

FIGURE & AXES MODEL

```
import matplotlib.pyplot as plt
import seaborn as sns

fig, ax = plt.subplots(figsize=(8, 4))
ax.plot(x, y) # draw on the axes
ax.set_title("Title")
fig.tight_layout()
plt.show()
```

Figure = the canvas; **Axes** = one chart on it. Prefer the object API (`ax. ....`) over `plt. ....`.

CORE CHART TYPES

<code>ax.plot(x, y, marker="o", ls="--")</code>	line
<code>ax.bar(cats, vals)</code>	vertical bars
<code>ax.barh(cats, vals)</code>	horizontal bars (long labels)
<code>ax.bar(x, a); ax.bar(x, b, bottom=a)</code>	stacked bars
<code>ax.scatter(x, y, s=20, c=z, alpha=.6)</code>	points; size s, colour c
<code>ax.hist(v, bins=20, edgecolor="w")</code>	distribution
<code>ax.boxplot([a, b])</code>	median, quartiles, outliers
<code>ax.pie(v, labels=l, autopct="%1.0f%%")</code>	parts of a whole (few slices)
<code>ax.axhline(0) / ax.axvline(x0)</code>	reference lines
<code>ax.fill_between(x, lo, hi, alpha=.2)</code>	shaded band

COMMON MISTAKES

- Bars that do not start at zero exaggerate differences.
- Too many pie slices or colours; use a sorted bar chart.
- Missing title, axis labels or units.
- Overplotted scatter: add `alpha` or use `hexbin`.
- Mixing `plt.` and `ax.` calls across several axes.
- Forgetting `fig.tight_layout()` so labels clip; or `bbox_inches="tight"` on save.
- Rainbow colour maps; use `viridis` or a sequential palette.
- Plotting unsorted dates: sort by the date first.

LABELS, LEGEND, TICKS

<code>ax.set(title="T", xlabel="X", ylabel="Y")</code>	set several at once
<code>ax.plot(x, y, label="Sales")</code> <code>ax.legend(loc="upper left", frameon=False)</code>	label lines, then show legend
<code>ax.set_xlim(0, 100); ax.set_ylim(bottom=0)</code>	axis limits
<code>ax.tick_params(axis="x", rotation=45)</code>	rotate tick labels
<code>ax.set_xticks(ticks)</code> <code>ax.set_xticklabels(names)</code>	custom ticks (set ticks first)
<code>ax.xaxis.set_major_formatter("x:,.0f")</code>	1,000 separators
<code>ax.grid(alpha=.3)</code>	light gridlines
<code>ax.annotate("Peak", xy=(x0, y0), xytext=(x1, y1), arrowprops=dict(arrowstyle=">"))</code>	note with arrow
<code>ax.text(x, y, "label", ha="center")</code>	free text
<code>ax.bar_label(bars, fmt="%0.0f")</code>	value labels on bars
<code>ax.spines[["top", "right"]].set_visible(False)</code>	cleaner frame

STYLE, COLOUR, SIZE

<code>plt.style.use("seaborn-v0_8-whitegrid")</code>	global style (see <code>plt.style.available</code>)
<code>color="#1f77b4", "tab:orange", "C2"</code>	hex / named / cycle colour
<code>lw=2, ls=":", alpha=.5</code>	line width / style / transparency
<code>cmap="viridis"</code>	colour map (also "Blues", "RdBu")
<code>plt.rcParams["font.size"] = 11</code>	global defaults
<code>figsize=(w, h)</code>	inches, e.g. (10, 5)

SUBPLOTS GRID & SAVING

```
fig, axes = plt.subplots(2, 2, figsize=(10, 7), sharex=True)

axes[0, 0].plot(x, y)
axes[0, 1].hist(v)
for ax in axes.flat:
    ax.grid(alpha=.3)
fig.suptitle("Overview")
fig.tight_layout()
fig.savefig("chart.png", dpi=300,
           bbox_inches="tight")

plt.subplots(1, 3)
layout="constrained"

transparent=True

Call savefig before plt.show().
```

PANDAS .PLOT

<code>df.plot(x="date", y="sales")</code>	line from DataFrame
<code>df.plot(kind="bar", x="cat", y="v")</code>	also "barh", "hist", "box", "scatter", "pie", "area"
<code>df.plot.scatter(x="a", y="b", c="c")</code>	method form
<code>df["v"].plot.hist(bins=30)</code>	from a Series
<code>ax = df.plot(...); ax.set_ylabel("f")</code>	returns Axes: keep customising
<code>df.plot(ax=ax)</code>	draw into an existing axes
<code>df.groupby("g")["v"].mean().plot.bar()</code>	aggregate, then plot
<code>stacked=True, subplots=True</code>	stack / one panel per column

SEABORN SETUP

<code>sns.set_theme(style="whitegrid", palette="deep", context="talk")</code>	one-line global look
<code>sns.load_dataset("tips")</code>	built-in practice data
<code>sns.color_palette("Set2")</code>	categorical palette
<code>ax=ax</code>	draw into your own Matplotlib axes
<code>sns.despine()</code>	remove top/right spines

SEABORN PLOTS

<code>sns.histplot(df, x="v", hue="g", kde=True)</code>	histogram + density
<code>sns.kdeplot(df, x="v", hue="g", fill=True)</code>	smooth density
<code>sns.countplot(df, x="cat", order=...)</code>	count of rows per category
<code>sns.barplot(df, x="cat", y="v", estimator="mean", errorbar="ci")</code>	mean per category with interval
<code>sns.boxplot(df, x="cat", y="v")</code>	spread by category
<code>sns.violinplot(df, x="cat", y="v")</code>	full distribution shape
<code>sns.scatterplot(df, x="a", y="b", hue="g", size="s")</code>	relationship, coloured by group
<code>sns.lineplot(df, x="date", y="v", hue="g")</code>	trend; mean + CI if repeated x
<code>sns.regplot(df, x="a", y="b")</code>	scatter + fitted line
<code>sns.heatmap(df, corr(numeric_only=True), annot=True, fmt=".2f", cmap="coolwarm", center=0)</code>	correlation matrix
<code>sns.pairplot(df, hue="g")</code>	grid of all pairs
<code>sns.catplot(df, x="c", y="v", col="g", kind="box")</code>	facets by column variable

WHICH CHART FOR WHICH QUESTION?

Trend over time	line
Compare categories	bar / barh (sorted)
Distribution of one number	histogram, KDE, box
Compare distributions	box / violin by group
Relationship of two numbers	scatter (+ regplot)
Many numeric relationships	corr heatmap, pairplot
Share of a whole	stacked bar; pie only if 2-4 slices
Two categories x a value	heatmap of pivot_table