Matplotlib

A tour of Matplotlib, from a first Figure and Axes to fully styled, multi-panel figures.

Figure & Axes

The Figure is the canvas; each Axes is a plot you draw on.

PURPOSE	COMMAND	RESULT
One figure, one axes.	`fig, ax = plt.subplots()`	
A grid of axes.	`fig, axs = plt.subplots(2, 3)`	
Set the size (inches).	`plt.subplots(figsize=(8, 5))`	
Draw on an axes.	`ax.plot([1, 2, 3], [1, 4, 9])`	
pyplot vs the object API.	`plt.plot(...) # implicit current axes` `ax.plot(...) # explicit, recommended`	
Show or close.	`plt.show() plt.close(fig)`	

plt.subplots() returns a Figure and one or more Axes; the explicit ax.* object API scales to grids and reuse

Core Plot Types

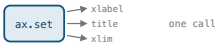
One Axes method per chart kind.

PURPOSE	COMMAND	RESULT
Line plot.	`ax.plot(x, y)`	
Scatter (color + size).	`ax.scatter(x, y, c=col, s=size)`	
Bars (vertical / horizontal).	`ax.bar(cats, vals) ax.barh(...)`	
Histogram of one variable.	`ax.hist(data, bins=20)`	
Image / heatmap of a matrix.	`ax.imshow(M, cmap="viridis")`	
Band / error bars.	`ax.fill_between(x, lo, hi) ax.errorbar`	

every chart is an Axes method; plot/scatter/bar/hist/imshow cover most needs, fill_between/errorbar add uncertainty

Labels, Title & Legend

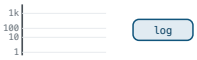
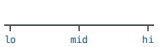
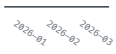
Name the axes, title the plot, key the series.

PURPOSE	COMMAND	RESULT
Label the axes.	`ax.set_xlabel("t"); ax.set_ylabel("v")`	
Give it a title.	`ax.set_title("Sales")`	
Key the series with a legend.	`ax.plot(..., label="A");ax.legend()`	
Set several at once.	`ax.set(xlabel=, ylabel=, title=, xlim=)`	 one call
Annotate a point.	`ax.annotate("peak", xy=(x,y), xytext=...)`	
Text and gridlines.	`ax.text(x, y, "hi");ax.grid(True)`	grid + text

label="A" plus ax.legend() builds the key automatically; ax.set(...) sets many properties in one call

Scales, Limits & Ticks

Control the range, the scale, and the tick labels.

PURPOSE	COMMAND	RESULT
Set axis limits.	`ax.set_xlim(0, 10); ax.set_ylim(...)`	 zoom range
Log-scale an axis.	`ax.set_yscale("log")`	
Custom tick positions / labels.	`ax.set_xticks([0,5,10], ["lo","mid","hi"])`	
Format the tick labels.	`ax.yaxis.set_major_formatter( PercentFormatter())`	 % / $ / dates
Rotate crowded date labels.	`fig.autofmt_xdate()`	 no overlap
Invert or share an axis.	`ax.invert_yaxis(); sharex=True`	 flip / sync axes

set_xlim/ylim bound the view; set_yscale("log") transforms it; locators and formatters control tick marks and labels

Multi-Plot Layout

Arrange several Axes on one Figure.

PURPOSE	COMMAND	RESULT
Pack subplots without overlap.	`plt.subplots(layout="constrained")`	 auto-spaced
Span cells with GridSpec.	`gs = fig.add_gridspec(2,2);` `fig.add_subplot(gs[0,:])`	 top spans 2 cols
Name panels with a mosaic.	`plt.subplot_mosaic("AB;CC")`	
Two y-axes on one plot.	`ax2 =ax.twinx()`	 shared x, two y
Stack panels sharing an axis.	`plt.subplots(2, 1, sharex=True)`	one x-axis
Title the whole figure.	`fig.suptitle("Report")`	 figure-level title

layout="constrained" auto-spaces panels; GridSpec and subplot_mosaic build uneven grids; twinx shares one x

Style & Color

Theme the whole plot or color one line.

PURPOSE	COMMAND	RESULT
Apply a style sheet.	`plt.style.use("ggplot")`	 restyled
Set a global default.	`plt.rcParams["figure.dpi"] = 150`	 every figure inherits it
Automatic color cycle.	`ax.plot(...) # auto C0, C1, C2 ...`	
Style one line.	`ax.plot(..., color="C1", lw=2, ls="--")`	 dashed orange
Color by value (colormap).	`ax.scatter(x, y, c=z, cmap="viridis")`	
Temporary style block.	`withplt.style.context("dark_background"):`	 scoped style

style.use sets the theme, rcParams sets global defaults; per-call color/lw/ls/cmap override for one artist

Reach Into the Artists

Every element is an Artist object you can capture and mutate.

PURPOSE	COMMAND	RESULT
Capture and mutate a handle.	`line, = ax.plot(...); line.set_color("red")`	
Add shapes (patches).	`ax.add_patch(Rectangle(...)); Circle(...)`	
Hide the top/right spines.	`ax.spines[["top","right"]].set_visible(False)`	
Attach a colorbar.	`fig.colorbar(im, ax=ax)`	
Everything is an Artist.	`Figure > Axes > Line2D / Text / Patch`	
Introspect and bulk-set.	`ax.get_lines(); plt.setp(line, lw=3)`	

plot returns the artists it drew; capture them to set_color/set_linewidth/... after the fact, or plt.setp many at once

Save & Export

Write the figure to disk at the size, format, and dpi you want.

PURPOSE	COMMAND	RESULT
Save a raster image.	`fig.savefig("plot.png", dpi=300)`	
Save a vector format.	`fig.savefig("plot.svg") .pdf`	
Trim the margins.	`fig.savefig(..., bbox_inches="tight")`	
Transparent background.	`fig.savefig(..., transparent=True)`	
Render headless (no display).	`matplotlib.use("Agg")`	
Close figures to free memory.	`plt.close("all")`	

savefig picks the format from the extension; bbox_inches="tight" trims margins; close figures in loops to free memory