# 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 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

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

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

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

## 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

## LEGEND

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

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

## 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

## 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