Noteskarts

Smart Notes. Bright Future.

COMPUTER APPLICATIONS IN PHARMACY

Unit 5 — Data Visualization with Matplotlib

Short Notes | Diagrams | Tables | Quick Concepts

For D.Pharm & B.Pharm Students

www.noteskarts.com

Get Noteskarts Prime — Full Access

Scan the QR code above with your phone camera to download Noteskarts Prime and unlock complete detailed notes, PYQ papers, MCQ tests & video lectures for D.Pharm, B.Pharm, GPAT & Pharmacy Competitive Exams.

Free Bonus — Noteskarts Puzzle Lab

Revise this exact topic through fun MCQ games — Rapid MCQ, Term Scramble, True or False & more. 100% free, no sign-up needed. Scan and play!

Introduction to Matplotlib

Matplotlib is Python's most widely used library for creating static, publication-quality plots and charts. In pharmacy and pharmaceutical research, visualizing data - such as a concentration-time curve or a dissolution profile - often reveals patterns that raw numbers alone cannot show.

Why Visualize Pharmaceutical Data?

- A graph shows trends, peaks, and rates of change at a glance
- Concentration-time curves reveal Cmax, Tmax, and elimination pattern visually
- Comparing formulations or batches is far easier with overlaid plots
- Visuals are essential for reports, research papers, and regulatory submissions

Installing & Importing Matplotlib

```
pip install matplotlib

import matplotlib.pyplot as plt   # 'plt' is the standard short alias
```

The Basic Plotting Workflow

```
plt.plot(x, y)        # 1. choose a plot type and pass the data
plt.xlabel('...')     # 2. label the x-axis
plt.ylabel('...')     # 3. label the y-axis
plt.title('...')      # 4. add a title
plt.show()            # 5. display the plot
```

pyplot vs Figure/Axes

plt.plot() (pyplot style) is quick for a single simple plot. For more control - e.g. multiple subplots - Matplotlib also offers an object-oriented style: fig, ax = plt.subplots() then ax.plot(...). Both are used throughout this unit.

Creating Line Plots

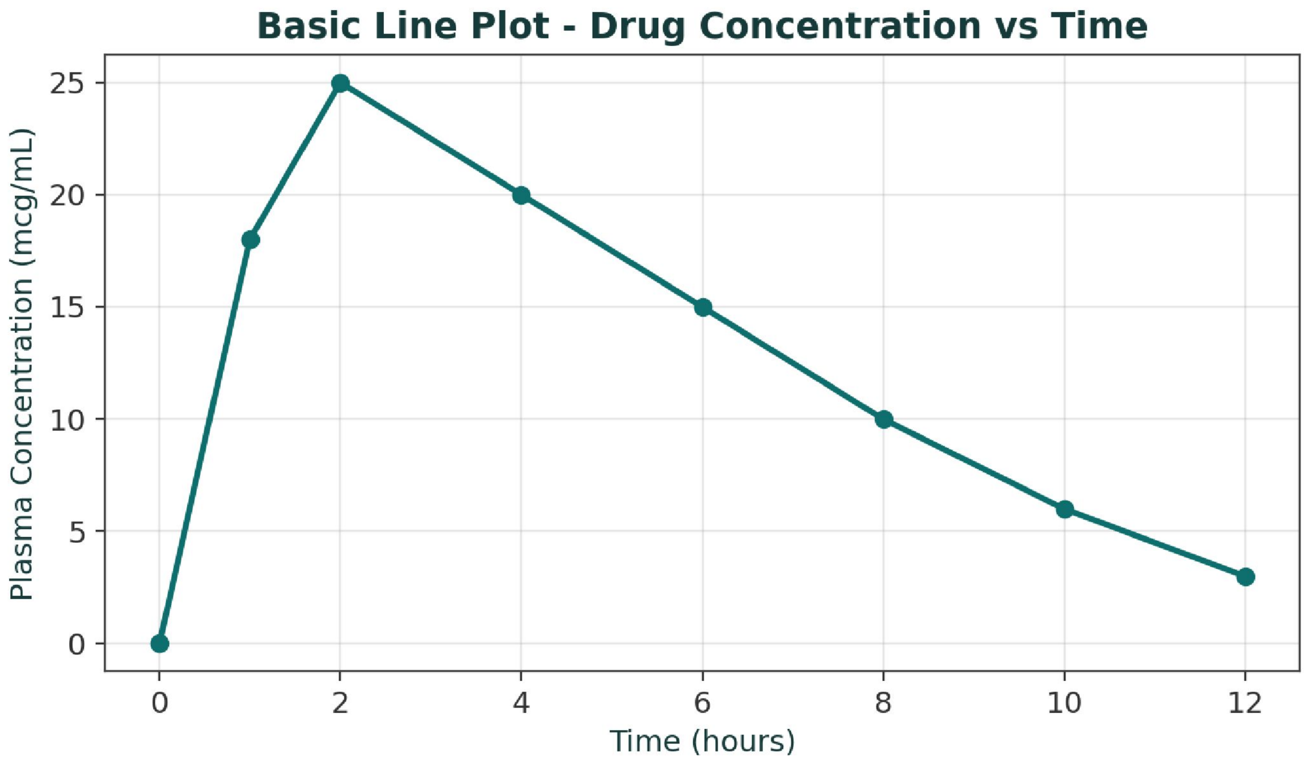
A line plot connects data points with a continuous line - ideal for showing how a value (like drug concentration) changes over a continuous variable (like time).

```
import matplotlib.pyplot as plt

time = [0, 1, 2, 4, 6, 8, 10, 12]          # hours
conc = [0, 18, 25, 20, 15, 10, 6, 3]       # mcg/mL

plt.plot(time, conc, color='teal', marker='o', linewidth=2)
plt.xlabel('Time (hours)')
plt.ylabel('Plasma Concentration (mcg/mL)')
plt.title('Drug Concentration vs Time')
```

```
plt.show()
```



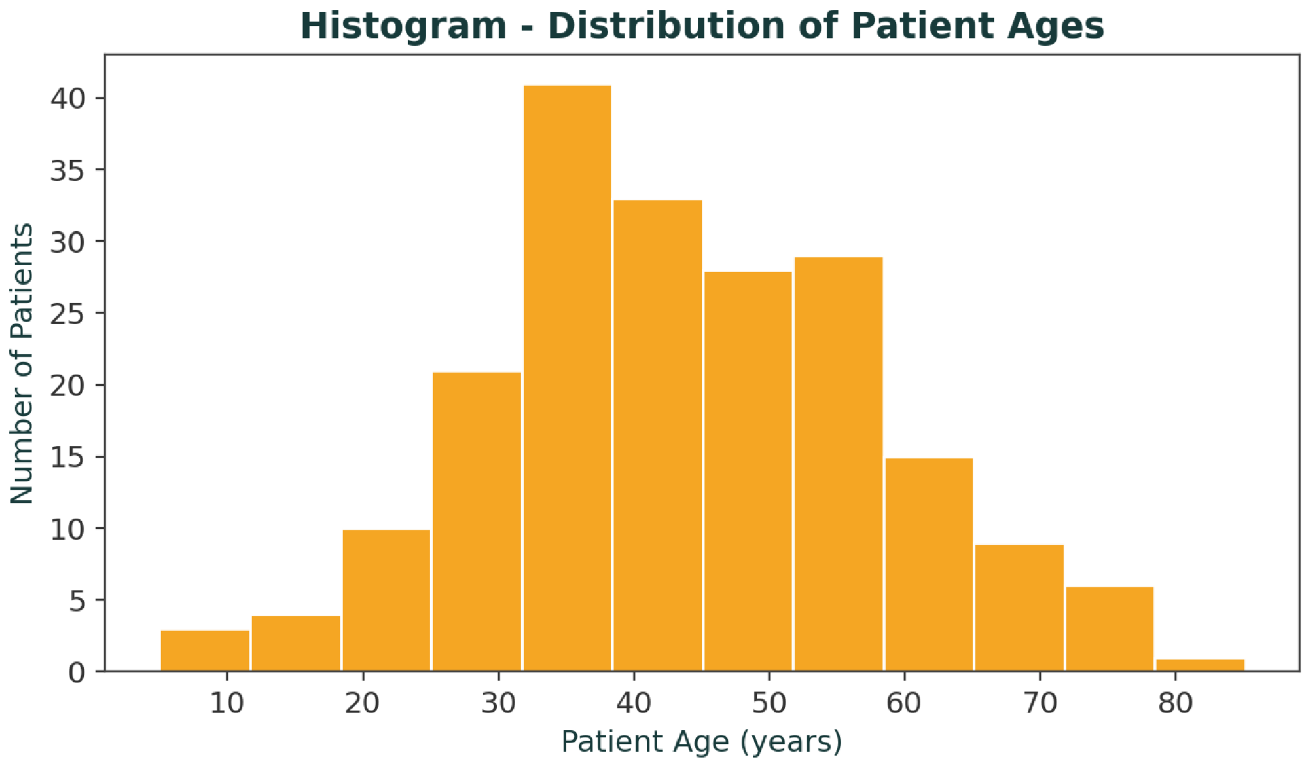
Parameter	Purpose
`color`	Line colour, e.g. 'teal', 'red', '#0F6E6E'
`marker`	Point style: 'o' circle, 's' square, '^' triangle
`linewidth / lw`	Thickness of the line
`linestyle / ls`	'-' solid, '--' dashed, ':' dotted

Creating Histograms

A histogram shows the frequency distribution of a single numeric variable by grouping values into 'bins' - useful for seeing how patient ages, doses, or lab values are spread out.

```
ages = [23, 45, 34, 67, 45, 52, 29, 61, 38, 44, 55, 48]

plt.hist(ages, bins=6, color='orange', edgecolor='white')
plt.xlabel('Patient Age (years)')
plt.ylabel('Number of Patients')
plt.title('Distribution of Patient Ages')
plt.show()
```



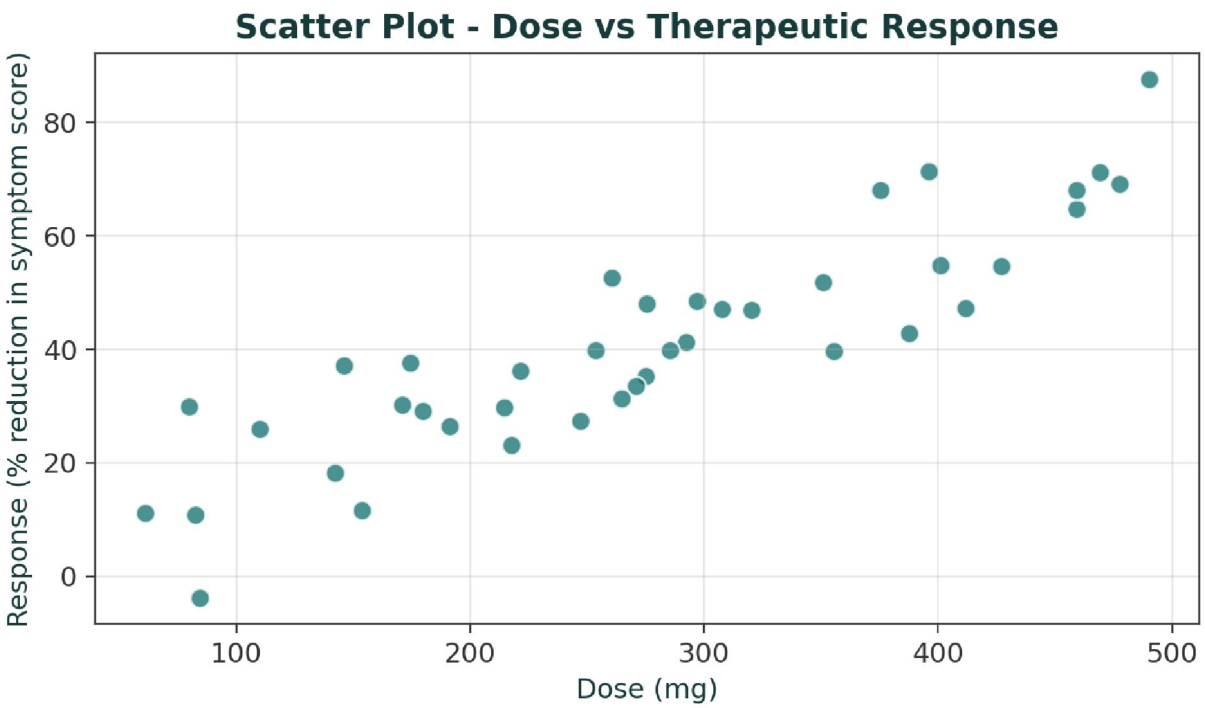
> **Choosing bins**
> The bins parameter controls how many groups the data is split into. Too few bins hides detail; too many makes the chart noisy. bins=10-15 is a reasonable starting point for most datasets.

Creating Scatter Plots

A scatter plot shows individual data points for two variables, revealing whether a relationship (correlation) exists between them - e.g. dose vs therapeutic response.

```
dose = [50, 100, 150, 200, 250, 300, 350, 400]
response = [8, 18, 25, 33, 38, 44, 50, 55]

plt.scatter(dose, response, color='teal', alpha=0.75, s=60)
plt.xlabel('Dose (mg)')
plt.ylabel('Response (% reduction in symptom score)')
plt.title('Dose vs Therapeutic Response')
plt.show()
```



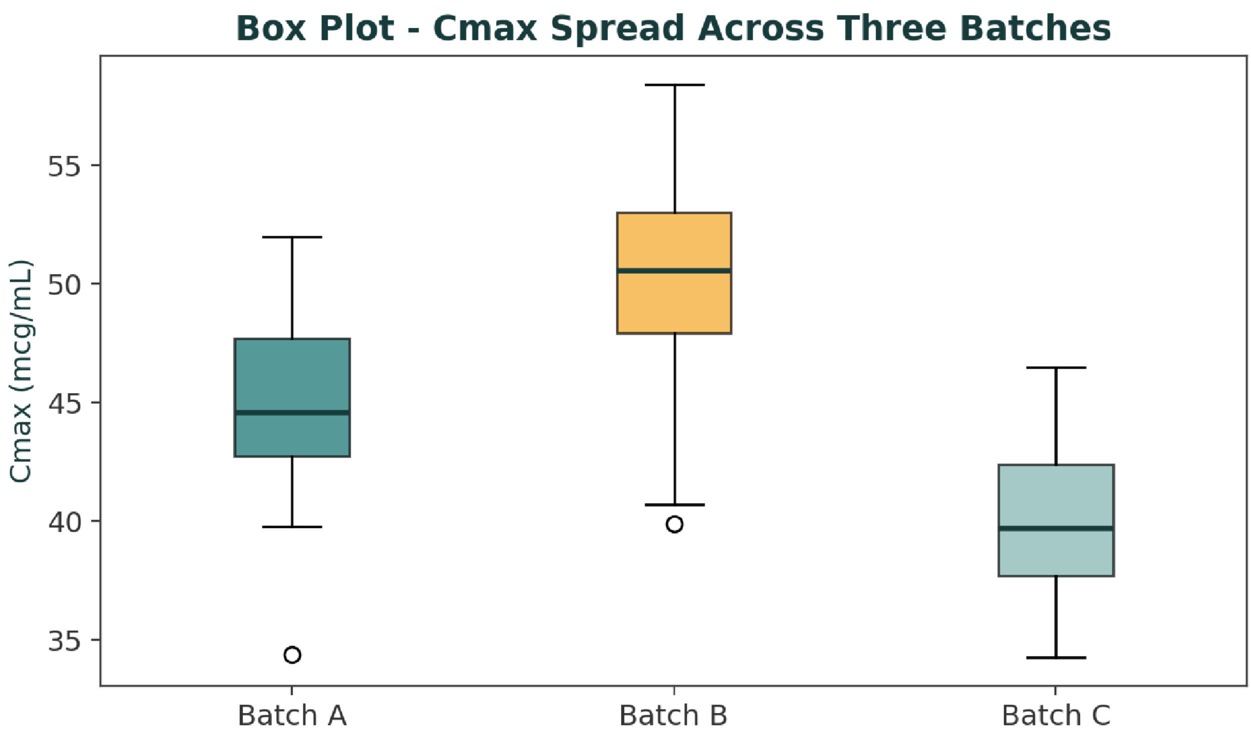
Parameter	Purpose
`s`	Marker (dot) size
`alpha`	Transparency (0 = invisible, 1 = solid) - helps see overlapping points
`c / color`	Colour of the points

Creating Box Plots

A box plot (box-and-whisker plot) summarises a dataset's spread using its median, quartiles, and outliers - excellent for comparing Cmax or dissolution values across multiple drug batches at once.

```
batchA = [42, 45, 44, 47, 43, 46, 45]
batchB = [50, 55, 48, 52, 58, 47, 53]
batchC = [39, 41, 40, 38, 42, 40, 41]

plt.boxplot([batchA, batchB, batchC], tick_labels=['Batch A','Batch
B','Batch C'])
plt.ylabel('Cmax (mcg/mL)')
plt.title('Cmax Spread Across Three Batches')
plt.show()
```



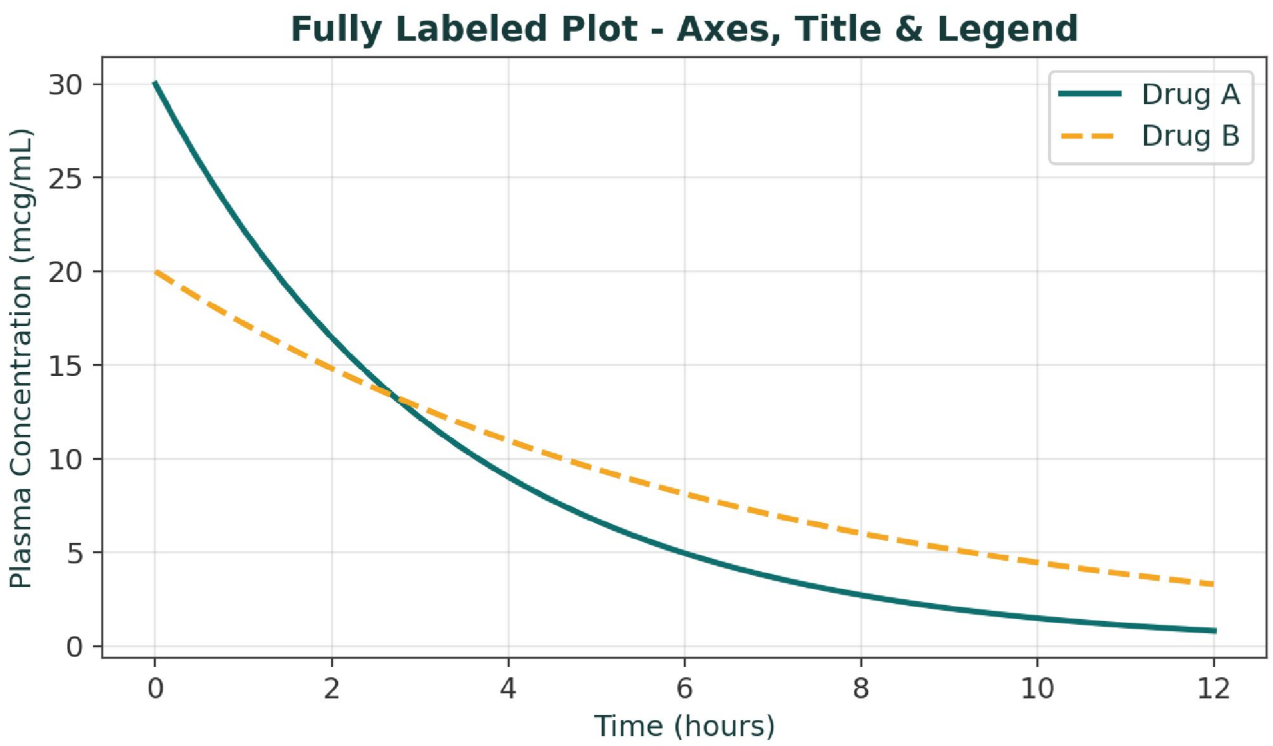
Box Plot Part	Meaning
`Box (top/bottom)`	75th percentile (Q3) and 25th percentile (Q1)
`Line inside box`	Median (50th percentile)
`Whiskers`	Range of typical (non-outlier) values
`Dots beyond whiskers`	Outliers - unusually high/low values

Labeling Axes, Titles & Legends

Every scientific plot should be fully labeled — an unlabeled graph is not acceptable in a report or publication. A legend is essential whenever more than one dataset is shown on the same plot.

```
plt.plot(time, drugA_conc, color='teal', label='Drug A')
plt.plot(time, drugB_conc, color='orange', linestyle='--', label='Drug B')

plt.xlabel('Time (hours)')
plt.ylabel('Plasma Concentration (mcg/mL)')
plt.title('Comparing Two Drugs')
plt.legend(loc='upper right')     # shows the labels defined above
plt.grid(alpha=0.3)               # optional light gridlines
plt.show()
```



Function	Purpose
`plt.xlabel() / plt.ylabel()`	Names the x-axis / y-axis
`plt.title()`	Adds a title above the plot
`label='...' (inside plot())`	Sets the name shown in the legend
`plt.legend()`	Displays the legend box; loc sets its position
`plt.grid()`	Adds light gridlines for easier reading

Concentration-Time Curves — Oral vs IV

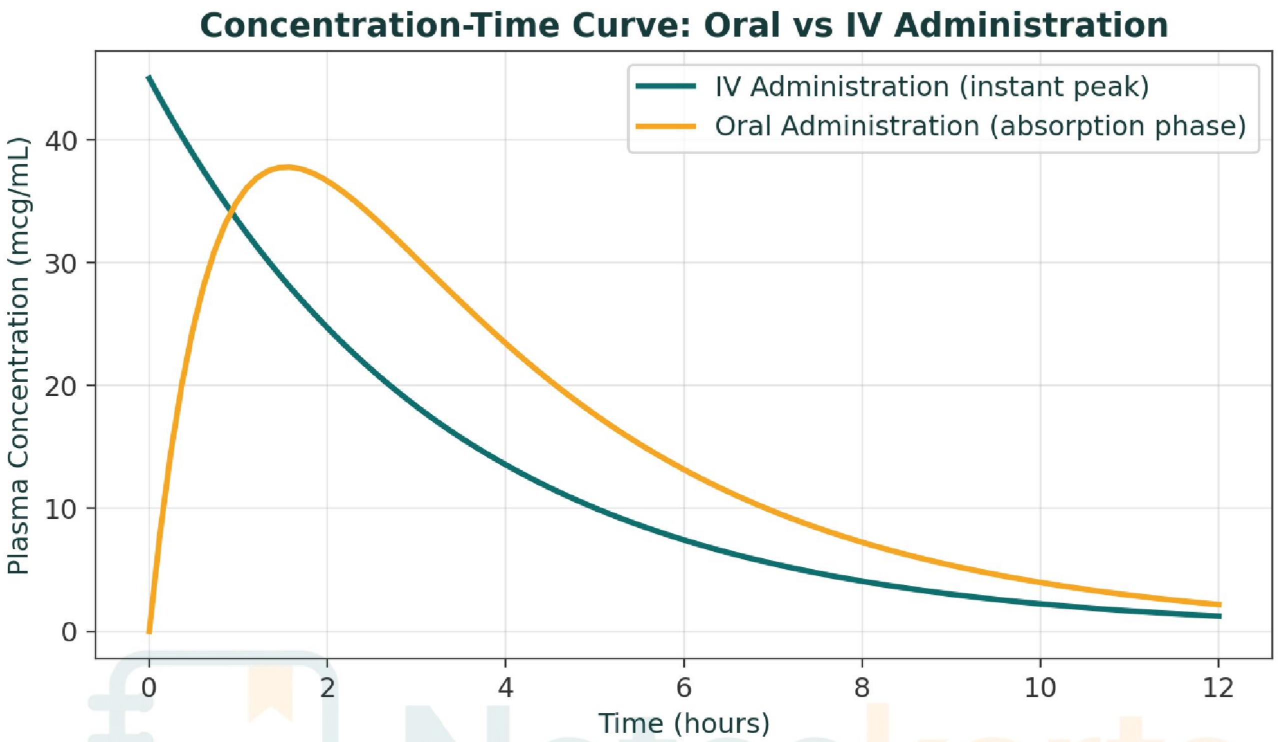
A concentration-time curve plots plasma drug concentration against time and is central to pharmacokinetics. IV administration shows an immediate peak (no absorption phase), while oral administration shows a rising absorption phase before reaching Cmax at Tmax.

```
import numpy as np
t = np.linspace(0, 12, 100)
ka, ke = 1.2, 0.3

iv_conc = 45 * np.exp(-ke * t)                              # IV: pure
elimination
oral_conc = (ka/(ka-ke)) * (np.exp(-ke*t) - np.exp(-ka*t)) * 60  # Oral:
absorption + elimination

plt.plot(t, iv_conc, color='teal', label='IV Administration')
```

```
plt.plot(t, oral_conc, color='orange', label='Oral Administration')
plt.xlabel('Time (hours)'); plt.ylabel('Plasma Concentration (mcg/mL)')
plt.title('Concentration-Time Curve: Oral vs IV'); plt.legend()
plt.show()
```



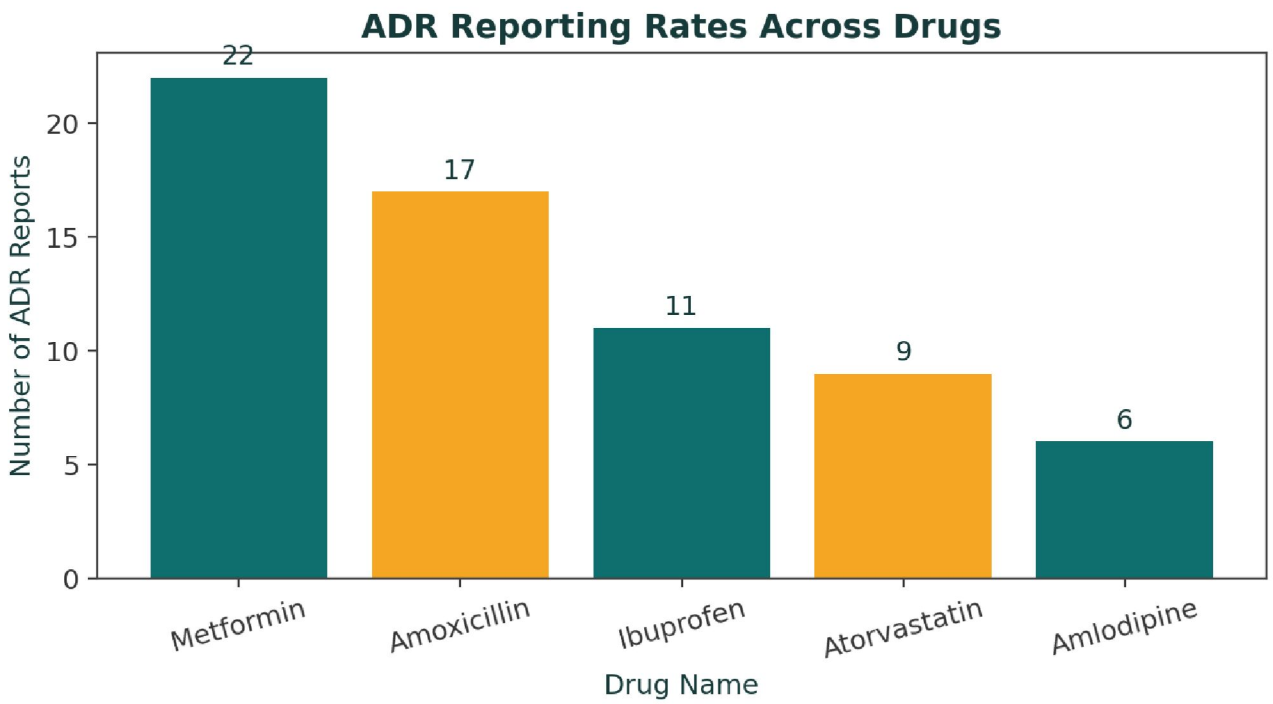
□ **Reading This Curve**
IV curve starts at its highest point and only declines (pure elimination phase). Oral curve rises during absorption, peaks at Tmax (Cmax), then declines during elimination - the area under either curve (AUC) reflects total drug exposure.

ADR Reporting Rates Across Drugs

A bar chart is the standard way to compare a count or rate across categories - here, the number of Adverse Drug Reaction (ADR) reports received for each drug.

```
drugs = ['Metformin','Amoxicillin','Ibuprofen','Atorvastatin','Amlodipine']
adr_counts = [22, 17, 11, 9, 6]

plt.bar(drugs, adr_counts, color=['teal','orange','teal','orange','teal'])
plt.xlabel('Drug Name'); plt.ylabel('Number of ADR Reports')
plt.title('ADR Reporting Rates Across Drugs')
plt.xticks(rotation=15)
plt.show()
```



Interpretation

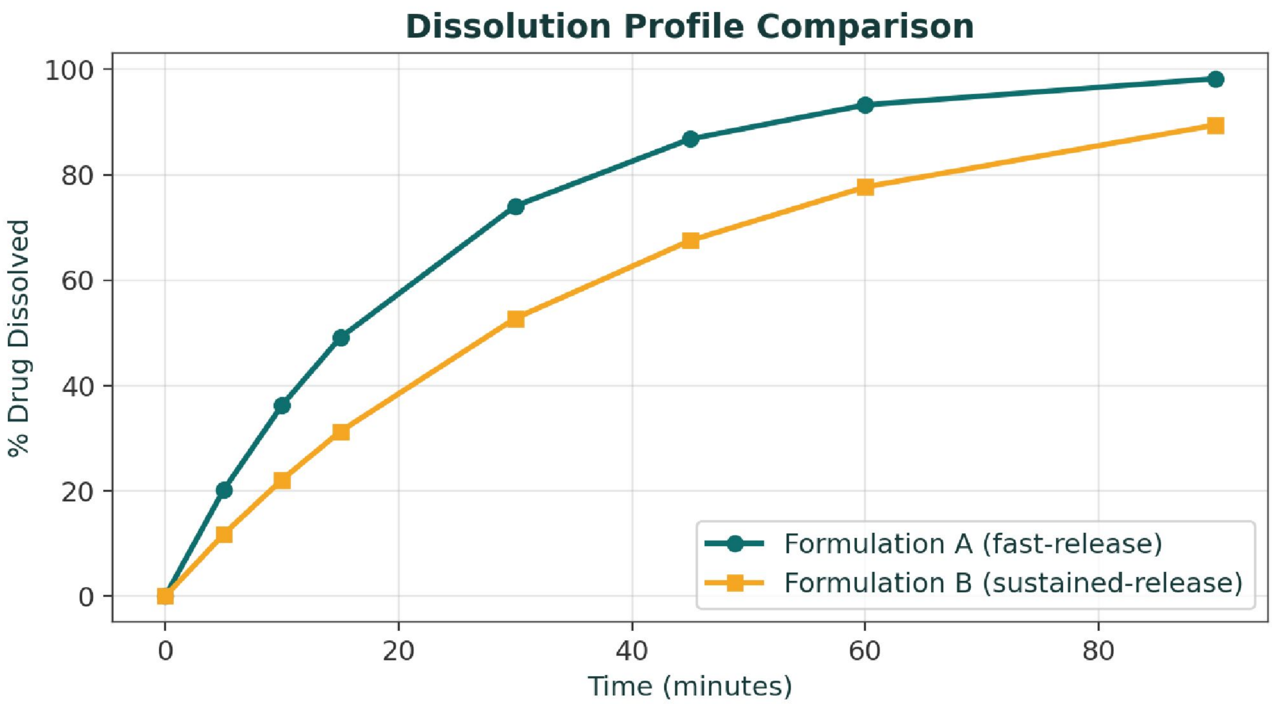
Metformin shows the highest ADR count here - but a bar chart alone doesn't show WHY: it could reflect higher prescription volume rather than a more dangerous drug. Always consider the total number of patients on each drug before concluding relative risk.

Dissolution Profiles

A dissolution profile plots the percentage of drug dissolved against time, used to compare how quickly different tablet formulations release their active ingredient - critical for quality control and bioequivalence studies.

```
time_min = [0, 5, 10, 15, 30, 45, 60, 90]
formA = [0, 22, 38, 52, 74, 86, 93, 98]     # fast-release
formB = [0, 10, 18, 26, 45, 58, 68, 80]     # sustained-release

plt.plot(time_min, formA, color='teal', marker='o', label='Formulation A')
plt.plot(time_min, formB, color='orange', marker='s', label='Formulation B')
plt.xlabel('Time (minutes)'); plt.ylabel('% Drug Dissolved')
plt.title('Dissolution Profile Comparison'); plt.legend(loc='lower right')
plt.show()
```



Scientific Interpretation of Plots

Creating a plot is only half the task - correctly reading and interpreting it is what makes it useful in pharmacy practice and research.

Plot Feature	What to Look For
`Peak (Cmax) height/time`	Higher/earlier peak may mean faster or more complete absorption
`Curve steepness (decline)`	Steeper decline = faster elimination (higher clearance)
`Overlapping curves`	Similar shape/AUC may suggest bioequivalence between formulations
`Box plot spread (IQR)`	Wider box = more batch-to-batch variability - a quality concern
`Outlier points`	May indicate an assay error OR a genuine unusual patient/batch response
`Scatter plot trend`	A clear upward pattern suggests dose-response correlation

Correlation is Not Causation

A scatter plot showing dose and response rising together does not, by itself, prove the dose CAUSES the

Plot Feature	What to Look For
response - confounding factors (age, diet, other medications) must be considered before drawing conclusions.	

Stay Connected with Noteskarts

Join our community for daily updates, free notes, and exam alerts

WhatsApp Channel | **Telegram Group** | **YouTube Channel**

Thank You for Studying with Noteskarts!

www.noteskarts.com

Smart Notes. Bright Future.