12,
             color='red',
             bbox=dict(facecolor='yellow', alpha=0.7),
```

```
            arrowprops=dict(facecolor='black', arrowstyle='->'))

# Add title and labels
plt.title('Scatter Plot with Single Point Annotation')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')

# Show the plot
plt.show()
```



[30]:
```
import matplotlib.pyplot as plt

# Sample data: Number of reviews vs. sentiment score
num_reviews = [15, 30, 45, 60, 75]
sentiment_scores = [2.5, 3.0, 3.5, 4.0, 4.5]

# Create a scatter plot
plt.scatter(num_reviews, sentiment_scores, color='orange', marker='*')

# Add title and labels
```

```
plt.title('Product Reviews Sentiment Analysis')
plt.xlabel('Number of Reviews')
plt.ylabel('Sentiment Score')

# Show the plot
plt.show()
```



[ ]: