ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN
Vietnam - Korea University of Information and Communication Technology

Chapter 7
Data Visualization with Matplotlib

Thu Huong Nguyen, PhD
Si Thin Nguyen, PhD

KHMT
Computer Science

http://vku.udn.vn/

About Authors

Thu Huong Nguyen

PhD in Computer Science at Université Côte d'Azur, France

Email: nthuong@vku.udn.vn

Address: Faculty of Computer Science, VKU

Si Thin Nguyen

PhD in Computer Science at Soongsil University, Korea

Email: nsthin@vku.udn.vn

Address: Faculty of Computer Science, VKU

Chapter Content

- Introduction
- Install
- Plot Types
- Line Plot
- Bar Chart
- Histogram
- Scatter Diagram

Introduction

➔ **Data Visualization:**

- is the process of presenting data in the form of graphs or charts.
- is also used in high-level data analysis for Machine Learning and Exploratory Data Analysis (EDA)

➔ **Matplotlib**

- is a library of Python which is used for data visualization.
- is easy to use and emulates MATLAB like graphs and visualization.

Install

➔ **Step 1 – Make sure Python and pip is preinstalled on your system:**

- Check Python : “python --version” ;
- Check pip : “pip -V”

➔ **Step 2** – Install Matplotlib

- Command : “pip install matplotlib”

➔ **Step 3 -** Check if it is installed successfully

- Command : “import matplotlib

Matplotlib.__version__”

Install

➔ **Step 1 – Make sure Python and pip is preinstalled on your system:**

- Check Python : “python --version” ;
- Check pip : “pip -V”

➔ **Step 2** – Install Matplotlib

- Command : “pip install matplotlib”

➔ **Step 3 -** Check if it is installed successfully

- Command : “import matplotlib

Matplotlib.__version__”

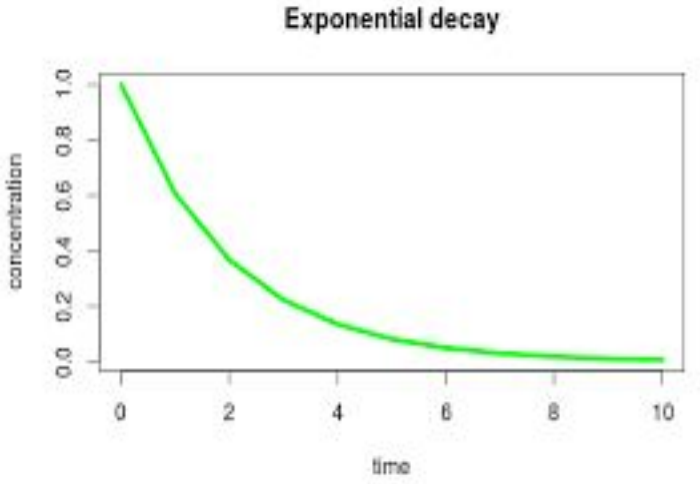
Plot Types

- Line Plot
- Bar Plot
- Histogram
- Scatter Plot
- Pie Chart
- Violin Chart
- Box Plot

Line Plot

➔ The method **plot()** creates a line plot

```
import matplotlib.pyplot as plt

x_axis = ['value_1', 'value_2', 'value_3', ...]
y_axis = ['value_1', 'value_2', 'value_3', ...]

plt.plot(x_axis, y_axis)
plt.title('title name')
plt.xlabel('x_axis name')
plt.ylabel('y_axis name')
plt.show()
```

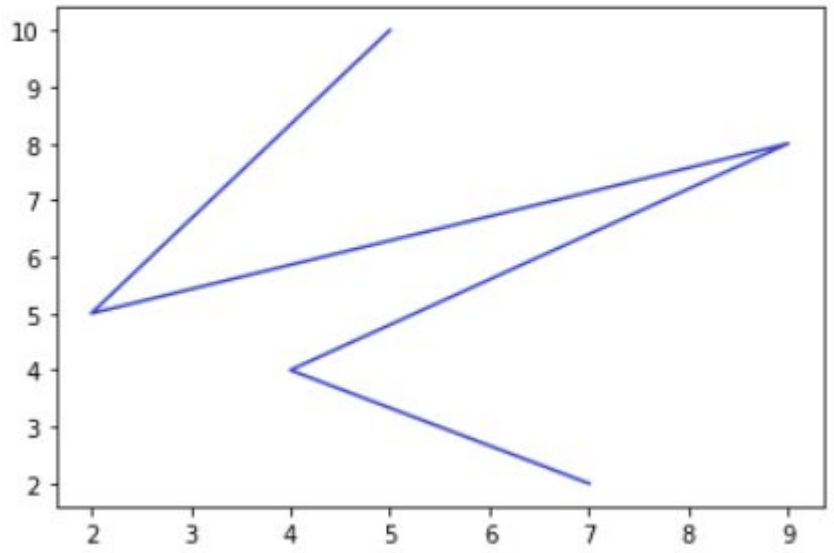


Line Plot

➔ Create your Line chart using lists

```
# importing matplotlib module
from matplotlib import pyplot as plt

# x-axis values
x = [5, 2, 9, 4, 7]

# Y-axis values
y = [10, 5, 8, 4, 2]

# Function to plot
plt.plot(x, y)

# function to show the plot
plt.show()
```

Output:



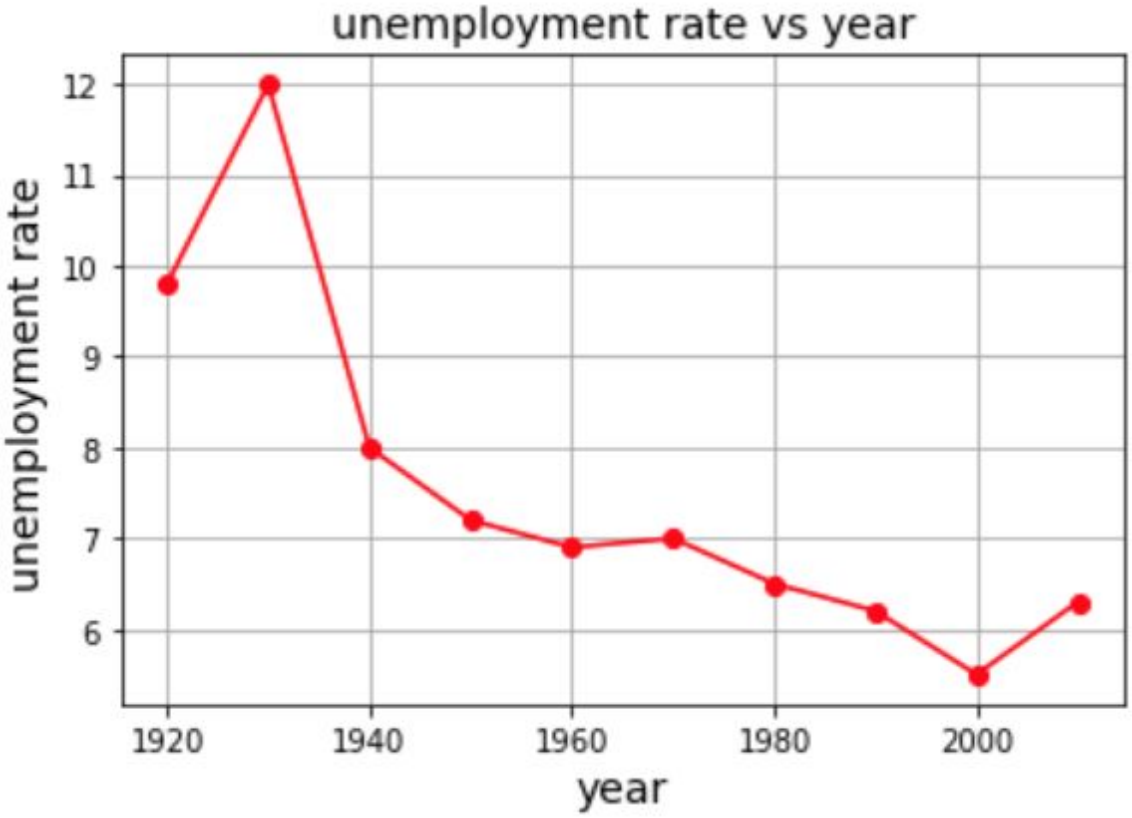
Line Plot

➔ Create line plot using Pandas DataFrame.

year	unemployment_rate
1920	9.8
1930	12
1940	8
1950	7.2
1960	6.9
1970	7
1980	6.5
1990	6.2

```
import pandas as pd
import matplotlib.pyplot as plt

data = {'year': [1920, 1930, 1940, 1950, 1960, 1970, 1980, 1990, 2000, 2010],
        'unemployment_rate': [9.8, 12, 8, 7.2, 6.9, 7, 6.5, 6.2, 5.5, 6.3]
       }

df = pd.DataFrame(data)

plt.plot(df['year'], df['unemployment_rate'], color='red', marker='o')
plt.title('unemployment rate vs year', fontsize=14)
plt.xlabel('year', fontsize=14)
plt.ylabel('unemployment rate', fontsize=14)
plt.grid(True)
plt.show()
```

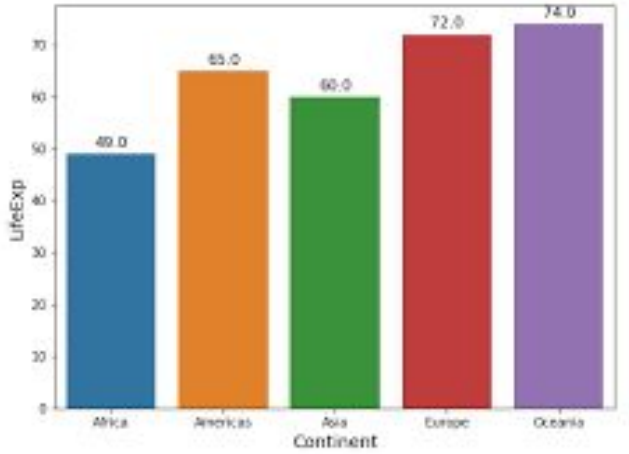
Line Plot


unemployment rate vs year
unemployment rate
year
12
11
10
9
8
7
6
1920
1940
1960
1980
2000

Bar Chart

- The bar graph is the graphical representation of **categorical data.**
- The method **bar()** creates a bar chart

```
import matplotlib.pyplot as plt

x_axis = ['value_1', 'value_2', 'value_3', ...]
y_axis = ['value_1', 'value_2', 'value_3', ...]

plt.bar(x_axis, y_axis)
plt.title('title name')
plt.xlabel('x_axis name')
plt.ylabel('y_axis name')
plt.show()
```

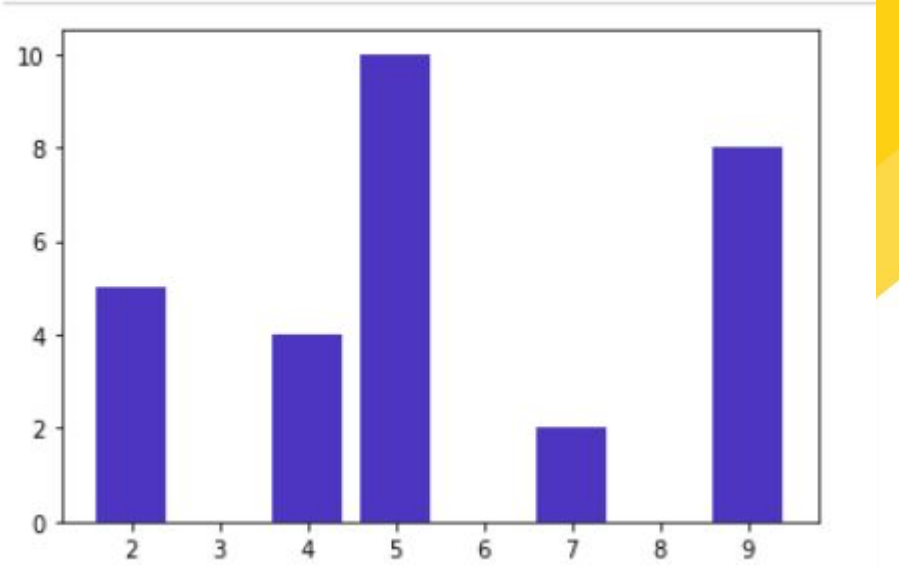


Bar Chart

➔ Create your bar chart using lists

```
# importing matplotlib module
from matplotlib import pyplot as plt

# x-axis values
x = [5, 2, 9, 4, 7]

# Y-axis values
y = [10, 5, 8, 4, 2]

# Function to plot
plt.bar(x, y)

# function to show the plot
plt.show()
```

Output:



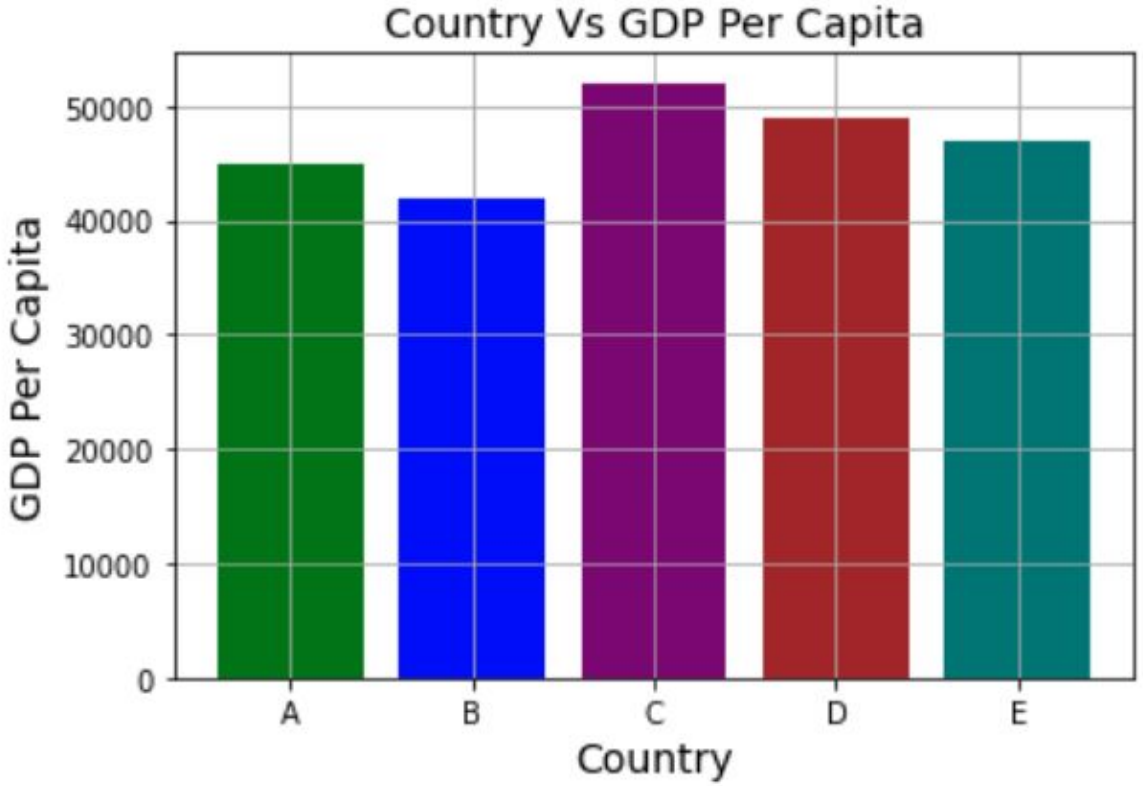
Bar Chart

➔ Create bar chart using Pandas DataFrame.

country	gdp per_ capita
A	45000
B	42000
C	52000
D	49000
E	47000

```
import matplotlib.pyplot as plt
import pandas as pd

data = {'country': ['A', 'B', 'C', 'D', 'E'],
        'gdp_per_capita': [45000, 42000, 52000, 49000, 47000]
       }
df = pd.DataFrame(data)

colors = ['green','blue','purple','brown','teal']
plt.bar(df['country'], df['gdp_per_capita'], color=colors)
plt.title('Country Vs GDP Per Capita', fontsize=14)
plt.xlabel('Country', fontsize=14)
plt.ylabel('GDP Per Capita', fontsize=14)
plt.grid(True)
plt.show()
```

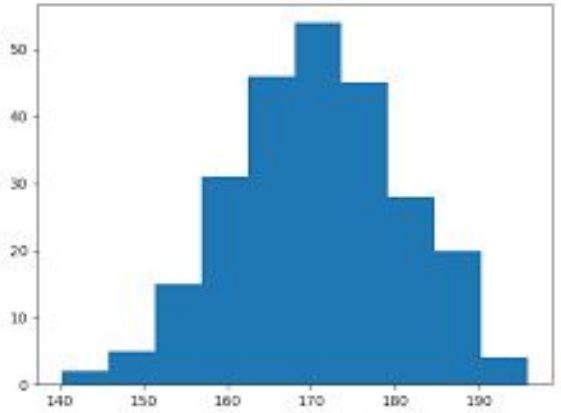
Bar Chart


Country Vs GDP Per Capita
GDP Per Capita
50000
40000
30000
20000
10000
0
A
B
C
D
E
Country

Histogram

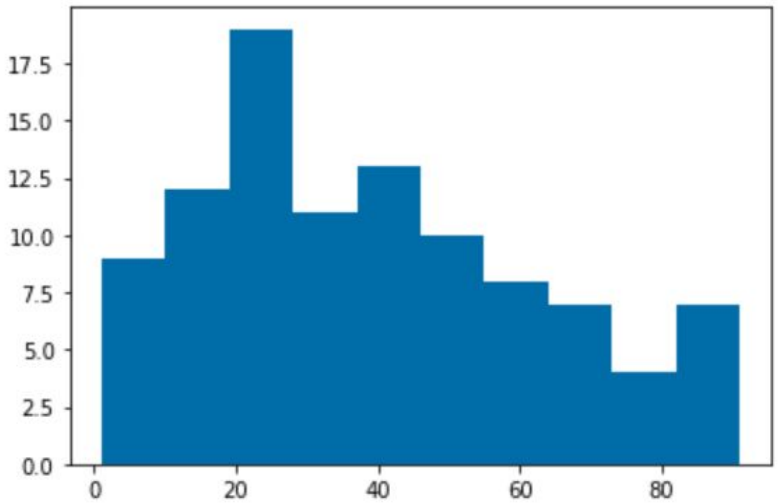
- A histogram is the graphical representation of **quantitative data.**
- The method **hist()** creates a histogram

```
import matplotlib.pyplot as plt

x = [value1, value2, value3,....]
plt.hist(x, bins=number of bins)
plt.show()
```



Histogram

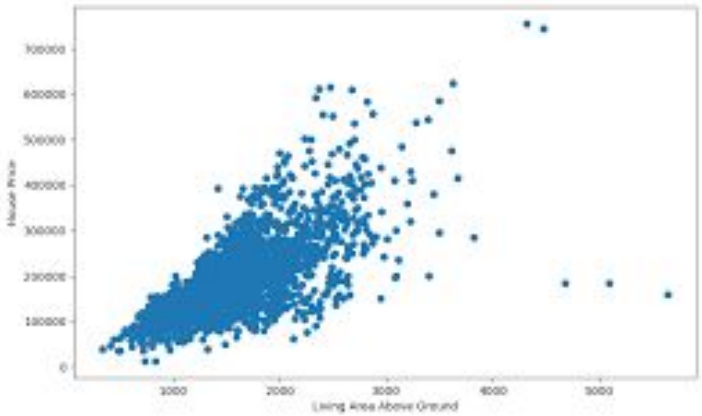
```
import matplotlib.pyplot as plt

x = [1, 1, 2, 3, 3, 5, 7, 8, 9, 10,
     10, 11, 11, 13, 13, 15, 16, 17, 18, 18,
     18, 19, 20, 21, 21, 23, 24, 24, 25, 25,
     25, 25, 26, 26, 26, 27, 27, 27, 27, 27,
     29, 30, 30, 31, 33, 34, 34, 34, 35, 36,
     36, 37, 37, 38, 38, 39, 40, 41, 41, 42,
     43, 44, 45, 45, 46, 47, 48, 48, 49, 50,
     51, 52, 53, 54, 55, 55, 56, 57, 58, 60,
     61, 63, 64, 65, 66, 68, 70, 71, 72, 74,
     75, 77, 81, 83, 84, 87, 89, 90, 90, 91
     ]

plt.hist(x, bins=10)
plt.show()
```



Scatter Diagram

➔ The method **scatter()** creates a scatter chart

```
import matplotlib.pyplot as plt

x_axis = ['value_1', 'value_2', 'value_3', ...]
y_axis = ['value_1', 'value_2', 'value_3', ...]

plt.scatter(x_axis, y_axis)
plt.title('title name')
plt.xlabel('x_axis name')
plt.ylabel('y_axis name')
plt.show()
```



Scatter Diagram

➔ Create your scatter diagram using lists

```
import matplotlib.pyplot as plt

unemployment_rate = [6.1, 5.8, 5.7, 5.7, 5.8, 5.6, 5.5, 5.3, 5.2, 5.2]
index_price = [1500, 1520, 1525, 1523, 1515, 1540, 1545, 1560, 1555, 1565]

plt.scatter(unemployment_rate, index_price, color='green')
plt.title('Unemployment Rate Vs Index Price', fontsize=14)
plt.xlabel('Unemployment Rate', fontsize=14)
plt.ylabel('Index Price', fontsize=14)
plt.grid(True)
plt.show()
```

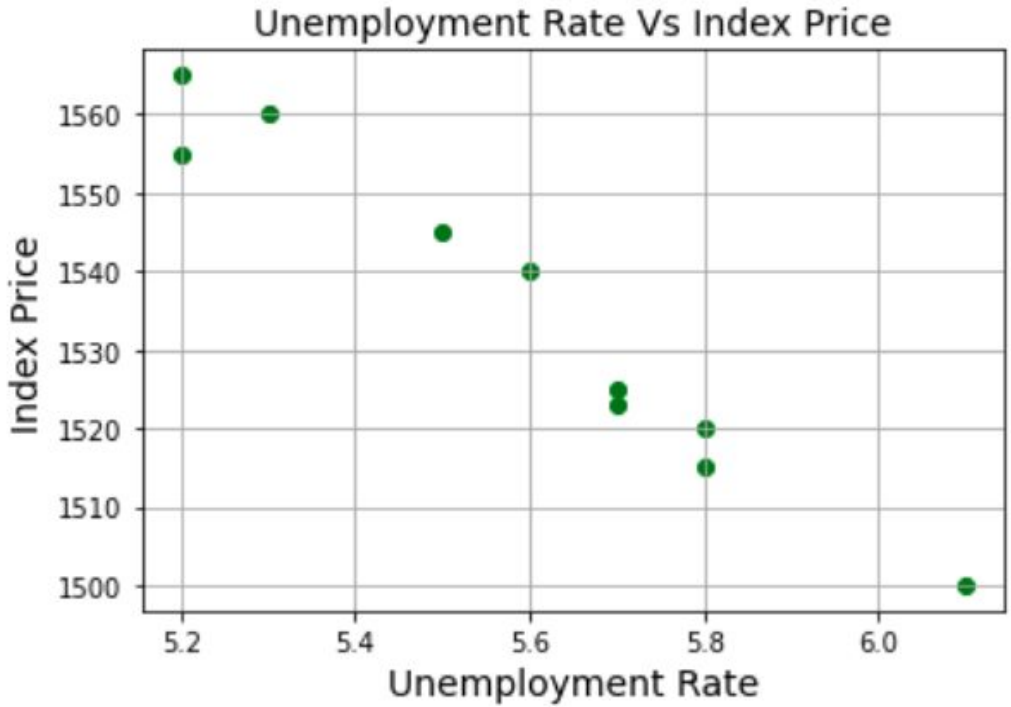
unemployment_rate	index_price
6.1	1500
5.8	1520
5.7	1525
5.7	1523
5.8	1515
5.6	1540
5.5	1545
5.3	1560
5.2	1555
5.2	1565

Scatter Diagram

➔ Create a scatter diagram using Dataframe

```
import pandas as pd
import matplotlib.pyplot as plt

data = {'unemployment_rate': [6.1, 5.8, 5.7, 5.7, 5.8, 5.6, 5.5, 5.3, 5.2, 5.2],
        'index_price': [1500, 1520, 1525, 1523, 1515, 1540, 1545, 1560, 1555, 1565]
        }

df = pd.DataFrame(data)

plt.scatter(df['unemployment_rate'], df['index_price'], color='green')
plt.title('Unemployment Rate Vs Index Price', fontsize=14)
plt.xlabel('Unemployment Rate', fontsize=14)
plt.ylabel('Index Price', fontsize=14)
plt.grid(True)
plt.show()
```

Scatter Diagram



Adding title and Labeling the Axes in the graph

➔ Add Title:

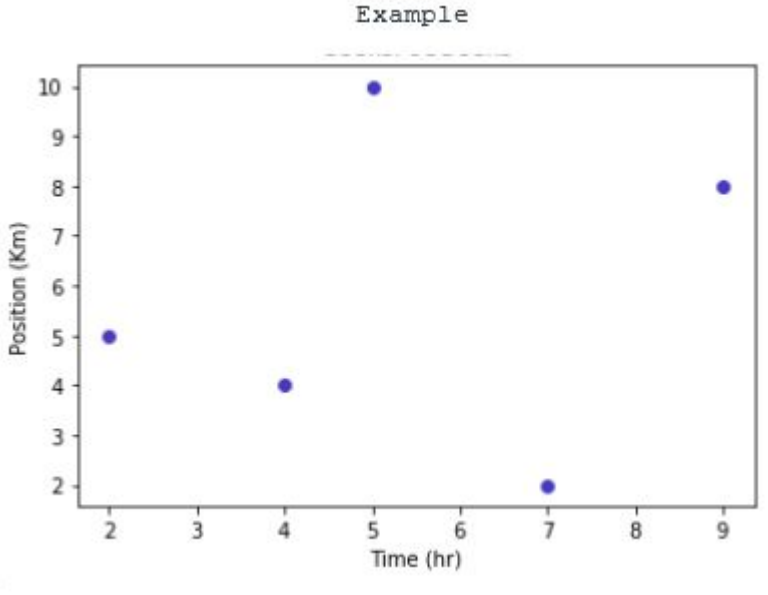
“matplotlib.pyplot.title("My title")”

➔ Label the x-axis and y-axis :

“matplotlib.pyplot.xlabel("Time (Hr)")
matplotlib.pyplot.ylabel("Position (Km)")”

Example

```
# x-axis values
x = [5, 2, 9, 4, 7]

# Y-axis values
y = [10, 5, 8, 4, 2]

# Function to plot
plt.scatter(x, y)

# Adding Title
plt.title("Example")

# Labeling the axes
plt.xlabel("Time (hr)")
plt.ylabel("Position (Km)")

# function to show the plot
plt.show()
```



c

Multiple Graphs

- **Multiple Graphs:** by repeating the **show()** function or use a function called **subplot()** in order to print them horizontally as well.
- Example:

```
from matplotlib import pyplot as plt


x = [1, 2, 3, 4, 5]
y = [1, 4, 9, 16, 25]
plt.scatter(x, y)

# function to show the plot
plt.show()

plt.plot(x, y)

# function to show the plot
plt.show()
```