

# Chapter 3 · Data Visualization with Matplotlib

CBSE Class 12 Informatics Practices (065) · Unit 1 = 25 marks · Ultimate Revision Sheet

## 60-SECOND RECALL

|                         |                                                                                        |
|-------------------------|----------------------------------------------------------------------------------------|
| Import                  | <code>import matplotlib.pyplot as plt</code>                                           |
| Only 3 plots prescribed | Line plot · Bar graph · Histogram                                                      |
| The three commands      | <code>plt.plot()</code> · <code>plt.bar()</code> · <code>plt.hist()</code>             |
| Customising             | <code>xlabel()</code> <code>ylabel()</code> <code>title()</code> <code>legend()</code> |
| Must end with           | <code>plt.show()</code>                                                                |
| Save                    | <code>plt.savefig("chart.png")</code> — before <code>show()</code>                     |

## 1. Purpose of Plotting

**WHY VISUALISE** Data visualisation presents data in a *graphical form* so that patterns, trends and outliers become immediately visible. Benefits: easier and faster *comparison*, quick identification of *trends*, better *decision making*, and clearer *communication* of results than raw numbers allow.

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

## 2. Line Plot

**WHEN TO USE** A *line plot* shows how a value *changes over a continuous range* — typically over *time*. Use it for trends.

```
x = [1, 2, 3, 4, 5]
y = [10, 25, 18, 40, 35]

plt.plot(x, y)
plt.xlabel("Month")
plt.ylabel("Sales")
plt.title("Monthly Sales")
plt.show()
```

### CUSTOMISING A LINE

```
plt.plot(x, y,
         color='red',          # or 'r', '#FF0000'
         linewidth=2,
         linestyle='dashed',   # 'solid' 'dotted' 'dashdot'
         marker='o',          # point marker
         markersize=8,
         label='Sales 2026')

plt.legend()                  # shows the label
plt.grid(True)
```

**Shorthand:** `plt.plot(x, y, 'r--o')` means red, dashed, circle markers.

| Marker | Symbol | Linestyle                       | Symbol |
|--------|--------|---------------------------------|--------|
| Circle | 'o'    | Solid                           | '-'    |
| Point  | '.'    | Dashed                          | '--'   |
| Star   | '*'    | Dotted                          | '...'  |
| Plus   | '+'    | Dash-dot                        | '-.'   |
| Cross  | 'x'    | <b>Colours:</b> r g b c m y k w |        |

### 3. Bar Graph

**WHEN TO USE** A bar graph compares values across discrete categories. Each bar's height (or length) represents the value.

```
subjects = ['IP', 'Maths', 'English', 'Eco']
marks = [92, 78, 85, 70]

plt.bar(subjects, marks, color='skyblue', width=0.5)
plt.xlabel("Subject")
plt.ylabel("Marks")
plt.title("Marks by Subject")
plt.show()
```

#### VARIATIONS

```
plt.barh(subjects, marks)           # HORIZONTAL bar graph

# Multiple bars side by side
x = np.arange(len(subjects))
plt.bar(x - 0.2, marks1, width=0.4, label='Term 1')
plt.bar(x + 0.2, marks2, width=0.4, label='Term 2')
plt.xticks(x, subjects)
plt.legend()
plt.show()
```

**TRAP** `plt.bar()` gives vertical bars; `plt.barh()` gives horizontal bars. Note also that `width` controls bar thickness in `bar()`, but the equivalent in `barh()` is `height`.

### 4. Histogram

**WHEN TO USE** A histogram shows the frequency distribution of continuous data by grouping values into bins (intervals). It answers "how many values fall in each range?"

```
data = [12, 15, 18, 22, 25, 28, 30, 33, 35, 38, 41, 45]

plt.hist(data, bins=5, color='green', edgecolor='black')
plt.xlabel("Marks")
plt.ylabel("Number of Students")
plt.title("Marks Distribution")
plt.show()
```

#### USEFUL PARAMETERS

```
plt.hist(data,
        bins=[0,20,40,60,80,100], # explicit bin edges
        color='green',
        edgecolor='black',
        cumulative=False,
        histtype='bar') # 'step' 'stepfilled'
```

#### BAR GRAPH vs HISTOGRAM — asked almost every year

| Basis                 | BAR GRAPH                     | HISTOGRAM                      |
|-----------------------|-------------------------------|--------------------------------|
| Data type             | Categorical / discrete        | Continuous / numerical         |
| X-axis                | Categories (names)            | Bins (ranges)                  |
| Gaps between bars     | YES — bars are separated      | NO — bars touch each other     |
| Bars can be reordered | Yes — order is arbitrary      | No — bins follow numeric order |
| Shows                 | Comparison between categories | Frequency distribution         |

## 5. Customising Any Plot

#### THE STANDARD BLOCK — write all of these

```
plt.xlabel("X axis label")
plt.ylabel("Y axis label")
plt.title("Chart Title")
plt.legend() # needs label= in the plot call
plt.grid(True)
plt.xticks([1,2,3], ['A','B','C']) # custom tick labels
plt.xlim(0, 10) # axis range
plt.ylim(0, 100)

plt.savefig("chart.png") # SAVE before show()
plt.show() # DISPLAY - always last
```

**THE TWO ORDER TRAPS** 1. `plt.savefig()` must come **BEFORE** `plt.show()`. After `show()` the figure is cleared and the saved file will be blank.

2. `plt.legend()` only works if you passed `label=` inside the plotting call. A legend with no labels shows nothing.

## 6. Plotting Directly from a DataFrame

```
import pandas as pd
df = pd.DataFrame({'Subject': ['IP', 'Maths', 'Eco'],
                  'Marks': [92, 78, 70]})

plt.bar(df['Subject'], df['Marks'])
plt.show()

# or the built-in Pandas plot
df.plot(kind='bar', x='Subject', y='Marks')
plt.show()
```

kind can be 'line', 'bar', 'barh', 'hist'.

## 7. Predict the Output / Identify the Error

**Q1 - what does this produce?** `plt.plot([1,2,3],[2,4,6], 'g--*')`

**Q2 - why is the saved image blank?** `plt.plot(x,y)`  
`plt.show()`  
`plt.savefig("a.png")`

**Q3 - why does no legend appear?** `plt.plot(x, y)`  
`plt.legend()`

**Q4 - which chart for this data?** Heights of 100 students grouped into ranges

## 8. Task Triggers wording → which chart / command?

| If the question says...                           | Use                                                              |
|---------------------------------------------------|------------------------------------------------------------------|
| "show the trend of sales over 12 months"          | <b>Line plot</b> — <code>plt.plot()</code>                       |
| "compare marks of 5 students"                     | <b>Bar graph</b> — <code>plt.bar()</code>                        |
| "show how many students fall in each marks range" | <b>Histogram</b> — <code>plt.hist()</code>                       |
| "frequency distribution"                          | <b>Histogram</b>                                                 |
| "horizontal bars"                                 | <code>plt.barh()</code>                                          |
| "add a heading to the chart"                      | <code>plt.title()</code>                                         |
| "label the axes"                                  | <code>plt.xlabel()</code> and <code>plt.ylabel()</code>          |
| "show which line is which"                        | <code>label=</code> + <code>plt.legend()</code>                  |
| "dotted red line with circle markers"             | <code>plt.plot(x,y, 'r:o')</code>                                |
| "save the chart as an image"                      | <code>plt.savefig()</code> <b>before</b> <code>plt.show()</code> |
| "display the chart"                               | <code>plt.show()</code>                                          |
| "divide the data into 6 intervals"                | <code>plt.hist(data, bins=6)</code>                              |

### MOST-ASKED BOARD QUESTIONS

1. [3-4M] Write code to plot the given data as a line / bar / histogram with labels, title and legend.
2. [2M] Differentiate between a bar graph and a histogram.
3. [2M] Identify and correct the error in the given plotting code.
4. [1-2M] Which chart is suitable for the given data, and why?
5. [1M] Which function saves a chart to a file? → `plt.savefig()`
6. [1M] Which library is used for plotting? → **Matplotlib (pyplot)**

**THE HABIT THAT SCORES** — Every plotting answer must contain **five lines**: the import, the plot call, `xlabel`, `ylabel`, `title`, and `show()`. Marks are allotted separately for the labels and title — a correct chart with no labels loses half the question.