

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 *alignment*. 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.

Table of Contents

[matplotlib.pyplot.bar](#)

- [Examples using matplotlib.pyplot.bar](#)

Related Topics

[Documentation overview](#)

- [API Overview](#)
 - [matplotlib.pyplot](#)
 - [matplotlib.pyplot](#)
 - Previous: [matplotlib.pyplot.axvspan](#)
 - Next: [matplotlib.pyplot.barbs](#)

[Show Page Source](#)

- '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.

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

Property	Description
<code>agg_filter</code>	a filter function, which takes a (m, n, 3) float array and a dpi value, and returns a (m, n, 3) array
<code>alpha</code>	float or None
<code>animated</code>	bool
<code>antialiased</code> or <code>aa</code>	unknown
<code>capstyle</code>	{'butt', 'round', 'projecting'}
<code>clip_box</code>	<code>Bbox</code>
<code>clip_on</code>	bool
<code>clip_path</code>	Patch or (Path, Transform) or None
<code>color</code>	color
<code>contains</code>	unknown
<code>edgecolor</code> or <code>ec</code>	color or None or 'auto'
<code>facecolor</code> or <code>fc</code>	color or None
<code>figure</code>	<code>Figure</code>

Property	Description
<code>fill</code>	bool
<code>gid</code>	str
<code>hatch</code>	{ '/', '\', ' ', '-', '+', 'x', 'o', 'O', '.', '*' }
<code>in_layout</code>	bool
<code>joinstyle</code>	{ 'miter', 'round', 'bevel' }
<code>label</code>	object
<code>linestyle</code> or <code>ls</code>	{ '-', '--', '-.', ':', '', (offset, on-off-seq), ... }
<code>linewidth</code> or <code>lw</code>	float or None
<code>path_effects</code>	AbstractPathEffect
<code>picker</code>	None or bool or callable
<code>rasterized</code>	bool or None
<code>sketch_params</code>	(scale: float, length: float, randomness: float)
<code>snap</code>	bool or None
<code>transform</code>	Transform
<code>url</code>	str
<code>visible</code>	bool
<code>zorder</code>	float

See also

`barh`

Plot a horizontal bar plot.

Notes

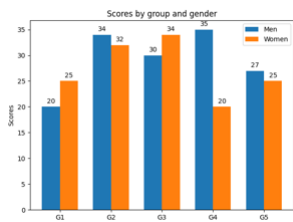
Stacked bars can be achieved by passing individual *bottom* values per bar. See [Stacked bar chart](#).

Note

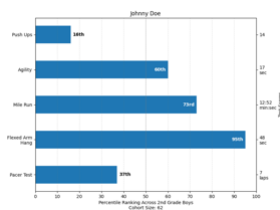
In addition to the above described arguments, this function can take a *data* keyword argument. If such a *data* argument is given, every other argument can also be string *s*, which is interpreted as `data[s]` (unless this raises an exception).

Objects passed as **data** must support item access (`data[s]`) and membership test (`s in data`).

Examples using `matplotlib.pyplot.bar`



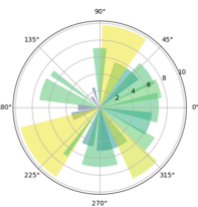
Grouped bar chart with labels



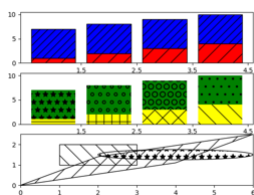
Percentiles as horizontal bar chart



Nested pie charts



Bar chart on polar axis



Hatch Demo

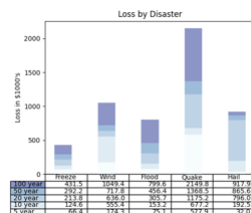
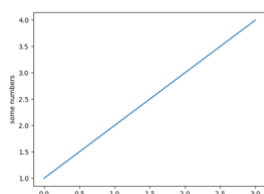
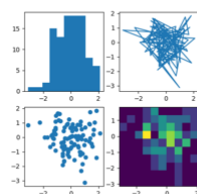


Table Demo



Pyplot tutorial



Sample plots in Matplotlib

© Copyright 2002 - 2012 John Hunter, Darren Dale, Eric Firing, Michael Droettboom and the Matplotlib development team; 2012 - 2020 The Matplotlib development team.

Last updated on Nov 12, 2020. Created using [Sphinx](#) 3.1.1. Doc version v3.3.3-2-g447bdbbf58.