Below we describe several common approaches to plotting with Matplotlib.

Contents

The Pyplot API

The `matplotlib.pyplot` module contains functions that allow you to generate many kinds of plots quickly. For examples that showcase the use of the `matplotlib.pyplot` module, see the Pyplot tutorial or the Pyplot Examples. We also recommend that you look into the object-oriented approach to plotting, described below.

`matplotlib.pyplot.plotting()`

Function	Description
`acorr`	Plot the autocorrelation of `x`.
`angle_spectrum`	Plot the angle spectrum.
`annotate`	Annotate the point `xy` with text `s`.
`arrow`	Add an arrow to the axes.
`autoscale`	Autoscale the axis view to the data (toggle).
`axes`	Add an axes to the figure.
`axhline`	Add a horizontal line across the axis.
`axhspan`	Add a horizontal span (rectangle) across the axis.
`axis`	Convenience method to get or set axis properties.
`axvline`	Add a vertical line across the axes.
`axvspan`	Add a vertical span (rectangle) across the axes.

Related Topics

Documentation overview
- The Matplotlib API
  - Previous: The Matplotlib API
  - Next: API Changes

This Page

Show Source

Quick search

Go

Function	Description
bar	Make a bar plot.
barbs	Plot a 2-D field of barbs.
barh	Make a horizontal bar plot.
box	Turn the axes box on or off.
boxplot	Make a box and whisker plot.
broken_barh	Plot horizontal bars.
cla	Clear the current axes.
clabel	Label a contour plot.
clf	Clear the current figure.
clim	Set the color limits of the current image.
close	Close a figure window.
cohere	Plot the coherence between *x* and *y*.
colorbar	Add a colorbar to a plot.
contour	Plot contours.
contourf	Plot contours.
csd	Plot the cross-spectral density.
delaxes	Remove an axes from the current figure.
draw	Redraw the current figure.
errorbar	Plot an errorbar graph.
eventplot	Plot identical parallel lines at specific positions.
figimage	Adds a non-resampled image to the figure.
figlegend	Place a legend in the figure.
fignum_exists	
figtext	Add text to figure.
figure	Creates a new figure.
fill	Plot filled polygons.
fill_between	Make filled polygons between two curves.
fill_betweenx	Make filled polygons between two horizontal curves.
findobj	Find artist objects.
gca	Get the current Axes instance on the current figure matching the given keyword args, or create one.
gcf	Get a reference to the current figure.

Function	Description
gci	Get the current colorable artist.
get_figlabels	Return a list of existing figure labels.
get_fignums	Return a list of existing figure numbers.
grid	Turn the axes grids on or off.
hexbin	Make a hexagonal binning plot.
hist	Plot a histogram.
hist2d	Make a 2D histogram plot.
hlines	Plot horizontal lines at each `y` from `xmin` to `xmax`.
hold	
imread	Read an image from a file into an array.
imsave	Save an array as in image file.
imshow	Display an image on the axes.
install_repl_displayhook	Install a repl display hook so that any stale figure are automatically redrawn when control is returned to the repl.
ioff	Turn interactive mode off.
ion	Turn interactive mode on.
ishold	
isinteractive	Return status of interactive mode.
legend	Places a legend on the axes.
locator_params	Control behavior of tick locators.
loglog	Make a plot with log scaling on both the *x* and *y* axis.
magnitude_spectrum	Plot the magnitude spectrum.
margins	Set or retrieve autoscaling margins.
matshow	Display an array as a matrix in a new figure window.
minorticks_off	Remove minor ticks from the current plot.
minorticks_on	Display minor ticks on the current plot.
over	
pause	Pause for *interval* seconds.
pcolor	Create a pseudocolor plot of a 2-D array.

Function	Description
pcolormesh	Plot a quadrilateral mesh.
phase_spectrum	Plot the phase spectrum.
pie	Plot a pie chart.
plot	Plot lines and/or markers to the Axes.
plot_date	A plot with data that contains dates.
plotfile	Plot the data in in a file.
polar	Make a polar plot.
psd	Plot the power spectral density.
quiver	Plot a 2-D field of arrows.
quiverkey	Add a key to a quiver plot.
rc	Set the current rc params.
rc_context	Return a context manager for managing rc settings.
rcdefaults	Restore the rc params from Matplotlib's internal defaults.
rgrids	Get or set the radial gridlines on a polar plot.
savefig	Save the current figure.
sca	Set the current Axes instance to *ax*.
scatter	Make a scatter plot of x vs y.
sci	Set the current image.
semilogx	Make a plot with log scaling on the *x* axis.
semilogy	Make a plot with log scaling on the y axis.
set_cmap	Set the default colormap.
setp	Set a property on an artist object.
show	Display a figure.
specgram	Plot a spectrogram.
spy	Plot the sparsity pattern on a 2-D array.
stackplot	Draws a stacked area plot.
stem	Create a stem plot.
step	Make a step plot.
streamplot	Draws streamlines of a vector flow.
subplot	Return a subplot axes positioned by the given grid definition.

Function	Description
subplot2grid	Create a subplot in a grid.
subplot_tool	Launch a subplot tool window for a figure.
subplots	Create a figure and a set of subplots This utility wrapper makes it convenient to create common layouts of subplots, including the enclosing figure object, in a single call.
subplots_adjust	Tune the subplot layout.
suptitle	Add a centered title to the figure.
switch_backend	Switch the default backend.
table	Add a table to the current axes.
text	Add text to the axes.
thetagrids	Get or set the theta locations of the gridlines in a polar plot.
tick_params	Change the appearance of ticks and tick labels.
ticklabel_format	Change the ScalarFormatter used by default for linear axes.
tight_layout	Automatically adjust subplot parameters to give specified padding.
title	Set a title of the current axes.
tricontour	Draw contours on an unstructured triangular grid.
tricontourf	Draw contours on an unstructured triangular grid.
tripcolor	Create a pseudocolor plot of an unstructured triangular grid.
triplot	Draw a unstructured triangular grid as lines and/or markers.
twinx	Make a second axes that shares the *x*-axis.
twiny	Make a second axes that shares the *y*-axis.
uninstall_repl_displayhook	Uninstalls the matplotlib display hook.
violinplot	Make a violin plot.
vlines	Plot vertical lines.
xcorr	Plot the cross correlation between *x* and *y*.

Function	Description
xkcd	Turns on xkcd sketch-style drawing mode.
xlabel	Set the *x* axis label of the current axis.
xlim	Get or set the *x* limits of the current axes.
xscale	Set the scaling of the *x*-axis.
xticks	Get or set the *x*-limits of the current tick locations and labels.
ylabel	Set the *y* axis label of the current axis.
ylim	Get or set the *y*-limits of the current axes.
yscale	Set the scaling of the *y*-axis.
yticks	Get or set the *y*-limits of the current tick locations and labels.

The Object-Oriented API

Most of these functions also exist as methods in the `matplotlib.axes.Axes` class. You can use them with the “Object Oriented” approach to Matplotlib.

While it is easy to quickly generate plots with the `matplotlib.pyplot` module, we recommend using the object-oriented approach for more control and customization of your plots. See the methods in the `matplotlib.axes.Axes()` class for many of the same plotting functions. For examples of the OO approach to Matplotlib, see the API Examples.

Colors in Matplotlib

There are many colormaps you can use to map data onto color values. Below we list several ways in which color can be utilized in Matplotlib.

For a more in-depth look at colormaps, see the Colormaps in Matplotlib tutorial.

`matplotlib.pyplot.colormaps()`

> Matplotlib provides a number of colormaps, and others can be added using `register_cmap()`. This function documents the built-in colormaps, and will also return a list of all registered colormaps if called.
>
> You can set the colormap for an image, pcolor, scatter, etc, using a keyword argument:

```
imshow(X, cmap=cm.hot)
```

or using the `set_cmap()` function:

```
imshow(X)
pyplot.set_cmap('hot')
pyplot.set_cmap('jet')
```

In interactive mode, `set_cmap()` will update the colormap post-hoc, allowing you to see which one works best for your data.

All built-in colormaps can be reversed by appending `_r`: For instance, `gray_r` is the reverse of `gray`.

There are several common color schemes used in visualization:

Sequential schemes

> for unipolar data that progresses from low to high

Diverging schemes

> for bipolar data that emphasizes positive or negative deviations from a central value

Cyclic schemes

> meant for plotting values that wrap around at the endpoints, such as phase angle, wind direction, or time of day

Qualitative schemes

> for nominal data that has no inherent ordering, where color is used only to distinguish categories

The base colormaps are derived from those of the same name provided with Matlab:

Colormap	Description
autumn	sequential linearly-increasing shades of red-orange-yellow
bone	sequential increasing black-white color map with a tinge of blue, to emulate X-ray film
cool	linearly-decreasing shades of cyan-magenta
copper	sequential increasing shades of black-copper
flag	repetitive red-white-blue-black pattern (not cyclic at endpoints)

Colormap	Description
gray	sequential linearly-increasing black-to-white grayscale
hot	sequential black-red-yellow-white, to emulate blackbody radiation from an object at increasing temperatures
hsv	cyclic red-yellow-green-cyan-blue-magenta-red, formed by changing the hue component in the HSV color space
inferno	perceptually uniform shades of black-red-yellow
jet	a spectral map with dark endpoints, blue-cyan-yellow-red; based on a fluid-jet simulation by NCSA [1]
magma	perceptually uniform shades of black-red-white
pink	sequential increasing pastel black-pink-white, meant for sepia tone colorization of photographs
plasma	perceptually uniform shades of blue-red-yellow
prism	repetitive red-yellow-green-blue-purple-…-green pattern (not cyclic at endpoints)
spring	linearly-increasing shades of magenta-yellow
summer	sequential linearly-increasing shades of green-yellow
viridis	perceptually uniform shades of blue-green-yellow
winter	linearly-increasing shades of blue-green

For the above list only, you can also set the colormap using the corresponding pylab shortcut interface function, similar to Matlab:

```
imshow(X)
hot()
jet()
```

The next set of palettes are from the Yorick scientific visualisation package, an evolution of the GIST package, both by David H. Munro:

Colormap	Description

gist_earth	mapmaker's colors from dark blue deep ocean to green lowlands to brown highlands to white mountains
gist_heat	sequential increasing black-red-orange-white, to emulate blackbody radiation from an iron bar as it grows hotter
gist_ncar	pseudo-spectral black-blue-green-yellow-red-purple-white colormap from National Center for Atmospheric Research [2]
gist_rainbow	runs through the colors in spectral order from red to violet at full saturation (like *hsv* but not cyclic)
gist_stern	"Stern special" color table from Interactive Data Language software

The following colormaps are based on the ColorBrewer color specifications and designs developed by Cynthia Brewer:

ColorBrewer Diverging (luminance is highest at the midpoint, and decreases towards differently-colored endpoints):

Colormap	Description
BrBG	brown, white, blue-green
PiYG	pink, white, yellow-green
PRGn	purple, white, green
PuOr	orange, white, purple
RdBu	red, white, blue
RdGy	red, white, gray
RdYlBu	red, yellow, blue
RdYlGn	red, yellow, green
Spectral	red, orange, yellow, green, blue

ColorBrewer Sequential (luminance decreases monotonically):

Colormap	Description
Blues	white to dark blue
BuGn	white, light blue, dark green
BuPu	white, light blue, dark purple
GnBu	white, light green, dark blue
Greens	white to dark green
Greys	white to black (not linear)
Oranges	white, orange, dark brown
OrRd	white, orange, dark red
PuBu	white, light purple, dark blue
PuBuGn	white, light purple, dark green

Colormap	Description
PuRd	white, light purple, dark red
Purples	white to dark purple
RdPu	white, pink, dark purple
Reds	white to dark red
YlGn	light yellow, dark green
YlGnBu	light yellow, light green, dark blue
YlOrBr	light yellow, orange, dark brown
YlOrRd	light yellow, orange, dark red

ColorBrewer Qualitative:

(For plotting nominal data, `ListedColormap` is used, not `LinearSegmentedColormap`. Different sets of colors are recommended for different numbers of categories.)

- Accent
- Dark2
- Paired
- Pastel1
- Pastel2
- Set1
- Set2
- Set3

Other miscellaneous schemes:

Colormap	Description
afmhot	sequential black-orange-yellow-white blackbody spectrum, commonly used in atomic force microscopy
brg	blue-red-green
bwr	diverging blue-white-red
coolwarm	diverging blue-gray-red, meant to avoid issues with 3D shading, color blindness, and ordering of colors [3]
CMRmap	“Default colormaps on color images often reproduce to confusing grayscale images. The proposed colormap maintains an aesthetically pleasing color image that automatically reproduces to a monotonic grayscale with discrete, quantifiable saturation levels.” [4]

Colormap	Description
cubehelix	Unlike most other color schemes cubehelix was designed by D.A. Green to be monotonically increasing in terms of perceived brightness. Also, when printed on a black and white postscript printer, the scheme results in a greyscale with monotonically increasing brightness. This color scheme is named cubehelix because the r,g,b values produced can be visualised as a squashed helix around the diagonal in the r,g,b color cube.
gnuplot	gnuplot's traditional pm3d scheme (black-blue-red-yellow)
gnuplot2	sequential color printable as gray (black-blue-violet-yellow-white)
ocean	green-blue-white
rainbow	spectral purple-blue-green-yellow-orange-red colormap with diverging luminance
seismic	diverging blue-white-red
nipy_spectral	black-purple-blue-green-yellow-red-white spectrum, originally from the Neuroimaging in Python project
terrain	mapmaker's colors, blue-green-yellow-brown-white, originally from IGOR Pro

The following colormaps are redundant and may be removed in future versions. It's recommended to use the names in the descriptions instead, which produce identical output:

Colormap	Description
gist_gray	identical to *gray*
gist_yarg	identical to *gray_r*
binary	identical to *gray_r*
spectral	identical to *nipy_spectral* [5]

Footnotes

[1] Rainbow colormaps, `jet` in particular, are considered a poor choice for scientific visualization by many researchers: Rainbow Color Map (Still) Considered Harmful

[2] Resembles "BkBlAqGrYeOrReViWh200" from NCAR Command Language. See Color Table Gallery

[3] See Diverging Color Maps for Scientific Visualization by

Kenneth Moreland.

[4] See A Color Map for Effective Black-and-White Rendering of Color-Scale Images by Carey Rappaport

[5] Changed to distinguish from ColorBrewer's *Spectral* map. `spectral()` still works, but `set_cmap('nipy_spectral')` is recommended for clarity.