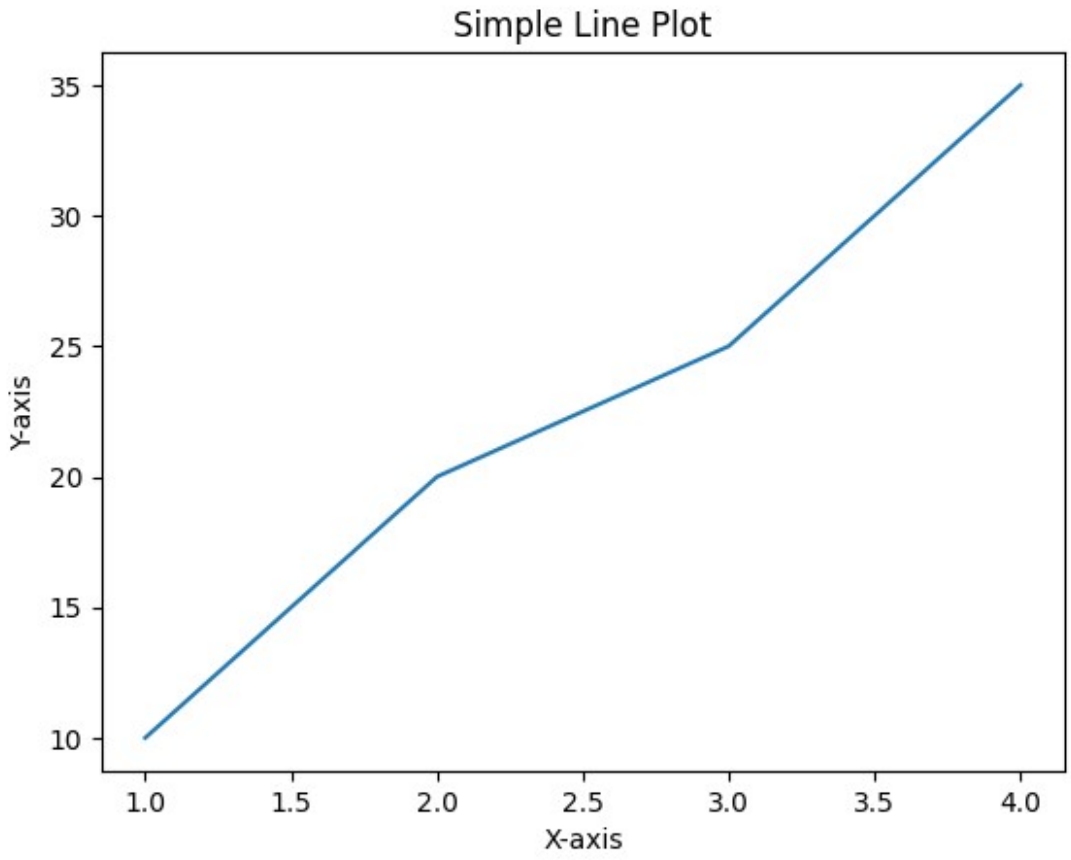
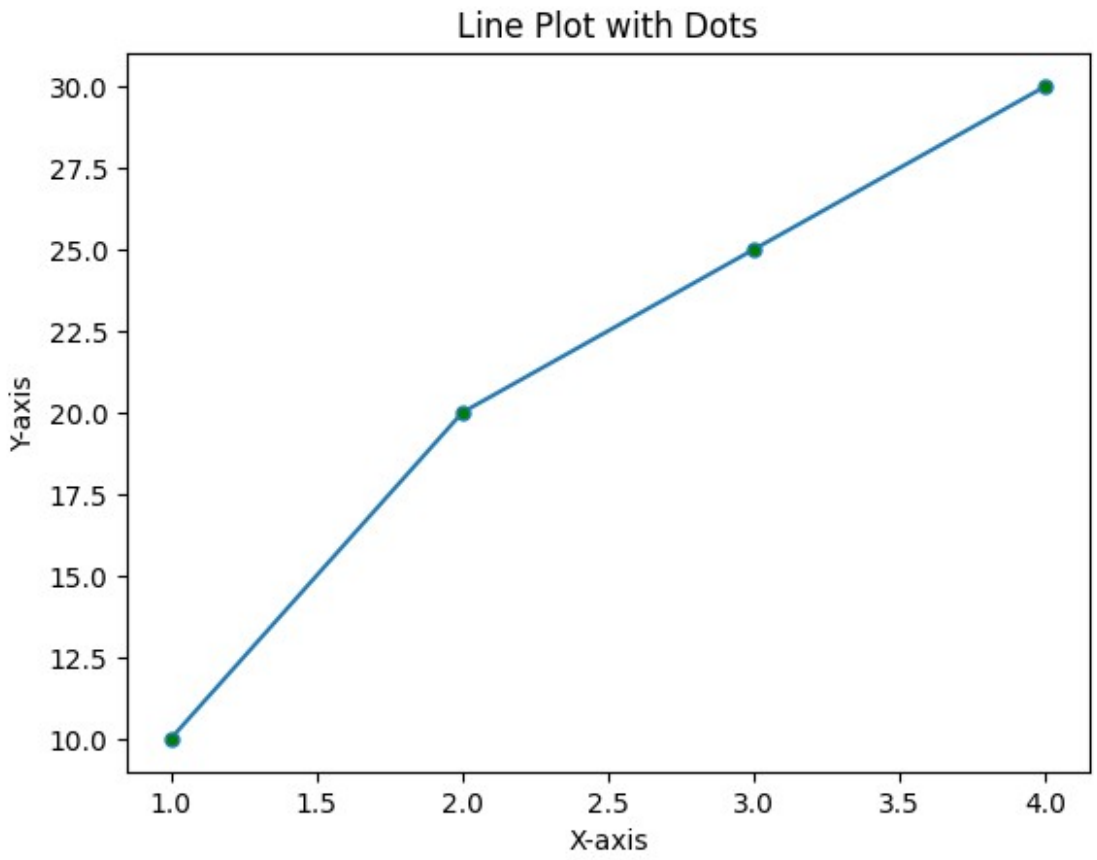
```python
import matplotlib.pyplot as plt

x = [1, 2, 3, 4]
y = [10, 20, 25, 35]

plt.plot(x, y)
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Simple Line Plot')
plt.show()
```

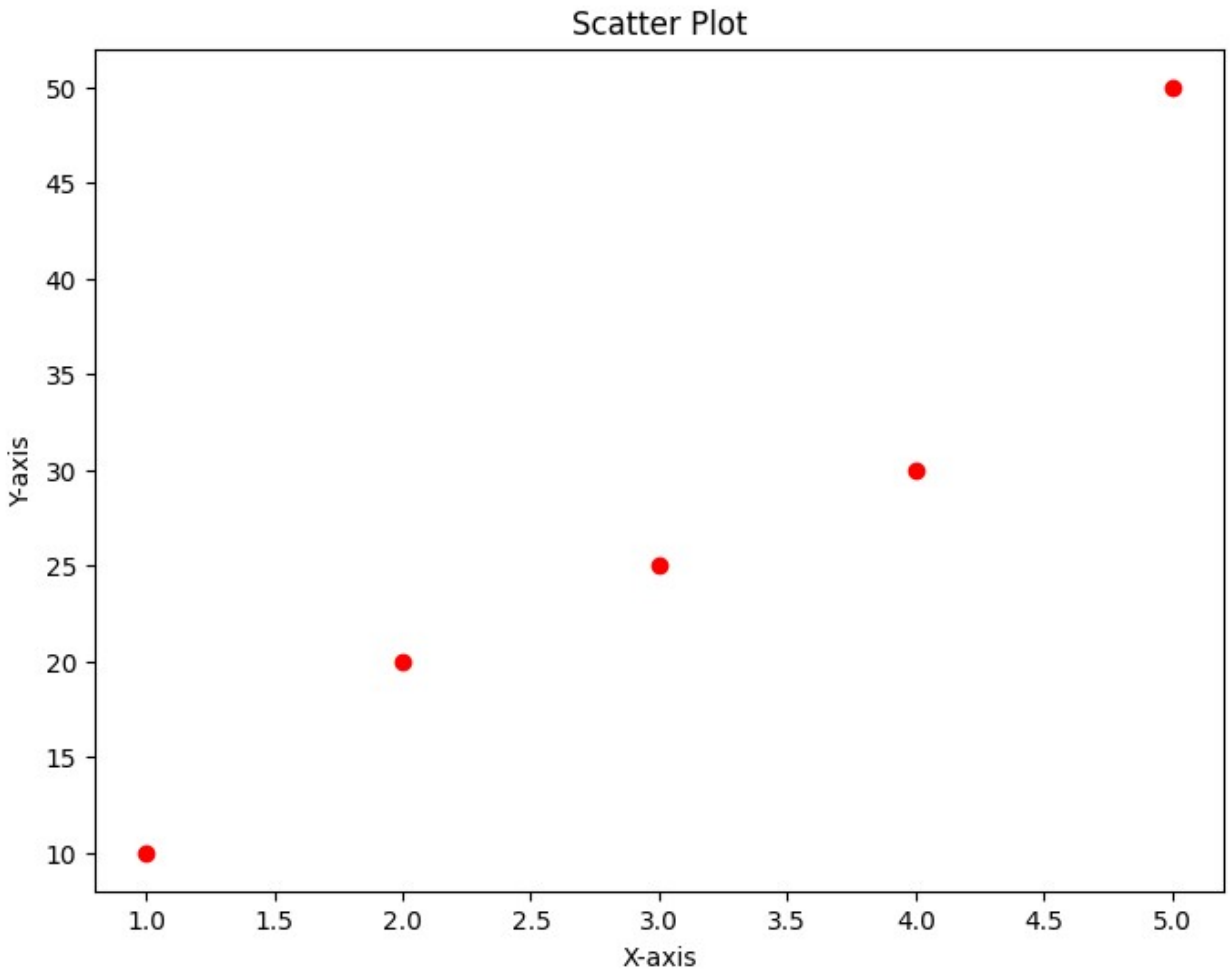



```python
import matplotlib.pyplot as plt

x = [1, 2, 3, 4]
y = [10, 20, 25, 30]

plt.plot(x, y, marker='o', markersize=5, markerfacecolor='green')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Line Plot with Dots')
plt.show()
```

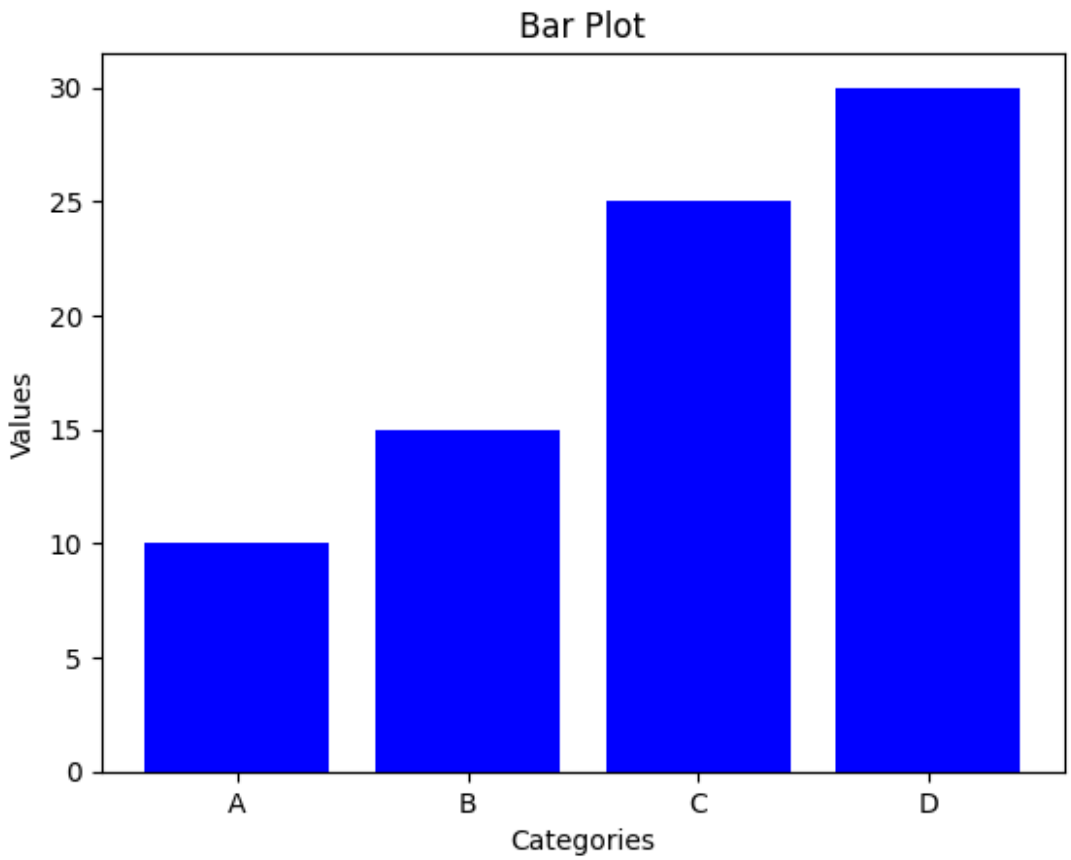


Scatter Plot-A scatter plot is used to display values for typically two variables for a set of data.

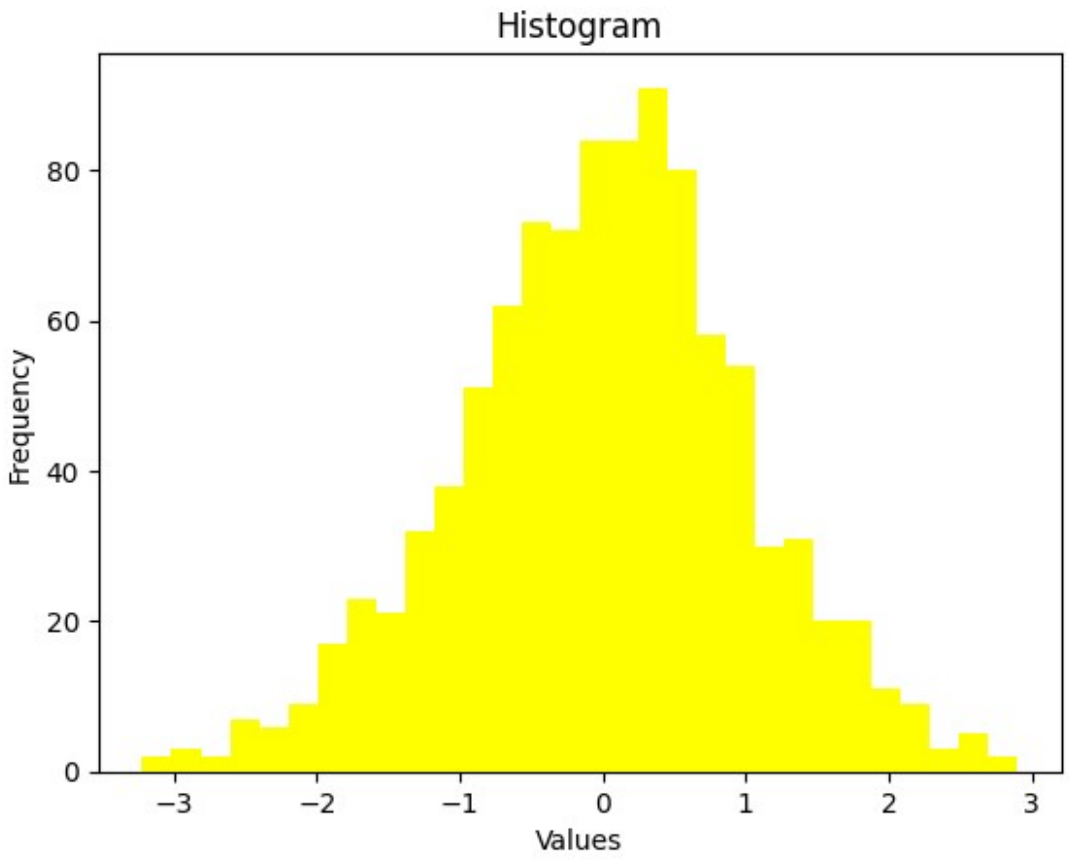
```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [10, 20, 25, 30, 50]

plt.figure(figsize=(8, 6))
plt.scatter(x, y, color='red')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Scatter Plot')
plt.show()
```



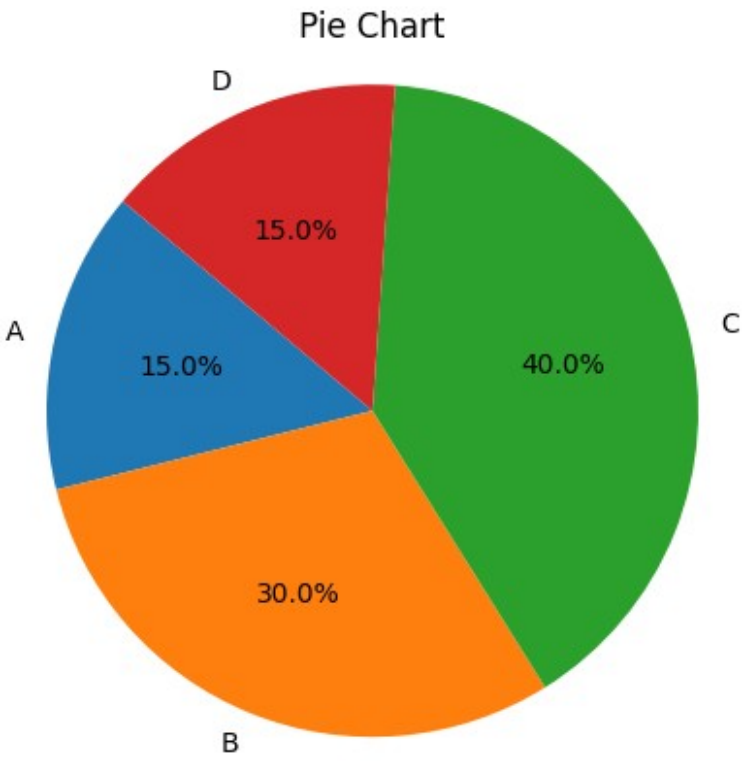

```
import matplotlib.pyplot as plt

x = ['A', 'B', 'C', 'D']
y = [10, 15, 25, 30]

plt.bar(x, y, color='blue')
plt.xlabel('Categories')
plt.ylabel('Values')
plt.title('Bar Plot')
plt.show()
```



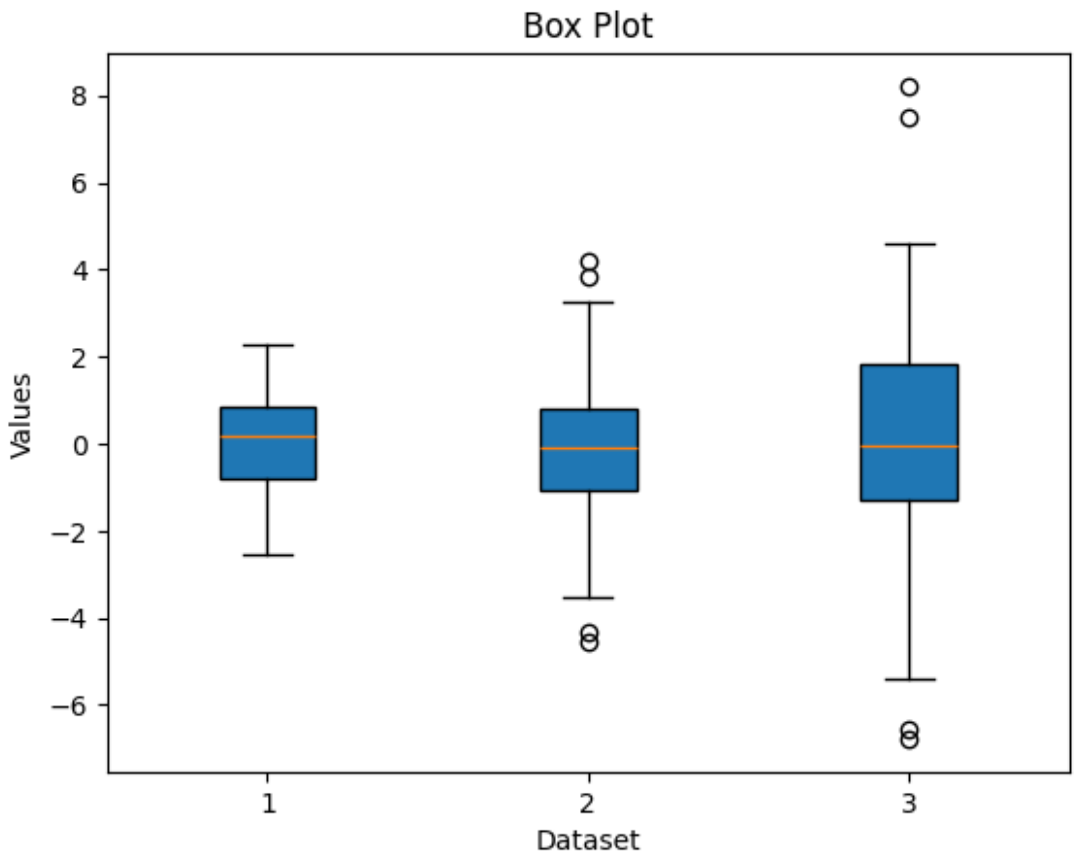

```
import matplotlib.pyplot as plt
import numpy as np

data = np.random.randn(1000)

plt.hist(data, bins=30, color='yellow')
plt.xlabel('Values')
plt.ylabel('Frequency')
plt.title('Histogram')
plt.show()
```



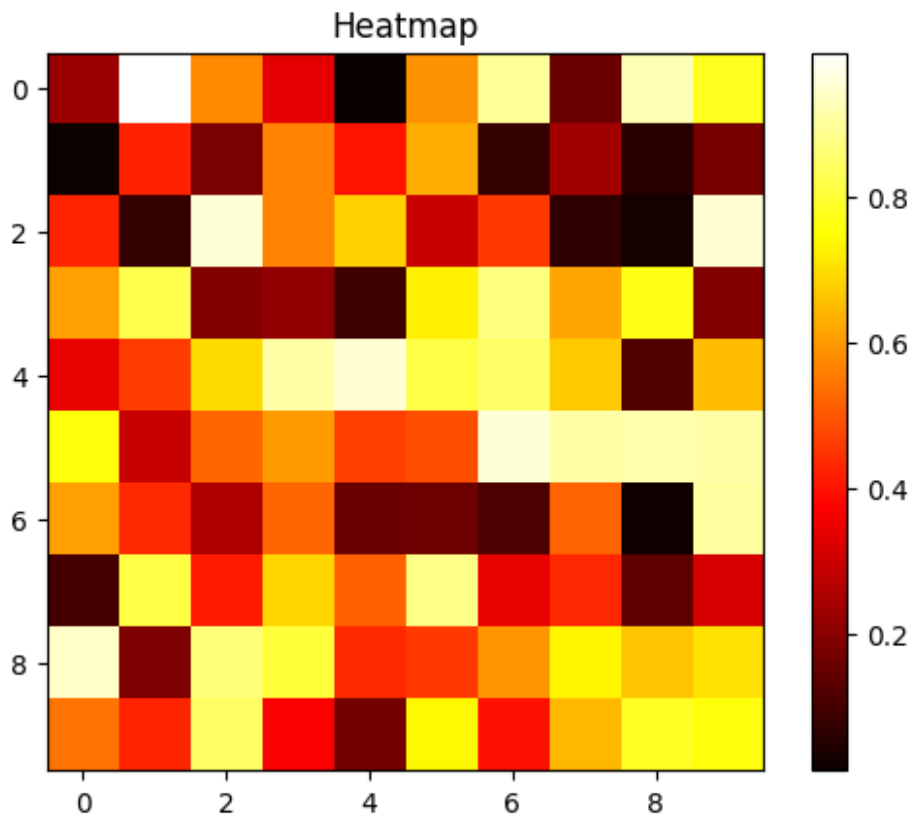

```python
import matplotlib.pyplot as plt

labels = ['A', 'B', 'C', 'D']
sizes = [15, 30, 40, 15]

plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140)
plt.axis('equal')
plt.title('Pie Chart')
plt.show()
```



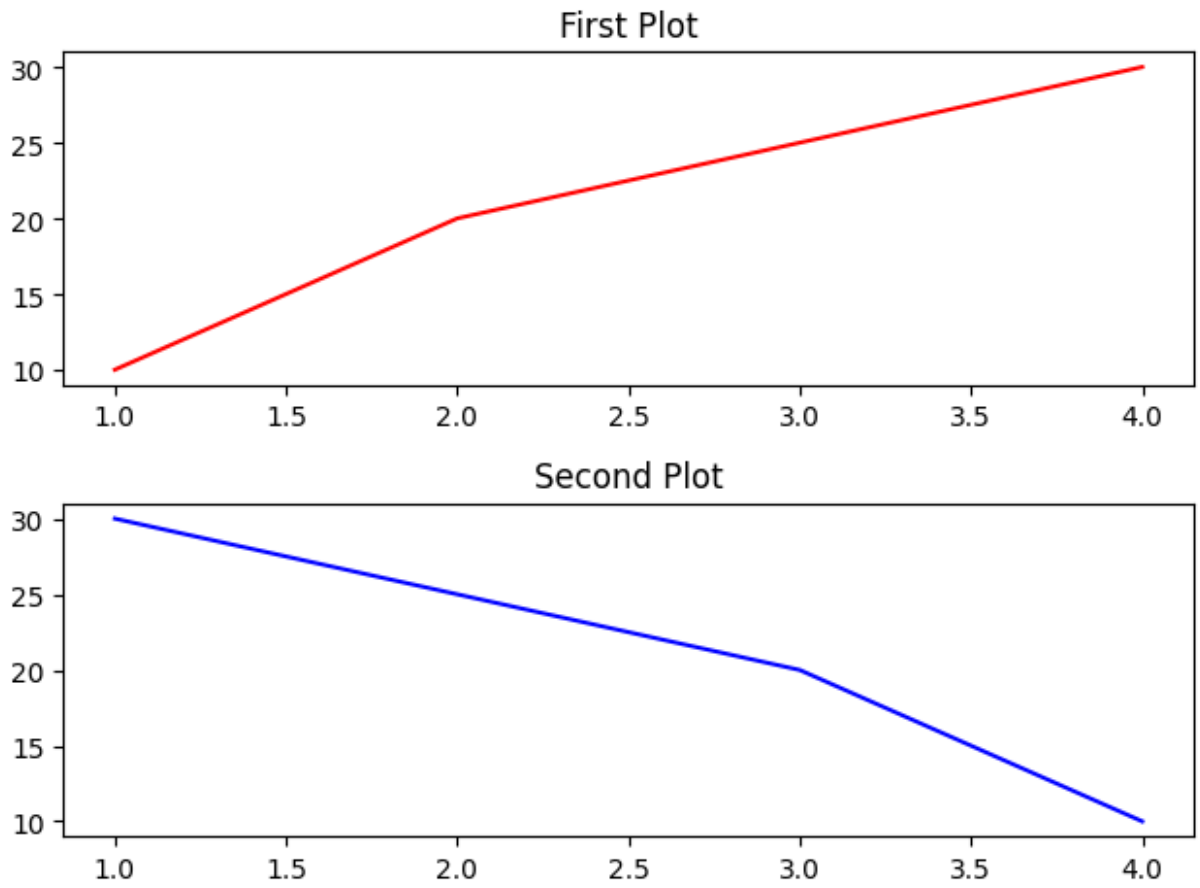

```
import matplotlib.pyplot as plt
import numpy as np

data = [np.random.normal(0, std, 100) for std in range(1, 4)]

plt.boxplot(data, vert=True, patch_artist=True)
plt.xlabel('Dataset')
plt.ylabel('Values')
plt.title('Box Plot')
plt.show()
```




```
import matplotlib.pyplot as plt
import numpy as np

data = np.random.rand(10, 10)

plt.imshow(data, cmap='hot', interpolation='nearest')
plt.colorbar()
plt.title('Heatmap')
plt.show()
```




```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4]
y1 = [10, 20, 25, 30]
y2 = [30, 25, 20, 10]

fig, axs = plt.subplots(2)

axs[0].plot(x, y1, 'r')
axs[0].set_title('First Plot')

axs[1].plot(x, y2, 'b')
axs[1].set_title('Second Plot')

plt.tight_layout()
plt.show()
```




```
from sklearn import datasets, preprocessing
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier

iris = datasets.load_iris()
X, y = iris.data[:, :2], iris.target

X_train, X_test, y_train, y_test = train_test_split(X,
y ,random_state=33)

scaler = preprocessing.StandardScaler().fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)

dt = DecisionTreeClassifier(random_state=33)
dt.fit(X_train, y_train)

y_pred = dt.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
```

```
Accuracy: 0.53
```