

Quick start

Y = np.cos(X)

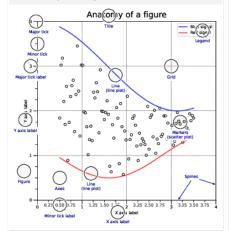
import numpy as np import matplotlib as mpl import matplotlib.pyplot as plt

X = np.linspace(0, 2*np.pi, 100)

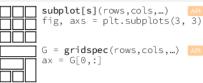
fig, ax = plt.subplots() ax.plot(X, Y, color='green')

fig.savefig("figure.pdf") fig.show()

Anatomy of a figure



Subplots layout



ax.inset axes(extent)

d=make_axes_locatable(ax) All ax = d.new_horizontal('10%')

Getting help

matplotlib.org

github.com/matplotlib/matplotlib/issues O discourse.matplotlib.org

stackoverflow.com/questions/tagged/matplotlib

III gitter.im/matplotlib

Basic plots

API

plot([X],Y,[fmt],...) X, Y, fmt, color, marker, linestyle

scatter(X,Y,...) X, Y, [s]izes, [c]olors, marker, cmap

bar[h](x,height,...) x, height, width, bottom, align, color

imshow(Z,...) Z, cmap, interpolation, extent, origin

contour[f]([X],[Y],Z,...) X, Y, Z, levels, colors, extent, origin

pcolormesh([X],[Y],Z,...)X, Y, Z, vmin, vmax, cmap

quiver([X],[Y],U,V,...) X, Y, U, V, C, units, angles

pie(X,...) Z, explode, labels, colors, radius

text(x,y,text,...) x, y, text, va, ha, size, weight, transform

fill[between][x](...) X, Y1, Y2, color, where

Advanced plots

API

step(X,Y,[fmt],...) X, Y, fmt, color, marker, where

boxplot(X,...) X, notch, sym, bootstrap, widths

errorbar(X,Y,xerr,yerr,...) [A] X, Y, xerr, yerr, fmt

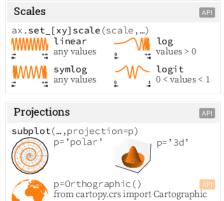
hist(X, bins, ...) X, bins, range, density, weights

violinplot(D,...) D, positions, widths, vert

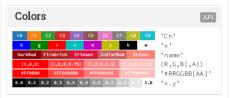
barbs([X],[Y], U, V, ...) X, Y, U, V, C, length, pivot, sizes

eventplot(positions,...) positions, orientation, lineoffsets

hexbin(X,Y,C,...) X, Y, C, gridsize, bins



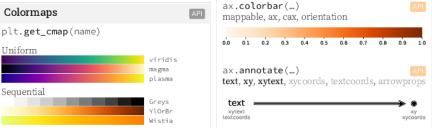




Diverging

Oualitative

Cyclic



Spectral

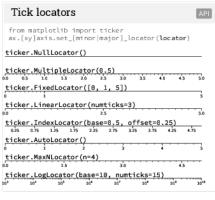
coolwarm

RdGv

twilight

Event handling

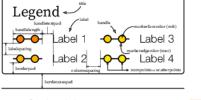
fig, ax = plt.subplots() def on_click(event): print(event) fig.canvas.mpl_connect('button_press_event', on_click)

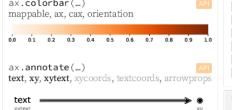


Tick formatters API from matplotlib import ticker ax.[xy]axis.set_[minor|major]_formatter(formatter) ticker.NullFormatter() ticker.FixedFormatter(['zero', 'one', 'two', ...]) ticker.FuncFormatter(lambda x, pos: "[%.2f]" % x) ticker.FormatStrFormatter('>%d<') ticker.ScalarFormatter() ticker.StrMethodFormatter('{x}') ticker.PercentFormatter(xmax=5)

Ornaments

ax.legend(...) handles, labels, loc, title, frameon





Ten simple rules

4. Captions Are Not Optional

5. Do Not Trust the Defaults

7. Do Not Mislead the Reader

9. Message Trumps Beauty

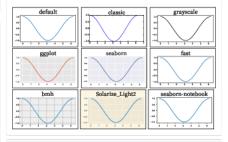
import matplotlib.animation as mpla T = np.linspace(0, 2*np.pi, 100)S = np.sin(T)line, = plt.plot(T, S)

def animate(i): line.set_ydata(np.sin(T+i/50)) anim = mpla.FuncAnimation(plt.gcf(), animate, interval=5) plt.show()

Styles

Animation

plt.style.use(style)



Quick reminder

ax.grid() ax.set_[xy]lim(vmin, vmax) ax.set_[xy]label(label) ax.set_[xy]ticks(ticks, [labels]) ax.set [xy]ticklabels(labels) ax.set_title(title)

ax.tick_params(width=10, ...) ax.set_axis_[on|off]()

fig.suptitle(title) fig.tight_layout() plt.gcf(), plt.gca() mpl.rc('axes', linewidth=1, ...) [fig|ax].patch.set_alpha(0) text=r'\$\frac{-e^{i\pi}}{2^n}\$'

Keyboard shortcuts

ctrl + s Save ctrl + W Close plot r Reset view f Fullscreen 0/1 f View forward b View back

Pan view O Zoom to rect

X X pan/zoom y Y pan/zoom

g Minor grid 0/1

G Major grid 0/1 X axis log/linear L Y axis log/linear

1. Know Your Audience

2. Identify Your Message

3. Adapt the Figure

6. Use Color Effectively

8. Avoid "Chartiunk"

10. Get the Right Tool