

MatPlotLib

Introduction to MatplotLib

- Matplotlib is a Python library used for creating static, animated, and interactive visualizations.
- Installation:
  - pip install matplotlib

- Example:

```
import matplotlib.pyplot as plt
plt.plot([5,2,3], [4,5,6])
plt.show()
```

Types of Plots:

- Line plot
- Bar chart
- Histogram
- Pie chart
- Scatter plot
- Box plot

Plot Function()

Syntax: plt.plot(x, y, format_string, **kwargs)

Main Parameters:

1. **x, y** → Data points (lists, arrays).
   - `x` : values on X-axis
   - `y` : values on Y-axis
2. **fmt (format string)** → Optional shorthand for color, marker, and line style.

   Example: `'r--o'` = red dashed line with circle markers.

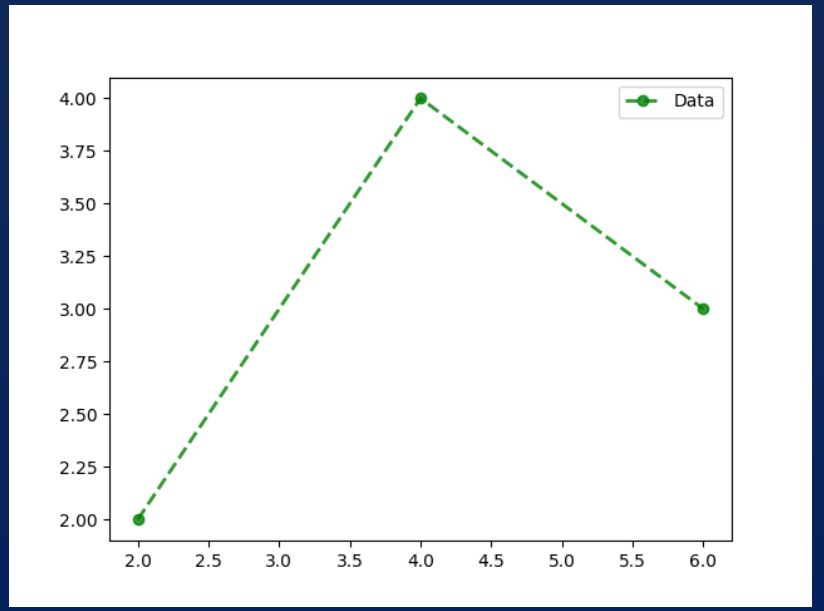
Plot Function() Continue.

🎨 Style Parameters (kwargs):

3. **color / c** → Line color (`'r'`, `'blue'`, `'#00FF00'`).
4. **linestyle / ls** → Line type (`'-'`, `'--'`, `'-.'`, `':'`, `'none'`).
5. **linewidth / lw** → Line thickness (e.g., `2` or `2.5`).
6. **marker** → Shape for data points (`'o'`, `'s'`, `'*'`, `'x'`, etc.).
7. **markersize / ms** → Size of markers.
8. **markeredgecolor / mec** → Color of marker border.
9. **markerfacecolor / mfc** → Fill color of marker.
10. **label** → Label for legend.
11. **alpha** → Transparency (0.0–1.0).
12. **antialiased / aa** → Smooth edges (`True` / `False`).
13. **drawstyle** → How line segments are drawn (`'default'`, `'steps'`, `'steps-pre'`, etc.).
14. **solid_capstyle / solid_joinstyle** → Line cap and join shapes (`'round'`, `'butt'`, `'miter'`, etc.).
15. **zorder** → Drawing order (higher = on top).

Example:

```
import matplotlib.pyplot as plt
x=[2,4,6]
y=[2,4,3]
plt.plot(x, y, color='green', linestyle='--', marker='o',
         linewidth=2, markersize=6, label='Data', alpha=0.8)
plt.legend()
plt.show()
```



Visualization in Pandas polt() function

- plot() in Pandas is used to create different types of plots directly from **DataFrames** or **Series** using **Matplotlib** in the background.
- Syntax:
  - DataFrame.plot(kind='line', **kwargs)
  - Series.plot(kind='line', **kwargs)

Common Parameters

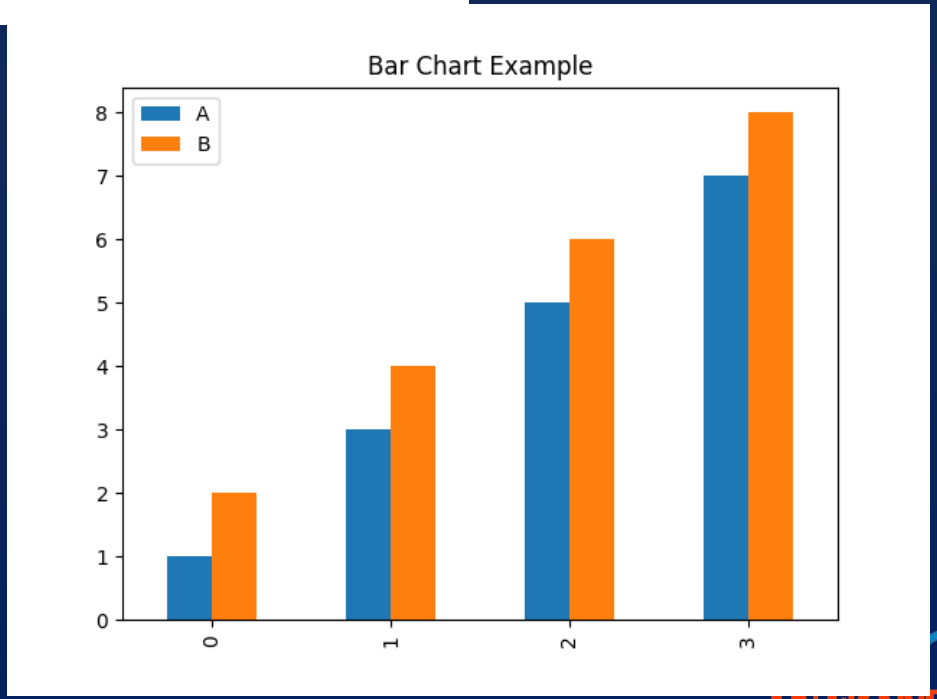
Parameter	Description	Example
kind	Type of plot (e.g. 'line', 'bar', 'hist', 'pie')	kind='bar'
x	Column name for x-axis	x='year'
y	Column name(s) for y-axis	y='sales' or y=['A','B']
color	Color(s) of the lines/bars	color=['red','blue']
title	Title of the plot	title='Sales Growth'
xlabel / ylabel	Axis labels	xlabel='Year'
legend	Show or hide legend	legend=True
grid	Show gridlines	grid=True
style	Line/marker style	style='--o'
figsize	Figure size (width, height)	figsize=(8,5)
alpha	Transparency (0.0–1.0)	alpha=0.7

Common Plot Types (kind values)

kind	Plot Type	Equivalent Matplotlib Function
'line'	Line plot	plt.plot()
'bar'	Vertical bar chart	plt.bar()
'barh'	Horizontal bar chart	plt.barh()
'hist'	Histogram	plt.hist()
'box'	Box plot	plt.boxplot()
'area'	Area chart	plt.stackplot()
'pie'	Pie chart	plt.pie()
'scatter'	Scatter plot	plt.scatter()
'hexbin'	Hexbin plot	plt.hexbin()
'kde' or 'density'	Density plot	plt.plot() (KDE line)

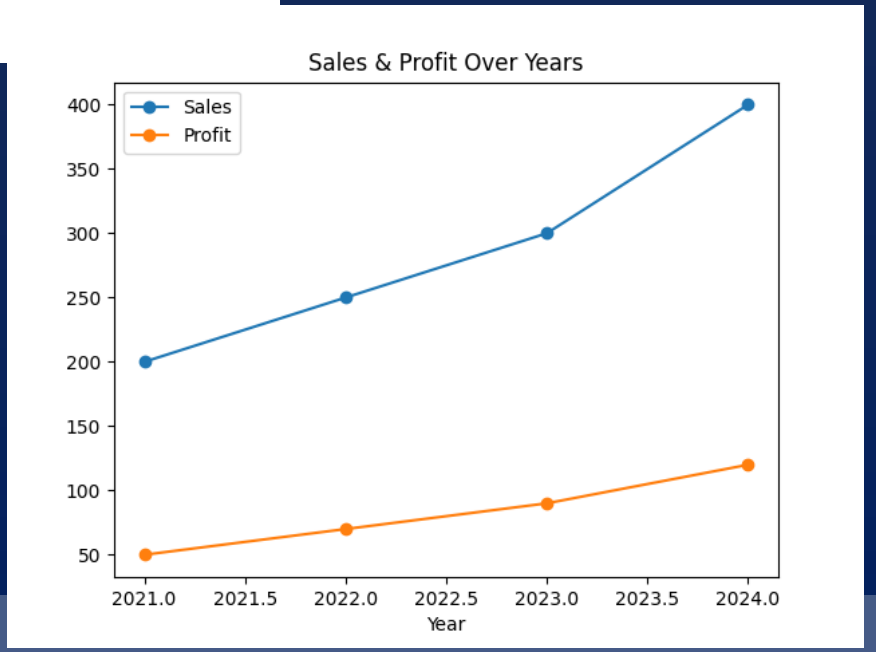
Example:

```
import pandas as pd
import matplotlib.pyplot as plt
data = {'A': [1, 3, 5, 7], 'B': [2, 4, 6, 8]}
df = pd.DataFrame(data)
df.plot(kind='bar', title='Bar Chart Example')
plt.show()
```



Example 2

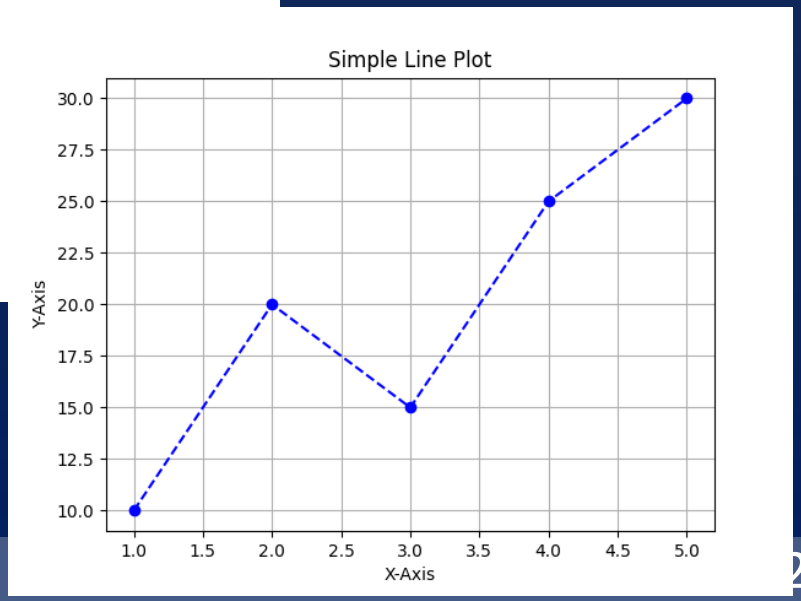
```
import pandas as pd
import matplotlib.pyplot as plt
data = {'Year': [2021, 2022, 2023, 2024],
        'Sales': [200, 250, 300, 400],
        'Profit': [50, 70, 90, 120]}
df = pd.DataFrame(data)
df.plot(x='Year', y=['Sales', 'Profit'], kind='line',
        marker='o', title='Sales & Profit Over Years')
plt.show()
```



Line Plot/Chart

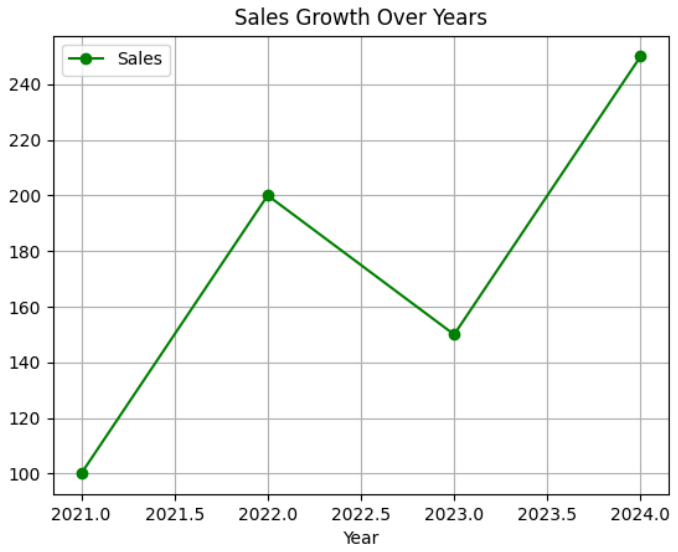
- A **line plot** shows data points connected by straight lines — used to visualize trends or changes over time.
- Using Matplotlib

```
import matplotlib.pyplot as plt
x = [1, 2, 3, 4, 5]
y = [10, 20, 15, 25, 30]
plt.plot(x, y, color='blue', marker='o', linestyle='--')
plt.title("Simple Line Plot")
plt.xlabel("X-Axis")
plt.ylabel("Y-Axis")
plt.grid(True)
plt.show()
```



Using Pandas

```
import pandas as pd
import matplotlib.pyplot as plt
data = {'Year': [2021, 2022, 2023, 2024],
        'Sales': [100, 200, 150, 250]}
df = pd.DataFrame(data)
df.plot(x='Year', y='Sales', kind='line', marker='o', color='green',
        title='Sales Growth Over Years', grid=True)
plt.show()
```



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

# Create DataFrame
df = pd.DataFrame({
    'Course': ['Python', 'Java', 'Web Design', 'Data Science'],
    'Students': [120, 90, 60, 30]
})

# Plot Pie Chart
df.plot(
    kind='pie',
    y='Students',
    labels=df['Course']
)

plt.show()
```

