Version 2.2.2

matplotlib.pyplot.bar

`matplotlib.pyplot.bar(*args, **kwargs)` ¶ [source]

Make a bar plot.

Call signatures:

```
bar(x, height, *, align='center', **kwargs)
bar(x, height, width, *, align='center', **kwargs)
bar(x, height, width, bottom, *, align='center', **
```

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

Each of *x*, *height*, *width*, and *bottom* may either be a scalar applying to all bars, or it may be a sequence of length N providing a separate value for each bar.

Parameters:

x : sequence of scalars

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

height : scalar or sequence of scalars

The height(s) of the bars.

width : scalar or array-like, optional

The width(s) of the bars (default: 0.8).

bottom : scalar or array-like, optional

Quick search

Go

Table Of Contents

matplotlib.pyplot.bar

- Examples using `matplotlib.pyplot.bar`

Related Topics

Documentation overview

- The Matplotlib API
  - matplotlib.pyplot
    - Previous: matplotlib.pyplot.axvspan
    - Next: matplotlib.pyplot.barbs

Show Page Source

The y coordinate(s) of the bars bases (default: 0).

align : {'center', 'edge'}, optional, 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 : scalar or array-like, optional

The colors of the bar faces.

edgecolor : scalar or array-like, optional

The colors of the bar edges.

linewidth : scalar or array-like, optional

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

tick_label : string or array-like, optional

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

xerr, yerr : scalar 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

Default: None

ecolor : scalar or array-like, optional, default: 'black'

The line color of the errorbars.

capsize : scalar, optional

The length of the error bar caps in points. Default: None, which will take the value from `rcParams["errorbar.capsize"]`.

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, optional, default: False

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

orientation : {'vertical', 'horizontal'}, optional

This is for internal use only. Please use `barh` for horizontal bar plots. Default: 'vertical'.

See also

`barh`

Plot a horizontal bar plot.

Notes

The optional arguments *color*, *edgecolor*, *linewidth*, *xerr*, and *yerr* can be either scalars or sequences of length equal to the number of bars. This enables you to use bar as the basis for stacked bar charts, or candlestick plots. Detail: *xerr* and *yerr* are passed directly to `errorbar()`, so they can also have shape 2xN for independent specification of lower and upper errors.

Other optional kwargs:

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`	float or None
`animated`	bool
`antialiased` or aa	bool or None
`capstyle`	[‘butt’ | ‘round’ | ‘projecting’]
`clip_box`	a `Bbox` instance
`clip_on`	bool
`clip_path`	[(`Path`, `Transform`) | `Patch` | None]
`color`	matplotlib color spec
`contains`	a callable function
`edgecolor` or ec	mpl color spec, None, ‘none’, or ‘auto’
`facecolor` or fc	mpl color spec, or None for default, or ‘none’ for no color
`figure`	a `Figure` instance
`fill`	bool
`gid`	an id string
`hatch`	[‘/’ | ‘\’ | ‘|’ | ‘-‘ | ‘+’ | ‘x’ | ‘o’ | ‘O’ | ‘.’ | ‘*’]

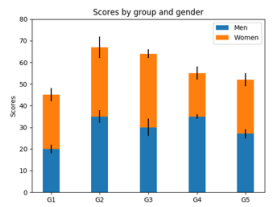
Property	Description
joinstyle	['miter' | 'round' | 'bevel']
label	object
linestyle or ls	['solid' | 'dashed', 'dashdot', 'dotted' | (offset, on-off-dash-seq) | '-' | '--' | '-.' | ':' | 'None' | ' ' | '']
linewidth or lw	float or None for default
path_effects	AbstractPathEffect
picker	[None | bool | float | callable]
rasterized	bool or None
sketch_params	(scale: float, length: float, randomness: float)
snap	bool or None
transform	Transform
url	a url string
visible	bool
zorder	float

Note

In addition to the above described arguments, this function can take a **data** keyword argument. If such a **data** argument is given, the following arguments are replaced by **data[<arg>]**:

- All arguments with the following names: 'bottom', 'color', 'ecolor', 'edgecolor', 'height', 'left', 'linewidth', 'tick_label', 'width', 'x', 'xerr', 'y', 'yerr'.
- All positional arguments.

Examples using `matplotlib.pyplot.bar`



Stacked Bar Graph

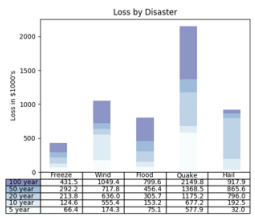
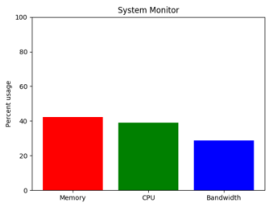
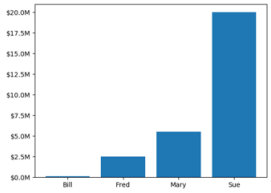


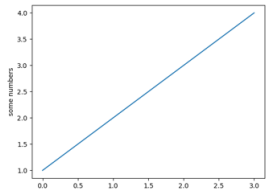
Table Demo



System Monitor



Custom Ticker1



Pyplot tutorial

Last updated on Jun 28, 2018. Created using Sphinx 1.7.5. Doc version v2.2.2-101-g15e1eadd0.