Search the docs ...

matplotlib
matplotlib.afm
matplotlib.animation
matplotlib.artist
matplotlib.axes
matplotlib.axis
matplotlib.backend_bases
matplotlib.backend_managers
matplotlib.backend_tools
matplotlib.backends
matplotlib.bezier
matplotlib.blocking_input
matplotlib.category
matplotlib.cbook
matplotlib.cm
matplotlib.collections
matplotlib.colorbar
matplotlib.colors
matplotlib.container
matplotlib.contour
matplotlib.dates
matplotlib.docstring
matplotlib.dviread
matplotlib.figure
matplotlib.font_manager
matplotlib.fontconfig_pattern
matplotlib.gridspec
matplotlib.image
matplotlib.legend

matplotlib.pyplot.bar

`matplotlib.pyplot.bar(x, height, width=0.8, bottom=None, *, align='center', data=None, **kwargs)` [source]

Make a bar plot.

The bars are positioned at *x* with the given *align*ment. Their dimensions are given by *height* and *width*. The vertical baseline is *bottom* (default 0).

Many parameters can take either a single value applying to all bars or a sequence of values, one for each bar.

Parameters:

x : ***float or array-like***

The x coordinates of the bars. See also *align* for the alignment of the bars to the coordinates.

height : ***float or array-like***

The height(s) of the bars.

width : ***float or array-like, default: 0.8***

The width(s) of the bars.

bottom : ***float or array-like, default: 0***

The y coordinate(s) of the bars bases.

align : ***{'center', 'edge'}, default: 'center'***

Alignment of the bars to the *x* coordinates:

- 'center': Center the base on the *x* positions.
- 'edge': Align the left edges of the bars with the *x* positions.

To align the bars on the right edge pass a negative *width* and `align='edge'`.

Returns:

`BarContainer`

Container with all the bars and optionally errorbars.

Other Parameters:

color : ***color or list of color, optional***

The colors of the bar faces.

edgecolor : ***color or list of color, optional***

The colors of the bar edges.

linewidth : ***float or array-like, optional***

Width of the bar edge(s). If 0, don't draw edges.

tick_label : ***str or list of str, optional***

The tick labels of the bars. Default: None (Use default numeric labels.)

xerr, yerr : ***float or array-like of shape(N,) or shape(2, N), optional***

If not *None*, add horizontal / vertical errorbars to the bar tips. The values are +/- sizes relative to the data:

- scalar: symmetric +/- values for all bars
- shape(N,): symmetric +/- values for each bar
- shape(2, N): Separate - and + values for each bar. First row contains the lower errors, the second row contains the upper errors.
- *None*: No errorbar. (Default)

See Different ways of specifying error bars for an example on the usage of `xerr` and `yerr`.

ecolor : ***color or list of color, default: 'black'***

The line color of the errorbars.

capsize : ***float, default: `rcParams["errorbar.capsize"]` (default: `0.0`)***

The length of the error bar caps in points.

error_kw : ***dict, optional***

Dictionary of kwargs to be passed to the `errorbar` method. Values of *ecolor* or *capsize* defined here take precedence over the independent kwargs.

log : ***bool, default: False***

If *True*, set the y-axis to be log scale.

data : ***indexable object, optional***

If given, all parameters also accept a string `s`, which is interpreted as `data[s]` (unless this raises an exception).

****kwargs** : ***`Rectangle` properties***

Property	Description
`agg_filter`	a filter function, which takes a (m, n, 3) float array and a dpi value, and returns a (m, n, 3) array
`alpha`	scalar or None
`angle`	unknown
`animated`	bool
`antialiased` or aa	bool or None
`bounds`	(left, bottom, width, height)
`capstyle`	`CapStyle` or {'butt', 'projecting', 'round'}
`clip_box`	`Bbox`
`clip_on`	bool
`clip_path`	Patch or (Path, Transform) or None
`color`	color
`edgecolor` or ec	color or None
`facecolor` or fc	color or None
`figure`	`Figure`
`fill`	bool

fill	bool
gid	str
hatch	{'/', '\', '|', '-', '+', 'x', 'o', 'O', '.', '*'}
height	unknown
in_layout	bool
joinstyle	JoinStyle or {'miter', 'round', 'bevel'}
label	object
linestyle or ls	{'-', '--', '-.', ':', '', (offset, on-off-seq), ...}
linewidth or lw	float or None
path_effects	AbstractPathEffect
rasterized	bool
sketch_params	(scale: float, length: float, randomness: float)
snap	bool or None
transform	Transform
url	str
visible	bool
width	unknown
x	unknown
	(float, float)
	unknown
	float

ℹ **See also**

barh
Plot a horizontal bar plot.

Notes

Stacked bars can be achieved by passing individual *bottom* values per bar. See Stacked bar chart.

Examples using matplotlib.pyplot.bar

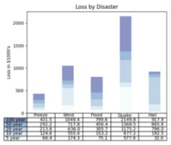
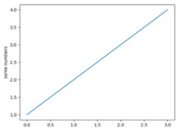


Table Demo



Pyplot tutorial

Created using Sphinx 4.3.0.