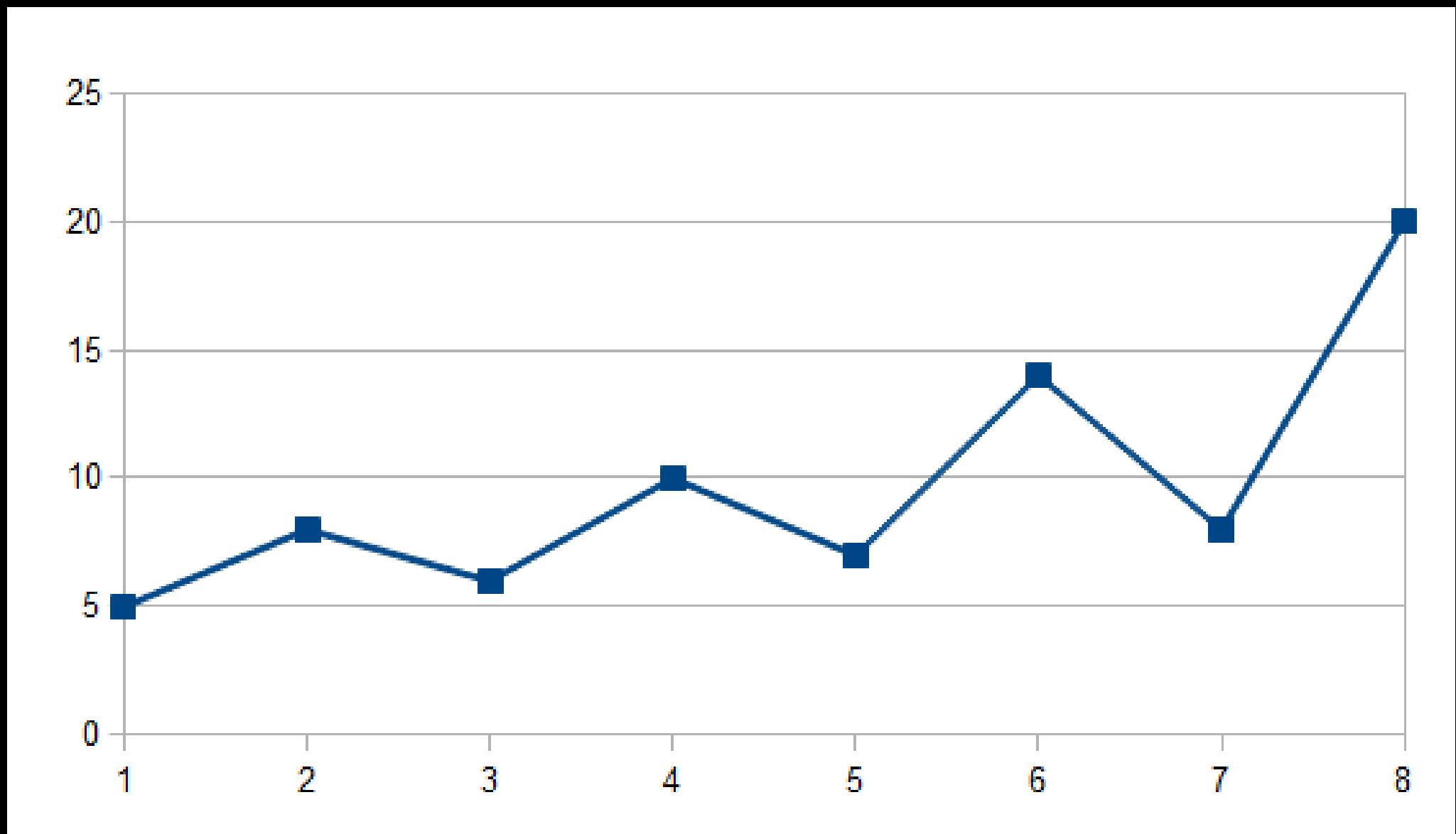


DATA VISUALIZATION IN PYTHON

DATA VISUALIZATION IS THE PROCESS OF REPRESENTING DATA IN A VISUAL FORMAT, SUCH AS CHARTS, GRAPHS, OR PLOTS, TO HELP UNDERSTAND THE UNDERLYING PATTERNS, TRENDS, AND INSIGHTS IN THE DATA. PYTHON, BEING A VERSATILE PROGRAMMING LANGUAGE, OFFERS SEVERAL LIBRARIES FOR CREATING VISUALLY APPEALING AND INFORMATIVE PLOTS.

LINE CHART



INTRODUCTION

A LINE CHART, OR LINE GRAPH, IS A TYPE OF CHART THAT DISPLAYS INFORMATION AS A SERIES OF DATA POINTS CONNECTED BY STRAIGHT LINE SEGMENTS. IT IS A BASIC TYPE OF CHART COMMON IN MANY FIELDS.

BASIC STRUCTURE

A LINE CHART IS OFTEN USED TO VISUALIZE A TREND IN DATA OVER INTERVALS OF TIME - A TIME SERIES - THUS THE LINE IS OFTEN DRAWN CHRONOLOGICALLY. THE X-AXIS REPRESENTS THE TIME INTERVALS, AND THE Y-AXIS REPRESENTS THE MEASURED VALUE.

UTILITY

LINE CHARTS ARE USED TO REPRESENT THE RELATIONSHIP BETWEEN TWO TYPES OF DATA BY PLOTTING DATA POINTS AND CONNECTING THEM WITH A LINE. THEY ARE PARTICULARLY USEFUL FOR SHOWING TRENDS OVER TIME.

EXAMPLE (1)

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

# Data
x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]

# Create a Figure
fig = plt.figure()

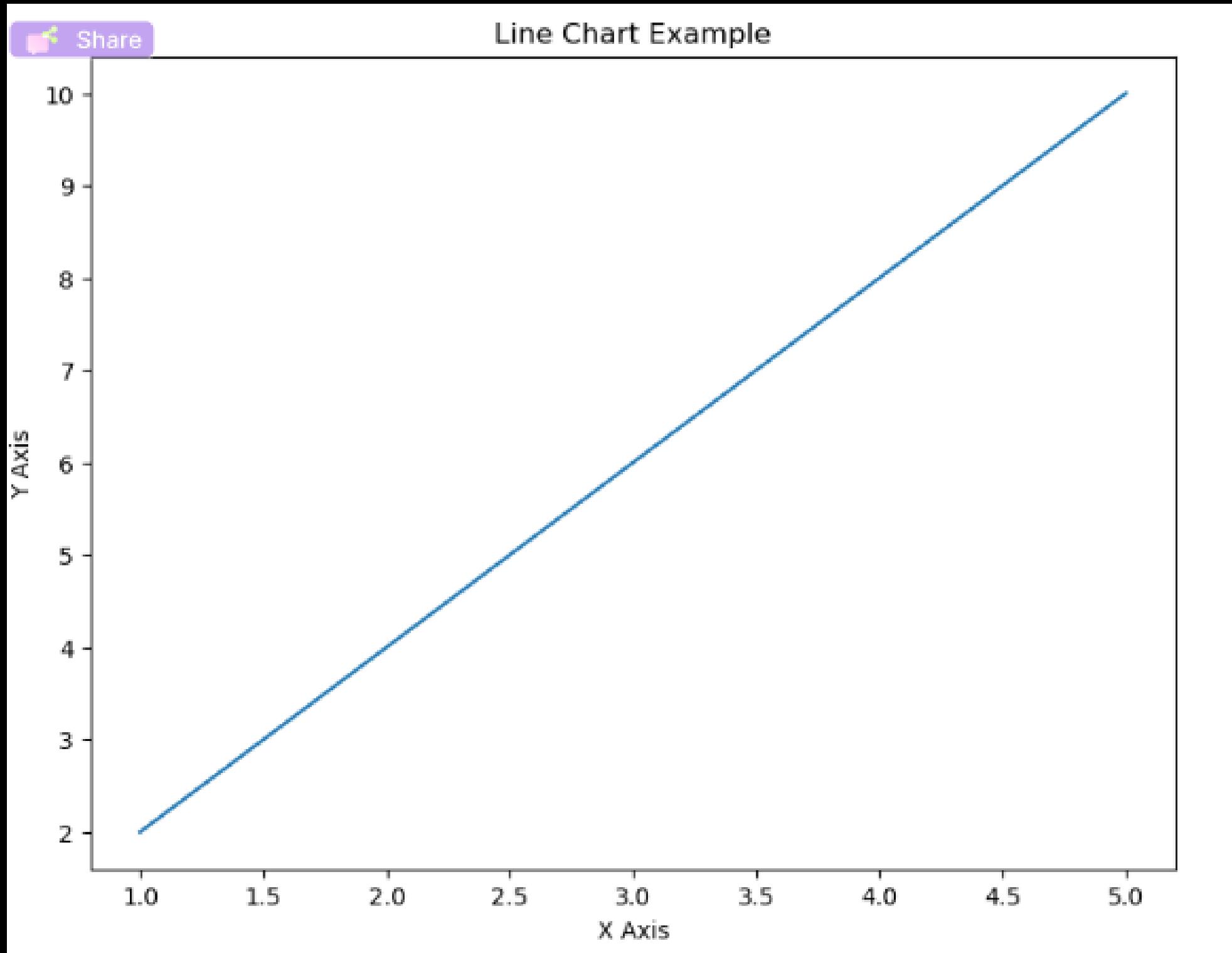
# Add Axes to the Figure
ax = fig.add_axes([0, 0, 1, 1])

# Plot data on the Axes
ax.plot(x, y)

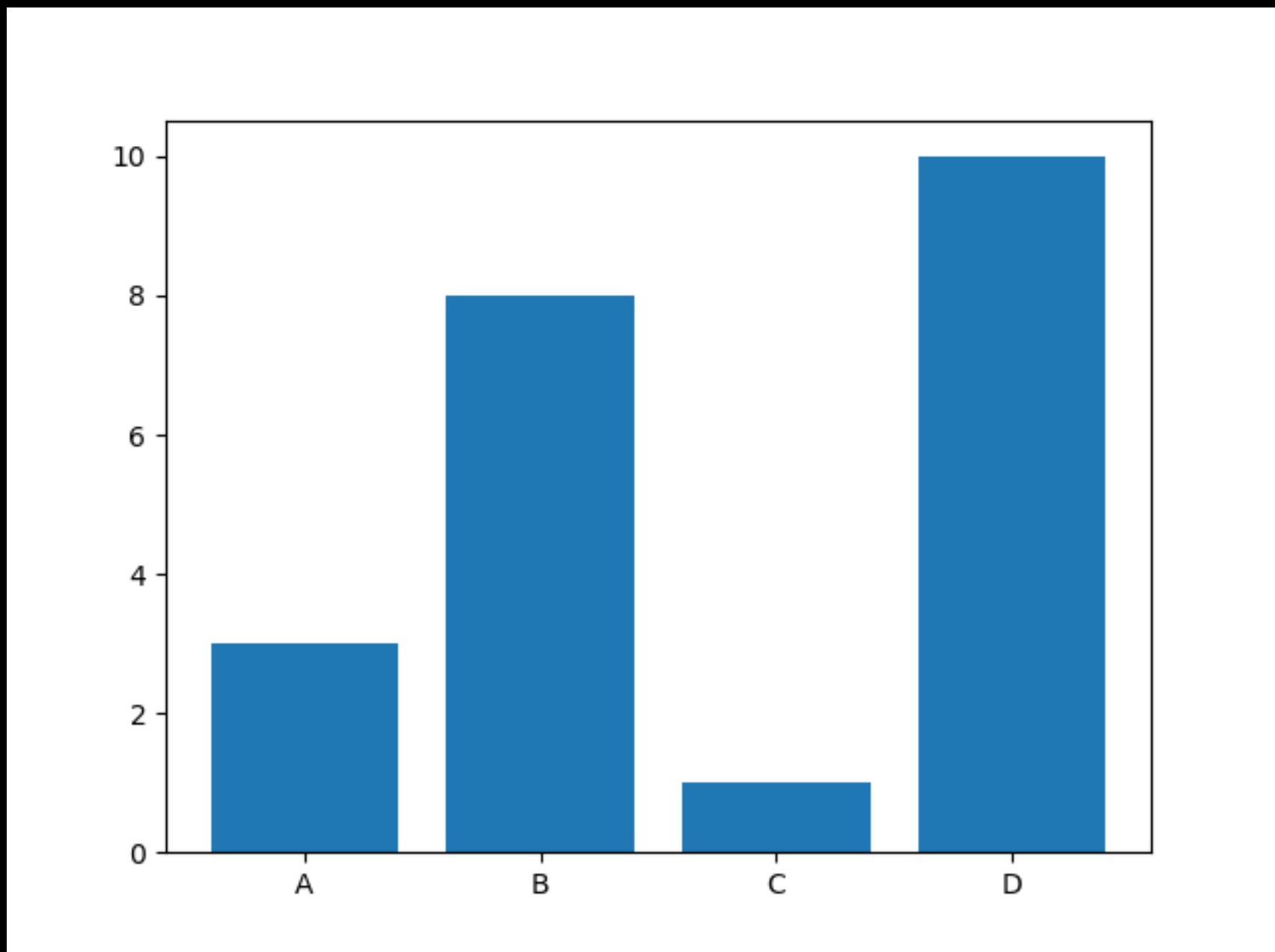
# Set the title and Labels
ax.set_title('Line Chart Example')
ax.set_xlabel('X Axis')
ax.set_ylabel('Y Axis')

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

EXAMPLE (1)



BAR CHART



INTRODUCTION

A BAR CHART, OR BAR GRAPH, IS A CHART OR GRAPH THAT PRESENTS CATEGORICAL DATA WITH RECTANGULAR BARS. EACH BAR REPRESENTS A CATEGORY AND THE HEIGHT OR LENGTH OF THE BAR IS PROPORTIONAL TO THE COUNT OR FREQUENCY OF THAT CATEGORY.

BASIC STRUCTURE

THE X-AXIS REPRESENTS THE CATEGORIES AND
THE Y-AXIS REPRESENTS THE FREQUENCY OR
COUNT.

UTILITY

BAR CHARTS ARE USED FOR COMPARING
MULTIPLE CATEGORIES OF DATA. THEY ARE
PARTICULARLY USEFUL WHEN YOU WANT TO
COMPARE DATA THAT ARE GROUPED INTO
CATEGORICAL BINS.

EXAMPLE (2)

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

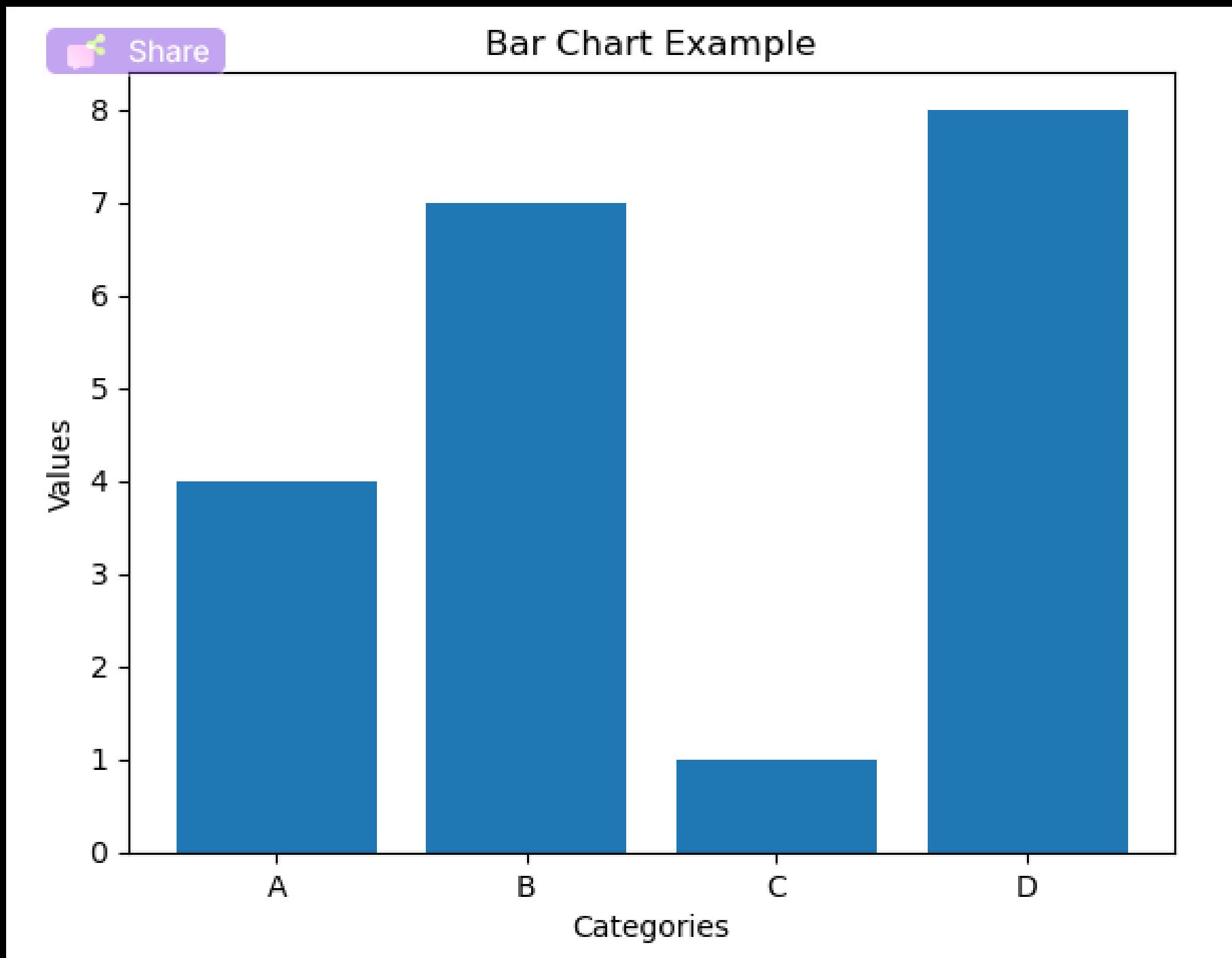
# Data
categories = ['A', 'B', 'C', 'D']
values = [4, 7, 1, 8]

# Plot
plt.bar(categories, values)

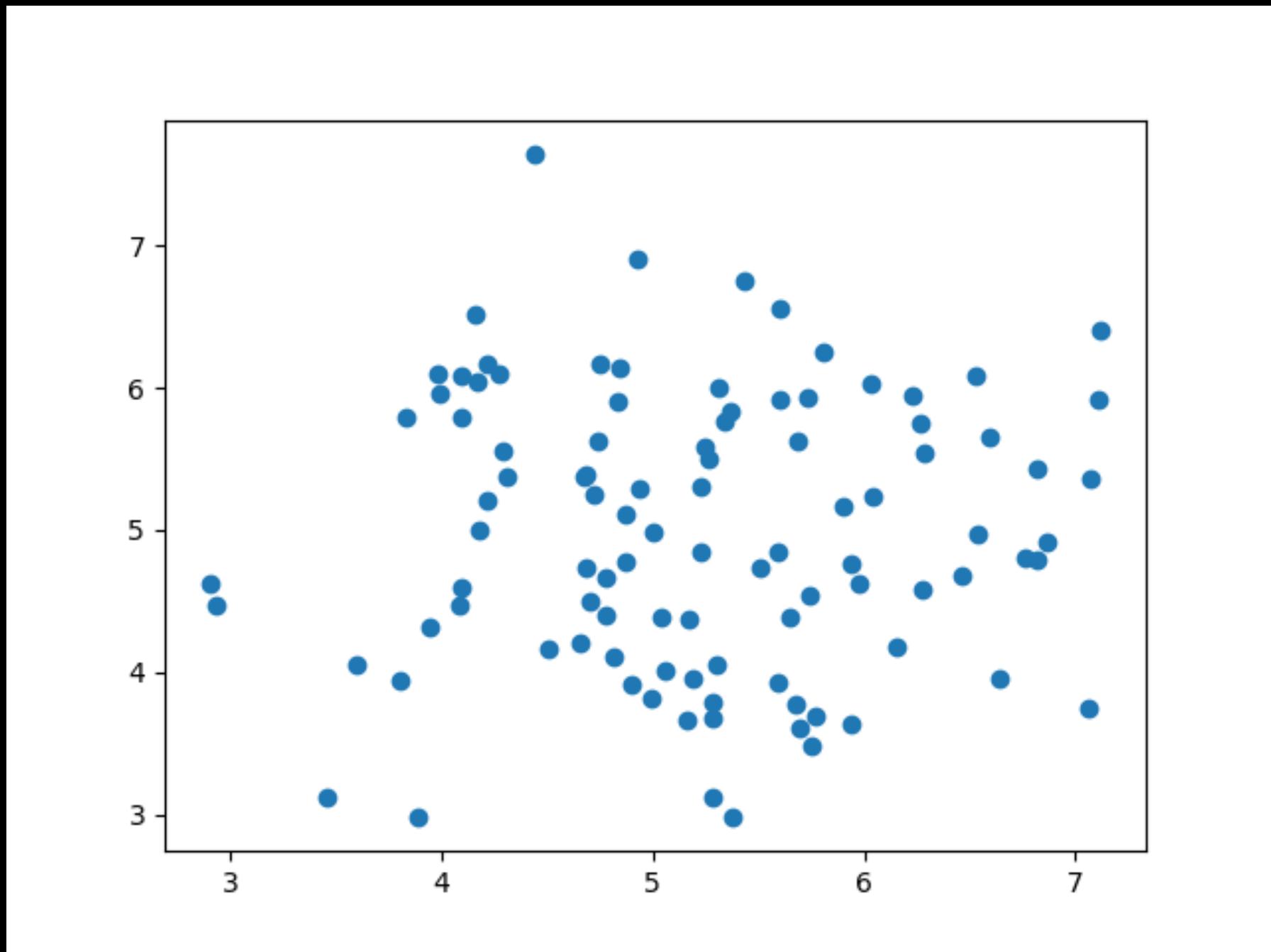
# Title and Labels
plt.title('Bar Chart Example')
plt.xlabel('Categories')
plt.ylabel('Values')

# Show plot
plt.show()
```

EXAMPLE (2)



SCATTER PLOT



INTRODUCTION

A SCATTER PLOT, OR SCATTER CHART, IS A TYPE OF DATA VISUALIZATION THAT USES DOTS TO REPRESENT THE VALUES OBTAINED FOR TWO DIFFERENT VARIABLES - ONE PLOTTED ALONG THE X-AXIS AND THE OTHER PLOTTED ALONG THE Y-AXIS.

BASIC STRUCTURE

A SCATTER PLOT IS USED TO UNDERSTAND THE RELATIONSHIP BETWEEN TWO CONTINUOUS VARIABLES. THE VALUES OF EACH POINT ON THE X AND Y AXIS ARE USED TO PLACE THE DOT ON THE CHART.

UTILITY

SCATTER PLOTS ARE USED TO OBSERVE AND SHOW RELATIONSHIPS BETWEEN TWO NUMERIC VARIABLES. THE DOTS IN A SCATTER PLOT NOT ONLY REPORT THE VALUES OF INDIVIDUAL DATA POINTS, BUT ALSO PATTERNS WHEN THE DATA ARE TAKEN AS A WHOLE.

EXAMPLE (3)

```
In [3]: import matplotlib.pyplot as plt

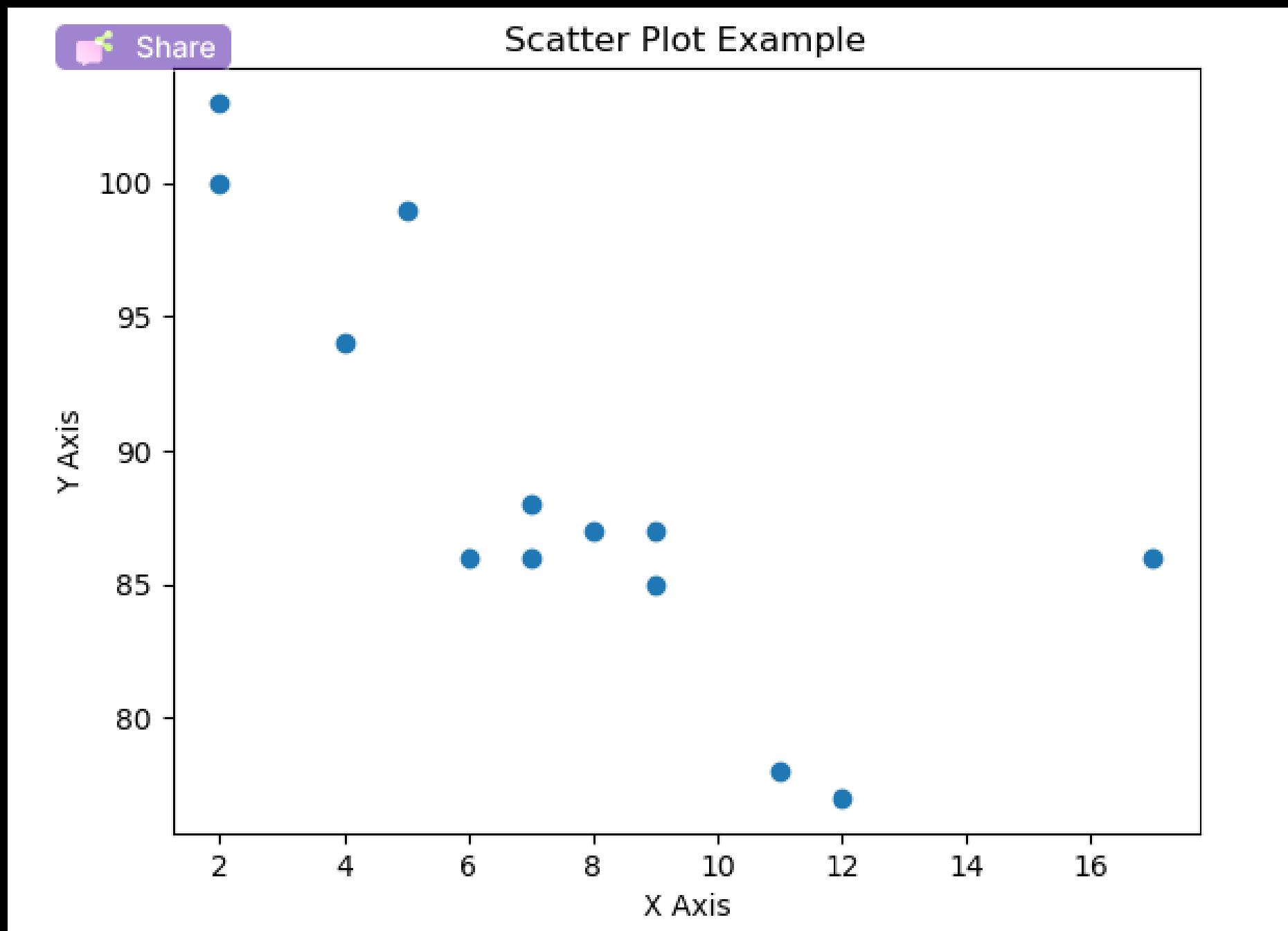
# Data
x = [5, 7, 8, 7, 2, 17, 2, 9, 4, 11, 12, 9, 6]
y = [99, 86, 87, 88, 100, 86, 103, 87, 94, 78, 77, 85, 86]

# Create a Scatter Plot
plt.scatter(x, y)

# Set the title and labels
plt.title('Scatter Plot Example')
plt.xlabel('X Axis')
plt.ylabel('Y Axis')

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

EXAMPLE (3)



TIME SERIES PLOT



INTRODUCTION

A TIME SERIES PLOT IS A LINE PLOT WHERE THE X-AXIS REPRESENTS TIME.

BASIC STRUCTURE

THE X-AXIS REPRESENTS THE TIME INTERVALS, AND THE Y-AXIS REPRESENTS THE MEASURED VALUE.

UTILITY

TIME SERIES PLOTS ARE USED TO OBSERVE HOW DATA CHANGES OVER TIME. THEY ARE PARTICULARLY USEFUL FOR IDENTIFYING TRENDS, SEASONAL PATTERNS, AND CYCLES IN DATA.

EXAMPLE (4)

```
import matplotlib.pyplot as plt
import datetime
import numpy as np
import pandas as pd

#define data
df = pd.DataFrame({'date': np.array([datetime.datetime(2020, 1, i+1)
                                         for i in range(12)]),
                    'sales': [3, 4, 4, 7, 8, 9, 14, 17, 12, 8, 8, 13]})

df2 = pd.DataFrame({'date': np.array([datetime.datetime(2020, 1, i+1)
                                         for i in range(12)]),
                    'returns': [1, 1, 2, 3, 3, 3, 4, 3, 2, 3, 4, 7]})

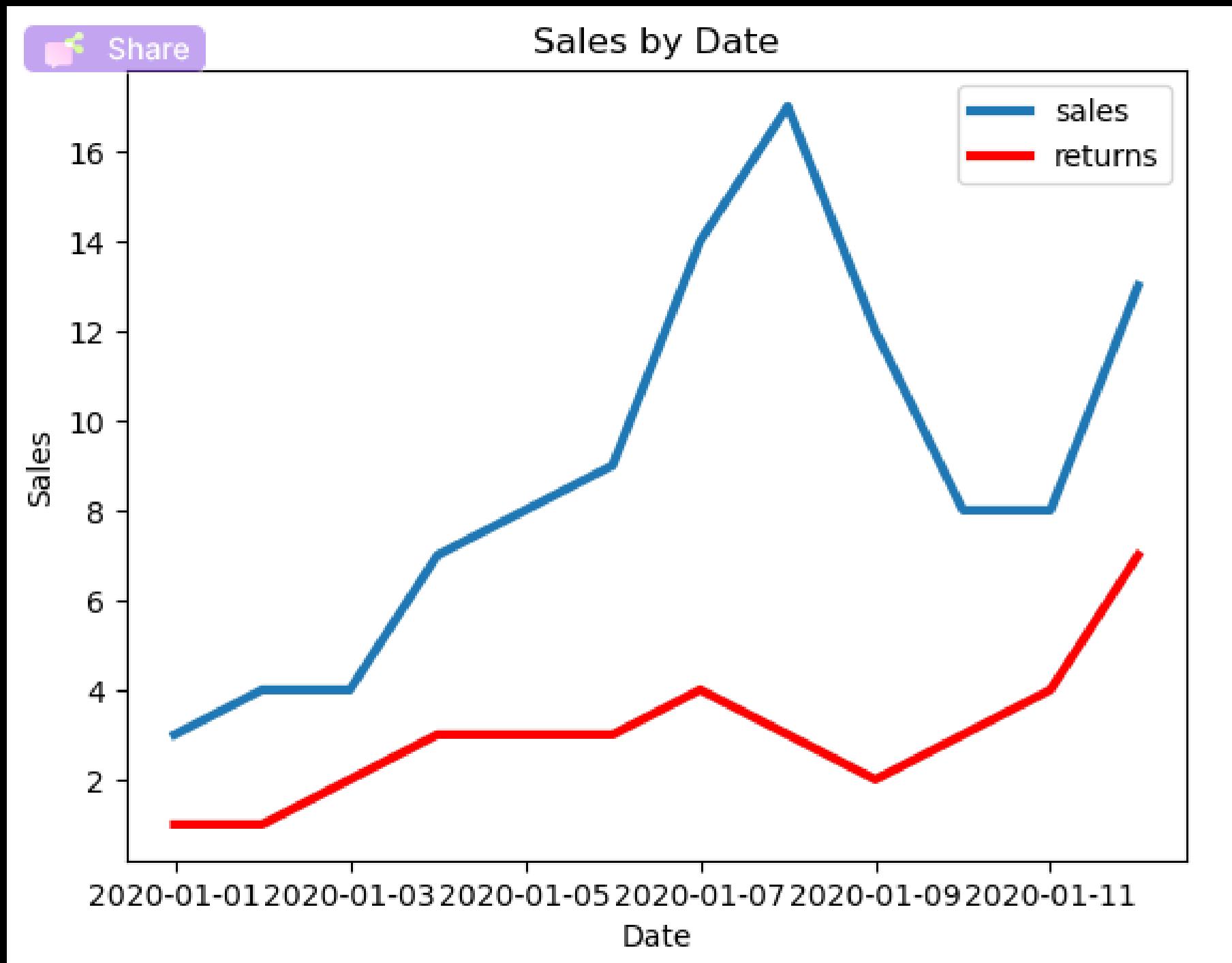
#plot both time series
plt.plot(df.date, df.sales, label='sales', linewidth=3)
plt.plot(df2.date, df2.returns, color='red', label='returns', linewidth=3)

#add title and axis labels
plt.title('Sales by Date')
plt.xlabel('Date')
plt.ylabel('Sales')

#add legend
plt.legend()

#display plot
plt.show()
```

EXAMPLE (4)

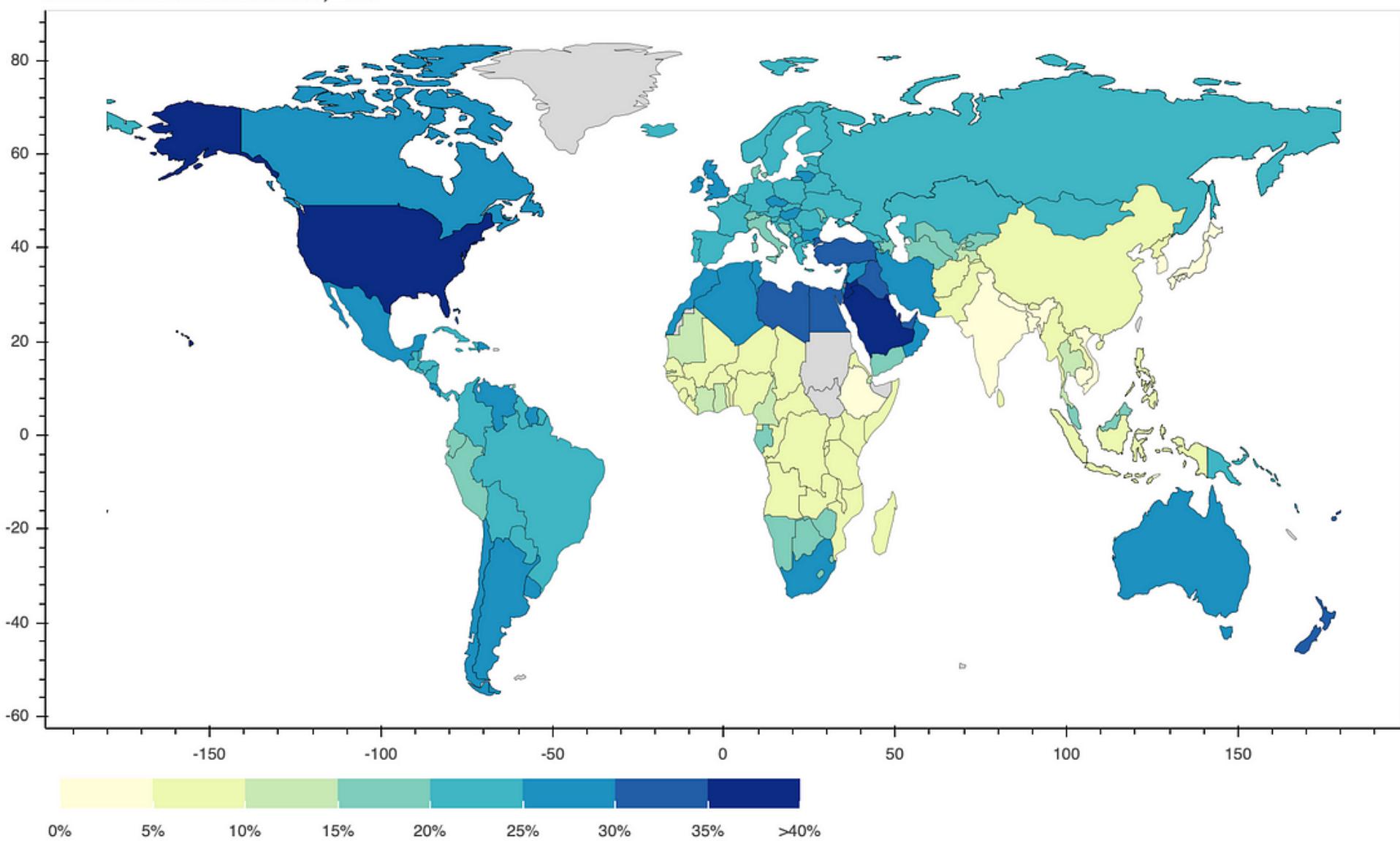


MAP



BokehJS 0.12.14 successfully loaded.

Share of adults who are obese, 2016



INTRODUCTION

MAPS CAN BE CREATED USING THE FOLIUM LIBRARY IN PYTHON.

BASIC STRUCTURE

THE MAP IS CREATED WITH A SPECIFIED LOCATION AND ZOOM LEVEL. MARKERS AND OTHER FEATURES CAN BE ADDED TO THE MAP.

UTILITY

MAPS ARE USED TO VISUALIZE GEOGRAPHIC DATA.
THEY ARE PARTICULARLY USEFUL FOR SHOWING
THE SPATIAL RELATIONSHIPS BETWEEN FEATURES.

EXAMPLE (5)

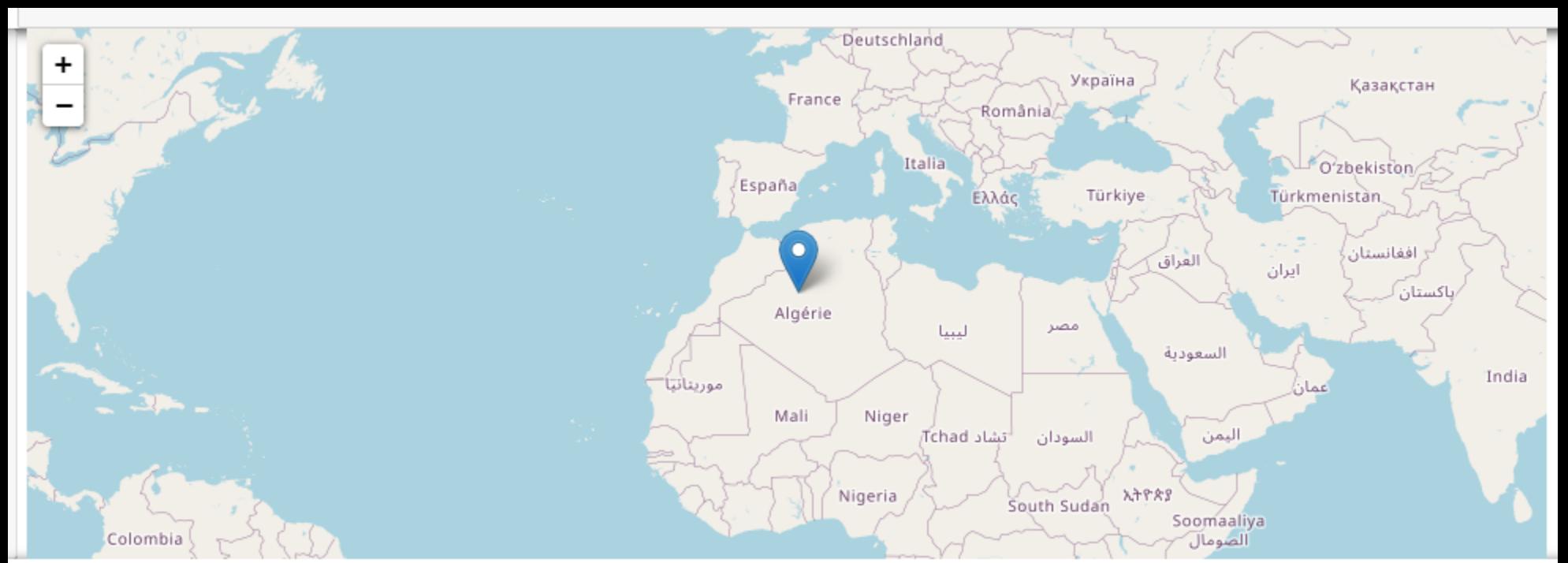
```
In [12]: import folium

# Create a map centered at an average location
map = folium.Map(location=[10,0], zoom_start=2.5)

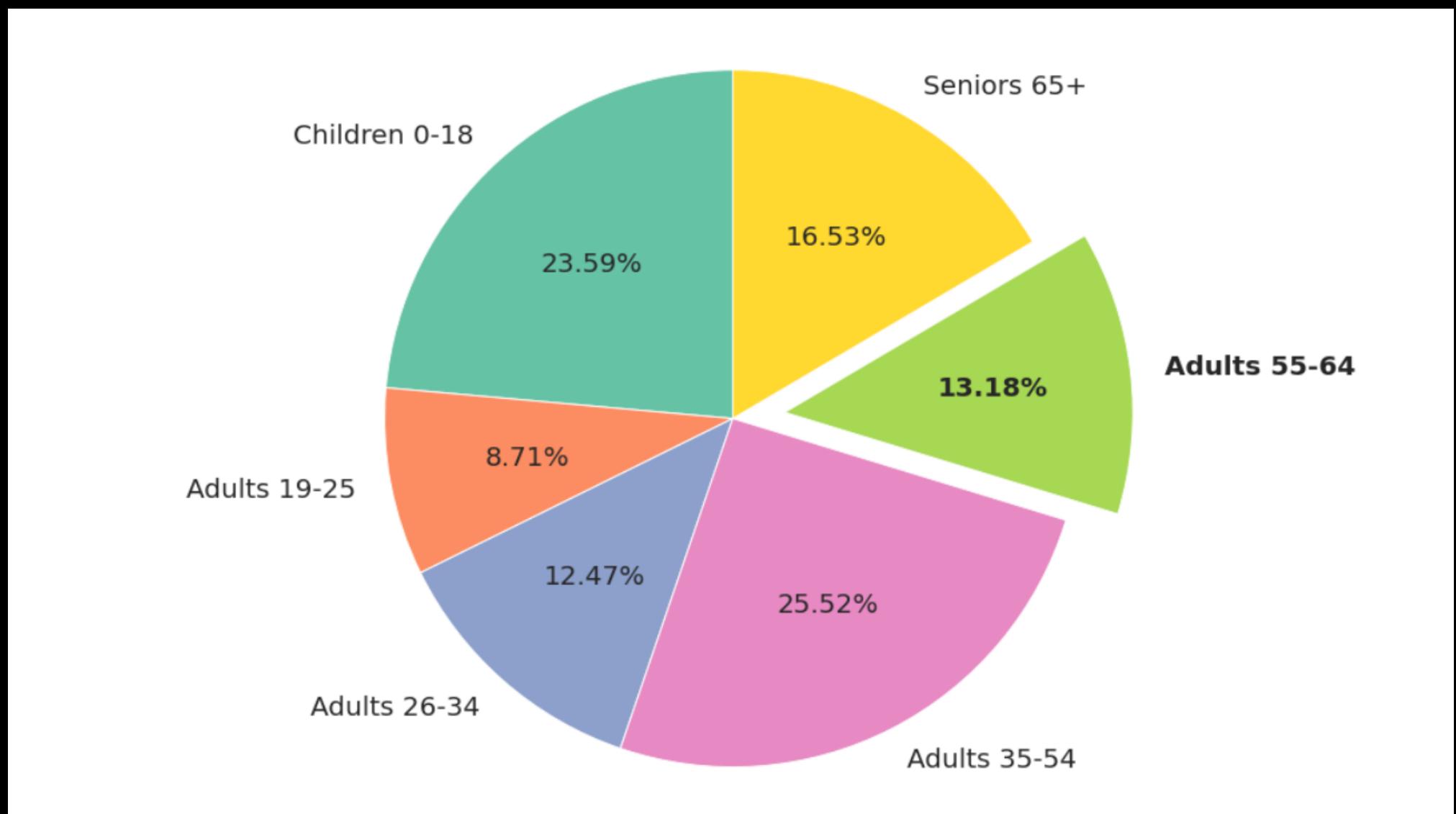
# Add a marker to the map
folium.Marker(location=[30,0], popup='Algeria').add_to(map)

# Show the map
map
```

EXAMPLE (5)



PIE CHART



INTRODUCTION

A PIE CHART IS A CIRCULAR STATISTICAL GRAPHIC DIVIDED INTO SLICES TO ILLUSTRATE NUMERICAL PROPORTION.

BASIC STRUCTURE

EACH SLICE OF THE PIE REPRESENTS A CATEGORY, AND THE SIZE OF THE SLICE IS PROPORTIONAL TO THE FREQUENCY OR COUNT OF THAT CATEGORY.

UTILITY

PIE CHARTS ARE USED TO VISUALIZE THE DISTRIBUTION OF CATEGORIES IN A DATASET. THEY ARE PARTICULARLY USEFUL FOR SHOWING THE PROPORTION OF EACH CATEGORY IN A DATASET.

EXAMPLE (6)

```
In [14]: import matplotlib.pyplot as plt

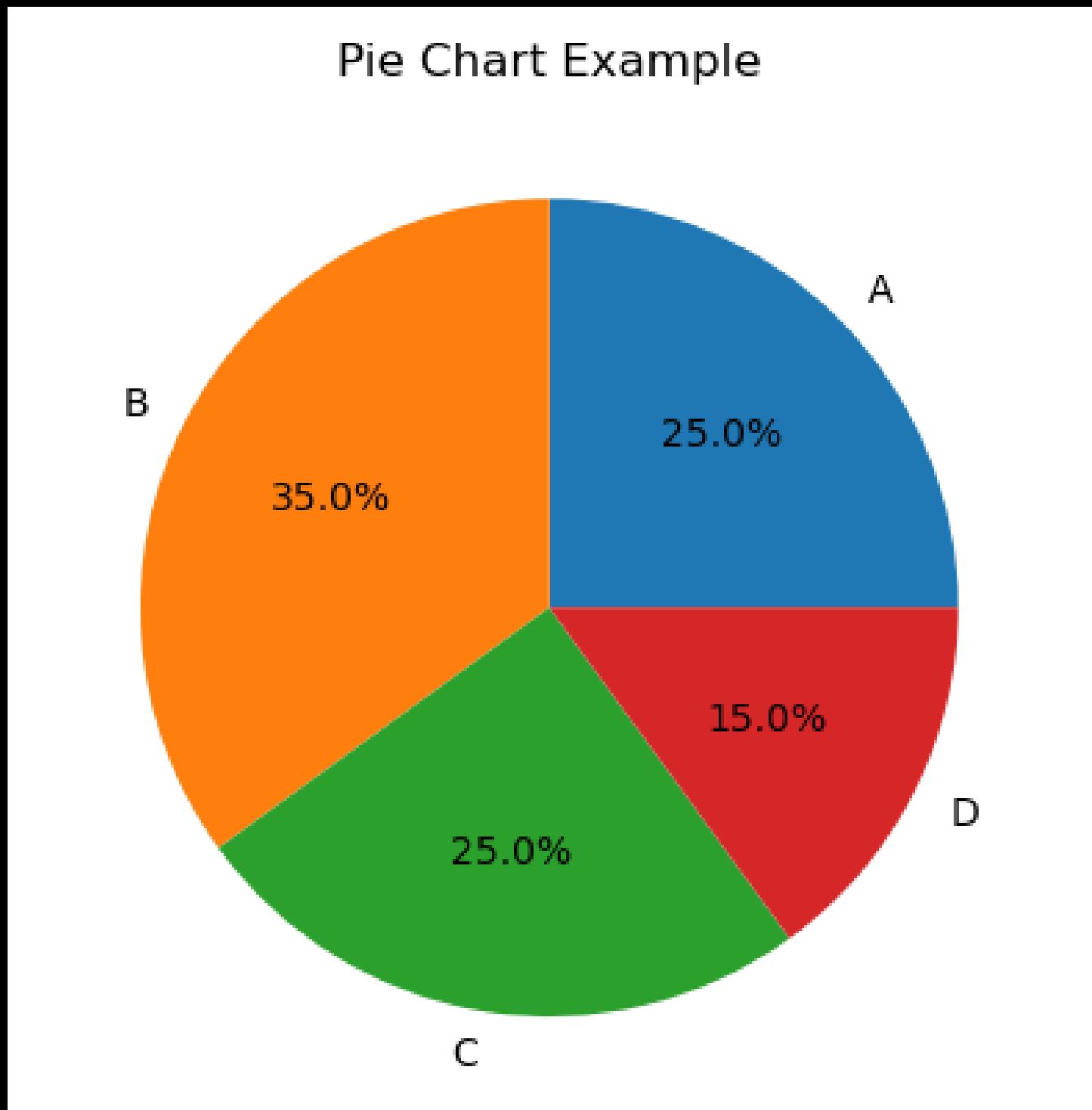
# Data
labels = ['A', 'B', 'C', 'D']
sizes = [25, 35, 25, 15]

# Create a Pie Chart
plt.pie(sizes, labels=labels, autopct='%1.1f%%')

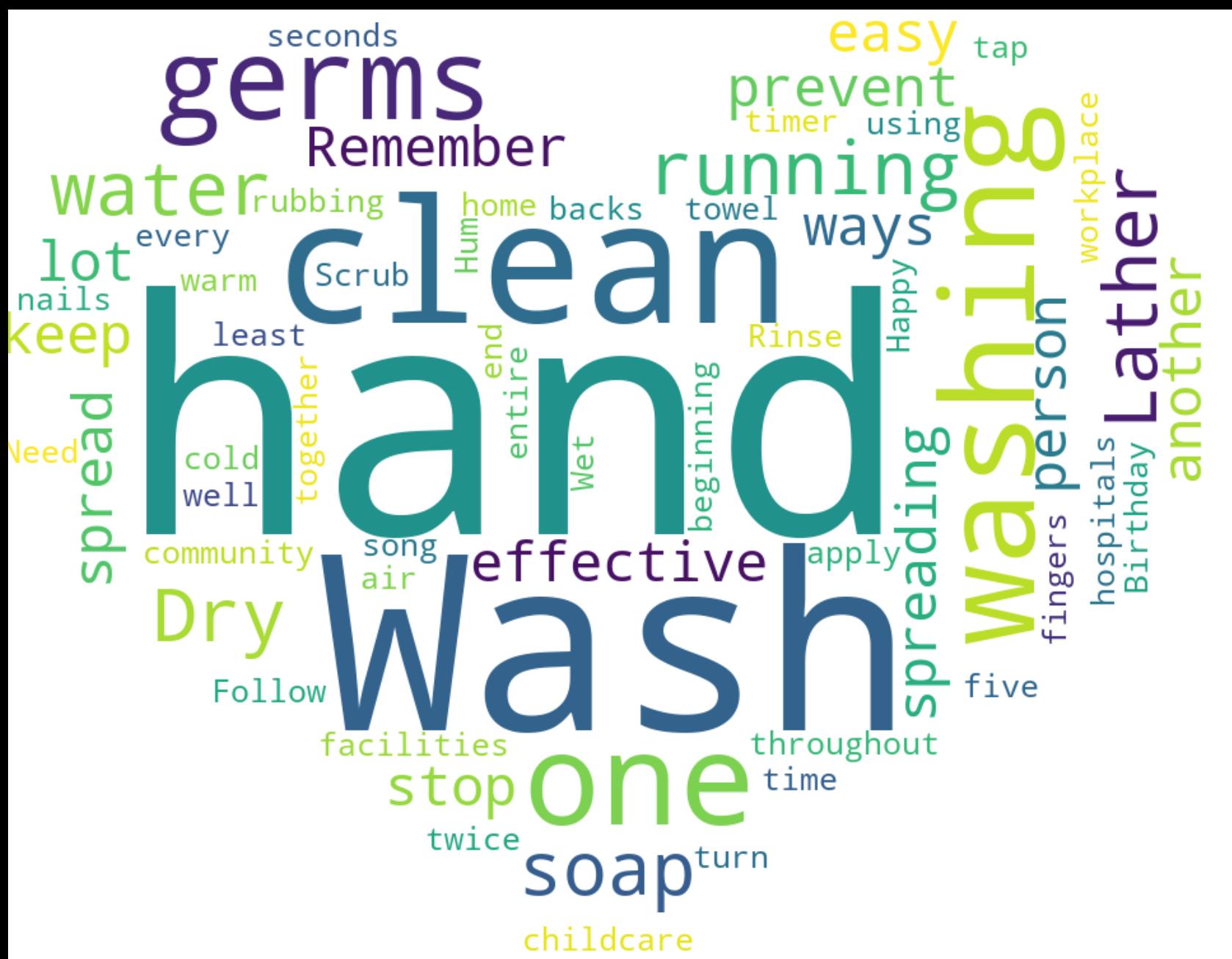
# Set the title
plt.title('Pie Chart Example')

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

EXAMPLE (6)



WORD CLOUD



INTRODUCTION

A WORD CLOUD IS A DATA VISUALIZATION
TECHNIQUE USED FOR REPRESENTING TEXT DATA.

BASIC STRUCTURE

THE SIZE OF EACH WORD IN THE WORD CLOUD IS
PROPORTIONAL TO ITS FREQUENCY IN THE TEXT
DATA.

UTILITY

WORD CLOUDS ARE USED TO VISUALIZE THE MOST FREQUENTLY OCCURRING WORDS IN A TEXT DATASET. THEY ARE PARTICULARLY USEFUL FOR QUICKLY IDENTIFYING THE MOST IMPORTANT WORDS IN A DATASET.

EXAMPLE (7)

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Data
text = 'Data Science Data Science Python Python Python Matplotlib Seaborn'

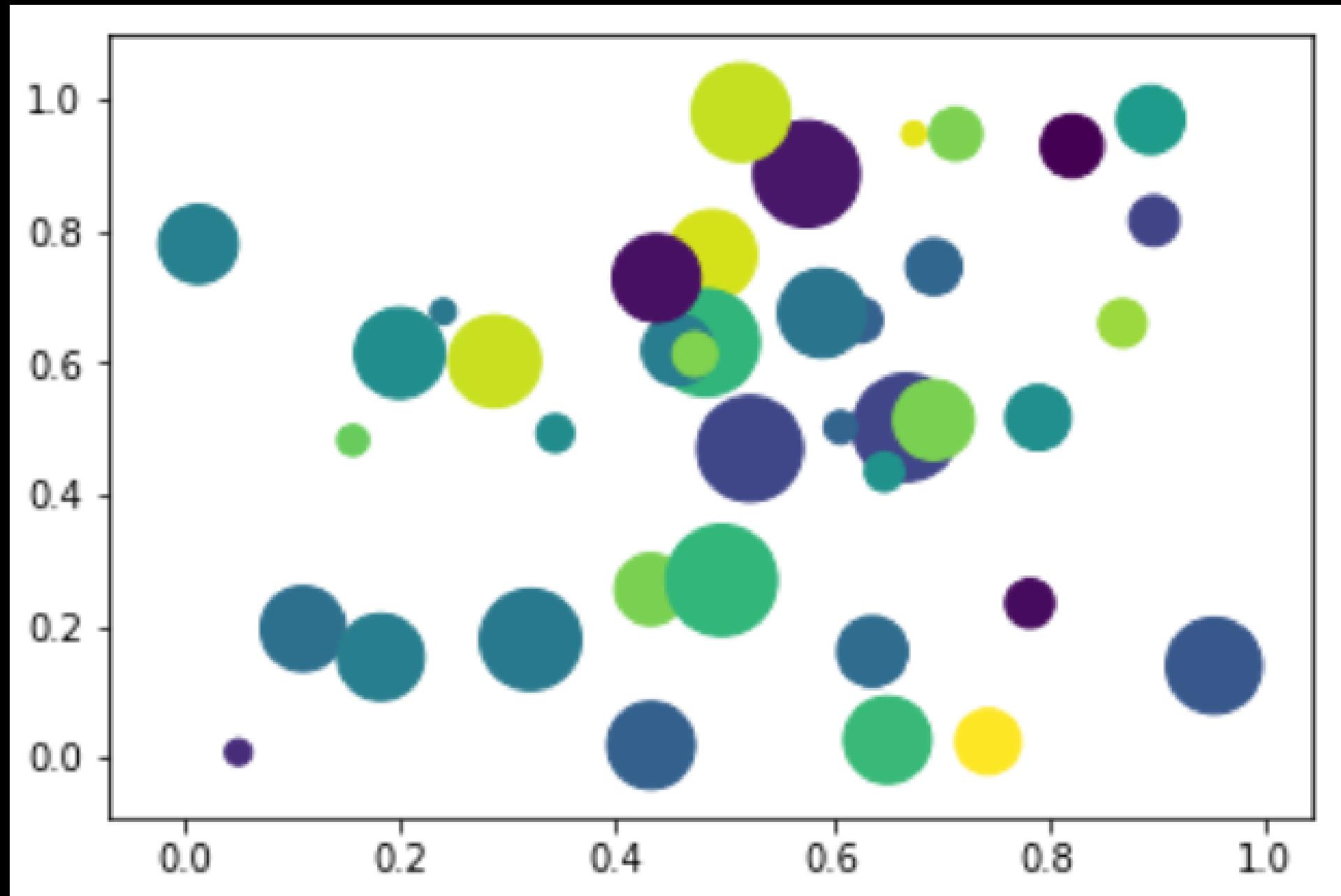
# Create a Word Cloud
wordcloud = WordCloud(width=480, height=480, margin=0).generate(text)

# Show the Word Cloud
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.margins(x=0, y=0)
plt.show()
```

EXAMPLE (7)

Python
Science
+ Data
Seaborn
Matplotlib

BUBBLE CHART



INTRODUCTION

A BUBBLE CHART IS A VARIATION OF A SCATTER PLOT.

BASIC STRUCTURE

EACH DOT IN A BUBBLE CHART REPRESENTS AN ENTITY, WITH ITS POSITION DETERMINED BY ITS VALUES FOR TWO VARIABLES, AND ITS SIZE DETERMINED BY A THIRD VARIABLE.

UTILITY

BUBBLE CHARTS ARE USED TO COMPARE AND SHOW THE RELATIONSHIPS BETWEEN CATEGORIES USING THEIR POSITIONS ON THE X AND Y AXIS AND THE SIZE OF THE BUBBLES.

EXAMPLE (8)

```
import matplotlib.pyplot as plt

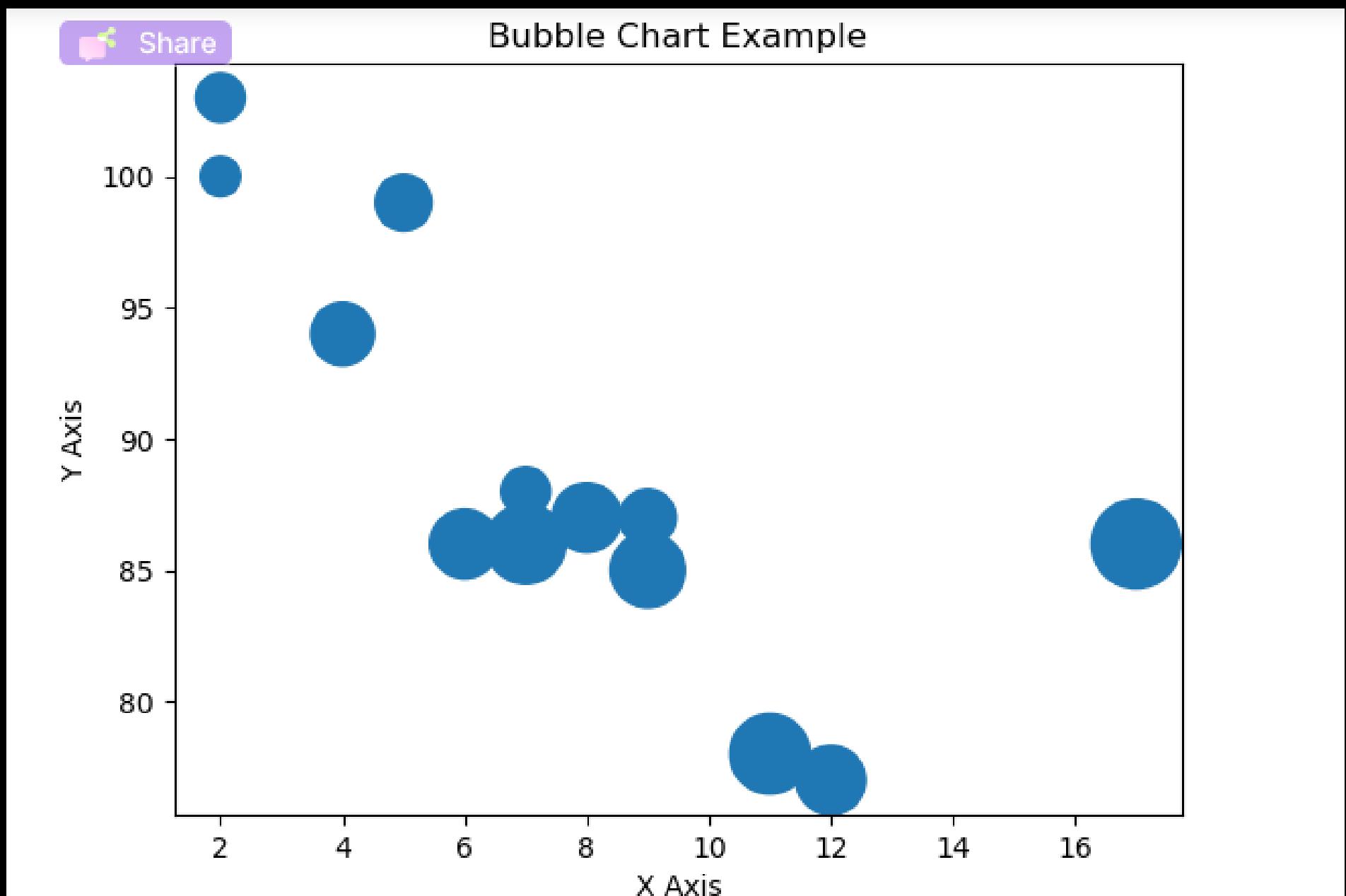
# Data
x = [5, 7, 8, 7, 2, 17, 2, 9, 4, 11, 12, 9, 6]
y = [99, 86, 87, 88, 100, 86, 103, 87, 94, 78, 77, 85, 86]
size = [400, 800, 600, 300, 200, 1000, 300, 400, 500, 800, 600, 700, 600]

# Create a Bubble Chart
plt.scatter(x, y, s=size)

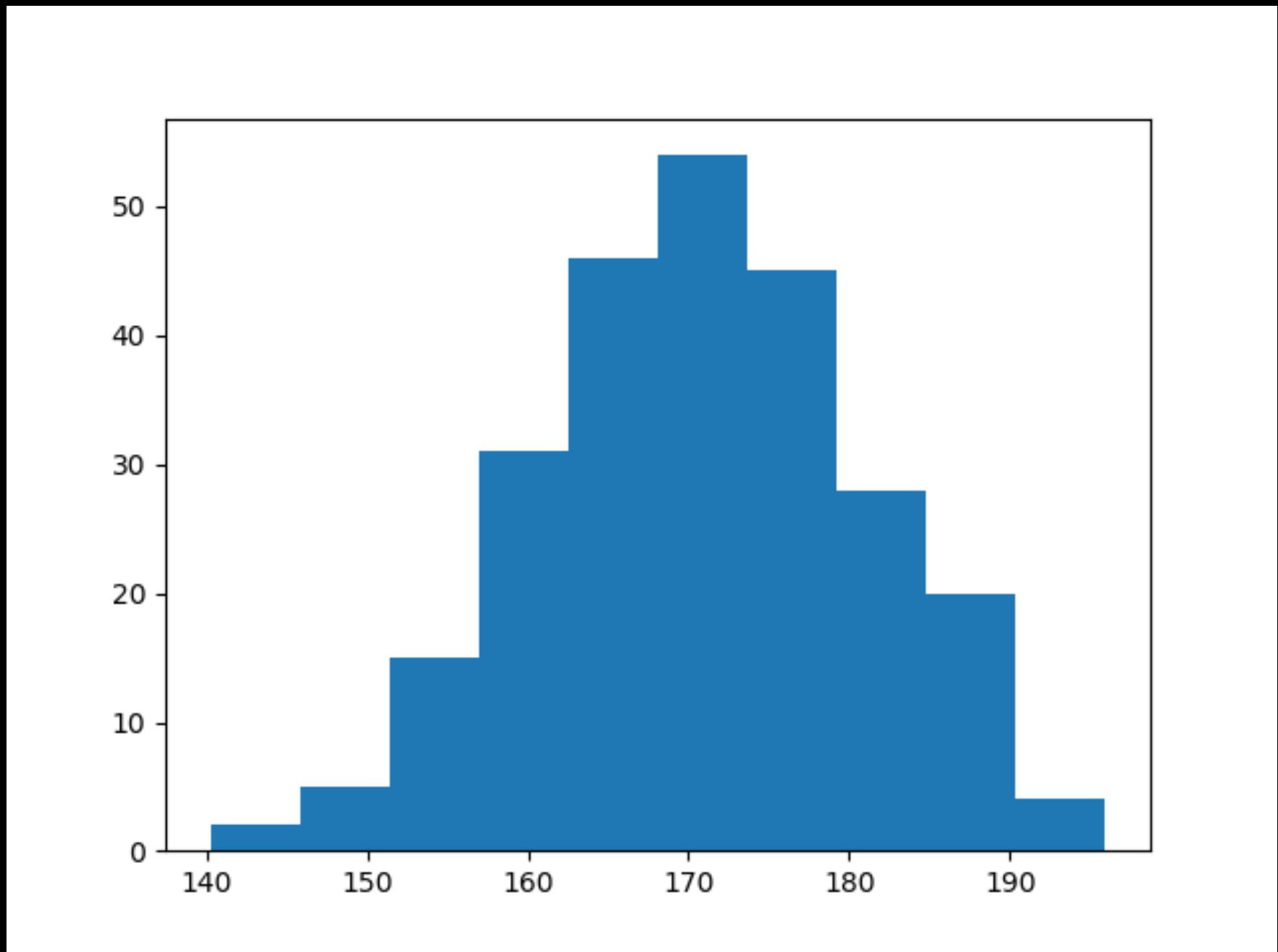
# Set the title and labels
plt.title('Bubble Chart Example')
plt.xlabel('X Axis')
plt.ylabel('Y Axis')

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

EXAMPLE (8)



HISTOGRAM



INTRODUCTION

A HISTOGRAM IS AN ACCURATE GRAPHICAL REPRESENTATION OF THE DISTRIBUTION OF A DATASET.

BASIC STRUCTURE

HISTOGRAMS DIVIDE THE DATA INTO A SET OF INTERVALS (OR "BINS") AND THEN COUNT HOW MANY DATA POINTS FALL INTO EACH BIN.

UTILITY

HISTOGRAMS ARE USED TO SHOW THE DISTRIBUTION OF A DATASET. THEY ARE PARTICULARLY USEFUL FOR IDENTIFYING THE CENTER, SPREAD, AND SKEWNESS OF YOUR DATA.

EXAMPLE (9)

```
import matplotlib.pyplot as plt

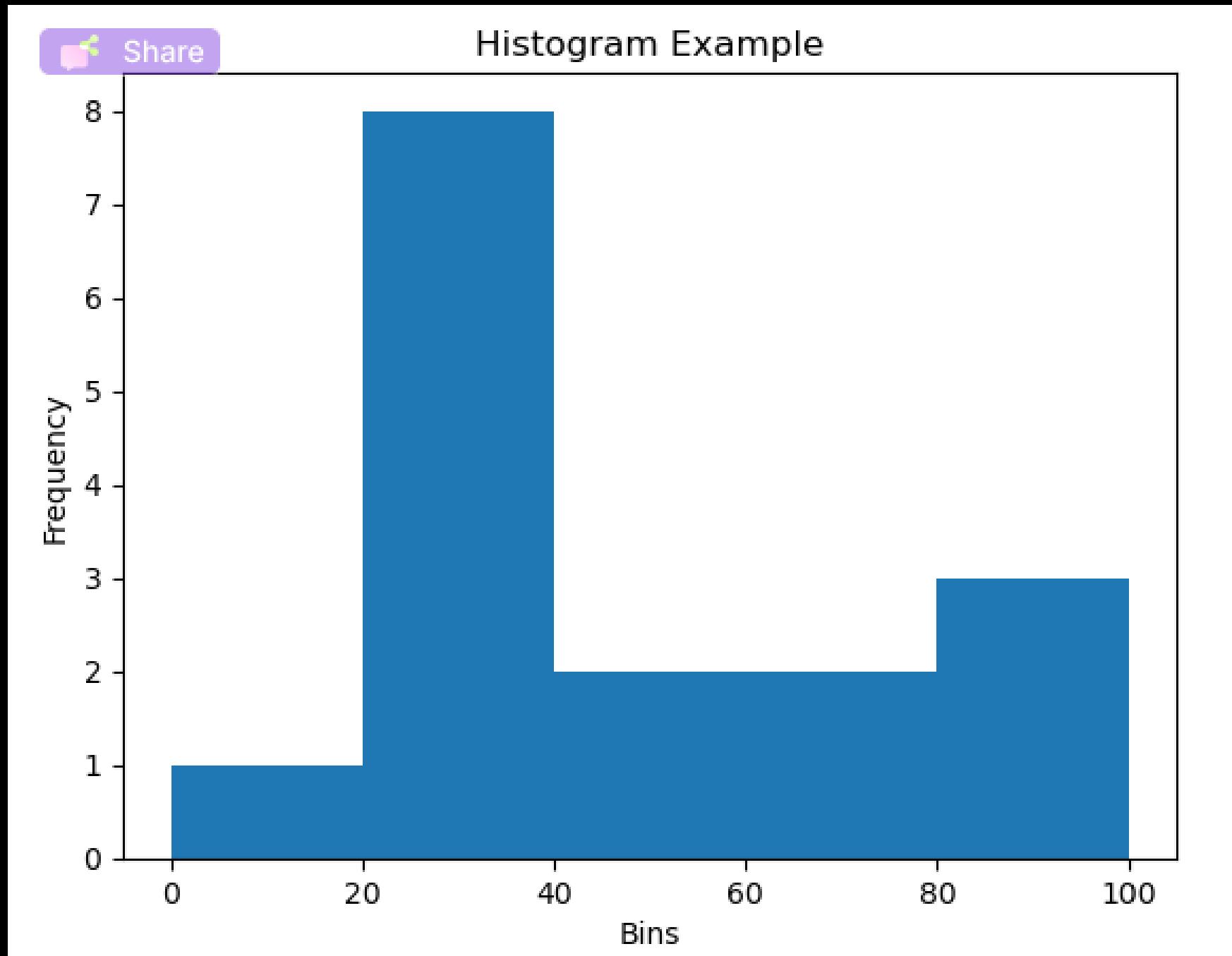
# Data
data = [10, 20, 20, 20, 30, 30, 30, 30, 30, 40, 50, 60, 70, 80, 90, 100]

# Create a Histogram
plt.hist(data, bins=[0, 20, 40, 60, 80, 100])

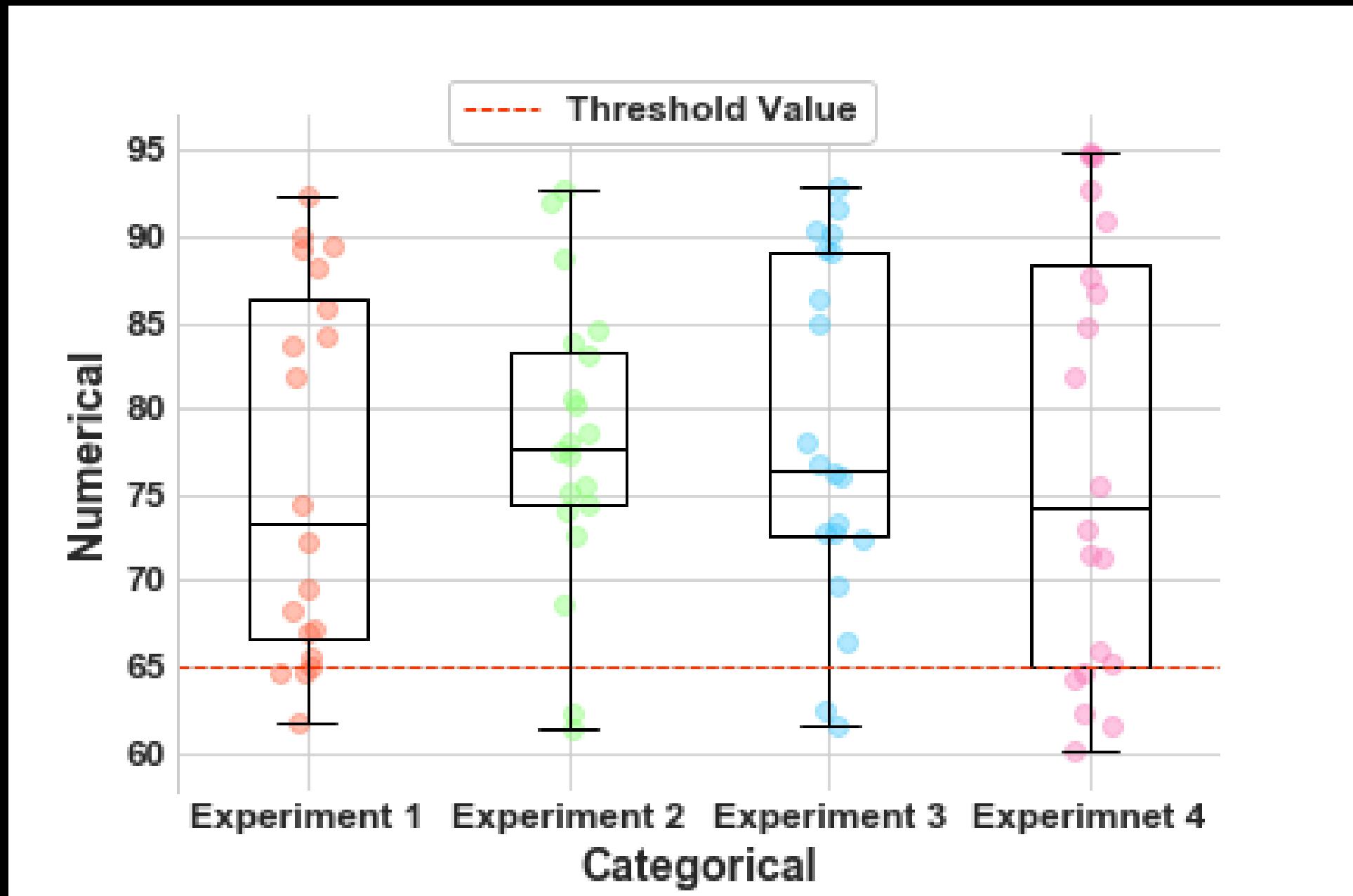
# Set the title and labels
plt.title('Histogram Example')
plt.xlabel('Bins')
plt.ylabel('Frequency')

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

EXAMPLE (9)



BOXPLOT



INTRODUCTION

A BOXPLOT IS A STANDARDIZED WAY OF DISPLAYING THE DISTRIBUTION OF DATA.

BASIC STRUCTURE

IT SHOWS PROPERTIES SUCH AS THE MINIMUM, FIRST QUARTILE, MEDIAN, THIRD QUARTILE, AND MAXIMUM. IN A BOXPLOT, A BOX IS CREATED FROM THE FIRST QUARTILE TO THE THIRD QUARTILE, A VERTICLE LINE IS ALSO THERE WHICH GOES THROUGH THE BOX AT THE MEDIAN.

UTILITY

BOXPLOTS ARE USED TO DISPLAY THE SUMMARY OF A SET OF DATA VALUES HAVING PROPERTIES LIKE MINIMUM, FIRST QUARTILE, MEDIAN, THIRD QUARTILE, AND MAXIMUM. THEY ARE PARTICULARLY USEFUL FOR IDENTIFYING OUTLIERS AND UNDERSTANDING THE DISTRIBUTION OF YOUR DATA.

EXAMPLE (10)

```
: import matplotlib.pyplot as plt

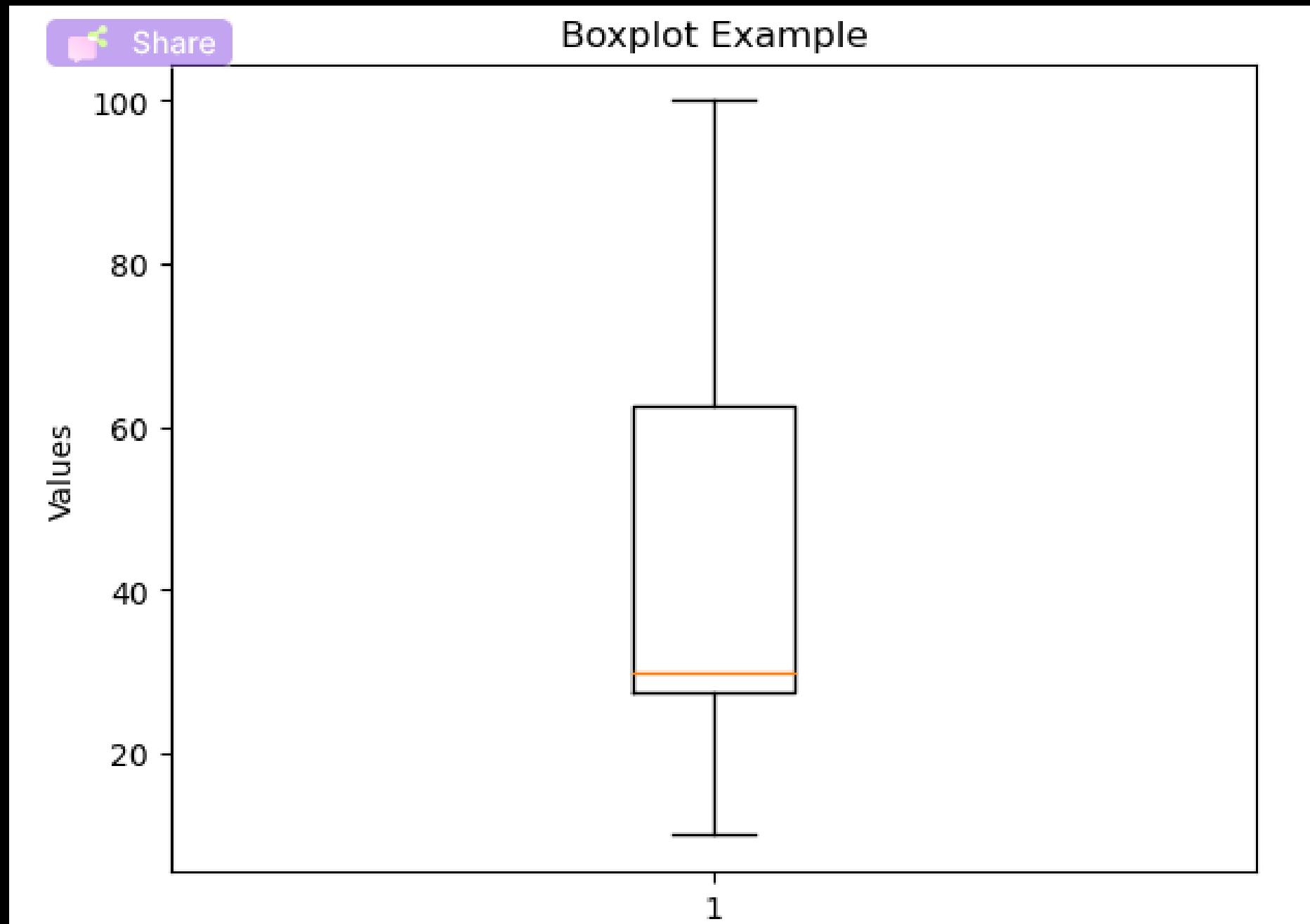
# Data
data = [10, 20, 20, 20, 30, 30, 30, 30, 30, 40, 50, 60, 70, 80, 90, 100]

# Create a Boxplot
plt.boxplot(data)

# Set the title and Labels
plt.title('Boxplot Example')
plt.ylabel('Values')

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

EXAMPLE (10)





CONTACTEZ - NOUS ! NOUS SERONS
RAVIS DE POUVOIR VOUS AIDER !



**ENVOYEZ - NOUS
UN MESSAGE !**