# Matplotlib Python plotting

## CONCEPTS

Prefer the object style: fig and ax methods. plt.plot() works on the implicit current axes and gets confusing with several plots.

**Figure** the canvas, saved as one file

- **Axes** one plot: data, limits, labels
  - **Axis / Artists** ticks, lines, text, patches

## FIGURE AND AXES

`fig, ax = plt.subplots()`
One figure, one axes

`fig, axs = plt.subplots(<rows>, <cols>)`
Grid of axes (2-D array)

`plt.subplots(figsize=(<w>, <h>))`
Size in inches

`plt.subplots(..., sharex=True, sharey=True)`
Share axis ranges

`plt.subplots(layout="constrained")`
Automatic non-overlapping layout

`fig, axd = plt.subplot_mosaic("AB;CC")`
Named axes layout; axd["A"]

## FIGURE AND AXES (cont.)

`fig.add_subplot(<rows>, <cols>, <n>)`
Add axes to an existing figure

`ax.twinx()`
Second y axis on the right

`plt.close("all")`
Free all open figures

## PLOT TYPES

`ax.plot(<x>, <y>, "o-", label="<name>")`
Line with markers

`ax.scatter(<x>, <y>, c=<vals>, s=<size>)`
Scatter, colour/size by data

`ax.bar(<x>, <height>)`
Vertical bars

`ax.barh(<y>, <width>)`
Horizontal bars

`ax.hist(<data>, bins=<n>)`
Histogram

`ax.imshow(<2d-array>, cmap="viridis")`
Show an array as an image

`ax.boxplot(<data>)`
Box-and-whisker plot

`ax.errorbar(<x>, <y>, yerr=<err>)`
Points with error bars

## PLOT TYPES (cont.)

`ax.fill_between(<x>, <lo>, <hi>, alpha=0.3)`
Shaded band

`ax.pie(<sizes>, labels=<names>)`
Pie chart

`ax.violinplot(<data>)`
Distribution shape per group

## LABELS AND TEXT

`ax.set_title("<text>")`
Axes title

`ax.set_xlabel("<text>")`
X axis label (ylabel likewise)

`ax.set(title="<t>", xlabel="<x>", ylabel="<y>")`
Set several properties at once

`fig.suptitle("<text>")`
Title over the whole figure

`ax.text(<x>, <y>, "<text>")`
Text at data coordinates

`ax.annotate("<text>", xy=(<x>, <y>), xytext=(<x>, <y>), arrowprops=dict(arrowstyle="→"))`
Label a point with an arrow

`ax.bar_label(<bars>)`
Value labels on bars

`ax.axhline(<y>) / ax.axvline(<x>)`
Reference line

## LEGEND

`ax.legend()`
Legend from label= arguments

`ax.legend(loc="upper left")`
Choose the corner

`ax.legend(bbox_to_anchor=(1, 1), loc="upper left")`
Place outside the axes

`ax.legend(ncols=<n>, frameon=False)`
Columns, no box

## LIMITS, SCALES, TICKS

`ax.set_xlim(<lo>, <hi>)`
X range (ylim likewise)

`ax.set_yscale("log")`
Log axis (xscale likewise)

`ax.invert_yaxis()`
Flip the y direction

`ax.set_xticks(<positions>, labels=<names>)`
Custom ticks and labels

`ax.tick_params(axis="x", rotation=45)`
Rotate or style tick labels

`ax.grid(True, alpha=0.3)`
Grid lines

`ax.set_aspect("equal")`
Equal unit lengths

## STYLE AND COLOUR

`ax.plot(..., color="<c>", linestyle="--", linewidth=<w>)`
Line style

`ax.scatter(..., alpha=0.5)`
Transparency

`plt.colorbar(<mappable>, ax=ax)`
Colour scale for scatter/ imshow

`cmap="viridis"`
Colormap name (also plasma, gray, coolwarm)

`plt.rcParams["<key>"] = <val>`
Change a global default

## SAVE AND SHOW

`fig.savefig("<file>.png", dpi=150)`
Save as raster

`fig.savefig("<file>.pdf")`
Vector output (also svg)

`fig.savefig(..., bbox_inches="tight")`
Trim whitespace

`plt.show()`
Display figures, blocks until closed