

🔍 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.scatter

`matplotlib.pyplot.scatter`(*x*, *y*, *s=None*, *c=None*, *marker=None*, *cmap=None*, *norm=None*, *vmin=None*, *vmax=None*, *alpha=None*, *linewidths=None*, ***, *edgecolors=None*, *plotnonfinite=False*, *data=None*, ***kwargs*) [\[source\]](#)

A scatter plot of *y* vs. *x* with varying marker size and/or color.

Parameters: **x, y : float or array-like, shape (n,)**

The data positions.

s : float or array-like, shape (n,), optional

The marker size in points**2. Default is `rcParams['lines.markersize'] ** 2`.

c : array-like or list of colors or color, optional

The marker colors. Possible values:

- A scalar or sequence of n numbers to be mapped to colors using *cmap* and *norm*.
- A 2D array in which the rows are RGB or RGBA.
- A sequence of colors of length n.
- A single color format string.

Note that *c* should not be a single numeric RGB or RGBA sequence

because that is indistinguishable from an array of values to be

colormapped. If you want to specify the same RGB or RGBA value for all

points, use a 2D array with a single row. Otherwise, value- matching will

have precedence in case of a size matching with *x* and *y*.

If you wish to specify a single color for all points prefer the *color* keyword argument.

Defaults to `None`. In that case the marker color is determined by the value of *color*, *facecolor* or *facecolors*. In case those are not specified or `None`, the marker color is determined by the next color of the *Axes*' current "shape and fill" color cycle. This cycle defaults to `rcParams["axes.prop_cycle"]` (default: `cycler('color', ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd', '#8c564b', '#e377c2', '#7f7f7f', '#bcbd22', '#17becf'])`).

marker : MarkerStyle, default: rcParams["scatter.marker"] (default: 'o')

The marker style. *marker* can be either an instance of the class or the text shorthand for a particular marker. See [matplotlib.markers](#) for more information about marker styles.

cmap : str or Colormap, default: rcParams["image.cmap"] (default: 'viridis')

A [Colormap](#) instance or registered colormap name. *cmap* is only used if *c* is an array of floats.

norm : Normalize, default: None

If *c* is an array of floats, *norm* is used to scale the color data, *c*, in the range 0 to 1, in order to map into the colormap *cmap*. If `None`, use the default [colors.Normalize](#).

vmin, vmax : float, default: None

vmin and *vmax* are used in conjunction with the default *norm* to map the color array *c* to the colormap *cmap*. If `None`, the respective min and max of the color array is used. It is an error to use *vmin/vmax* when *norm* is given.

alpha : float, default: None

The alpha blending value, between 0 (transparent) and 1 (opaque).

linewidths : float or array-like, default: rcParams["lines.linewidth"] (default: 1.5)

The linewidth of the marker edges. Note: The default *edgecolors* is 'face'. You may want to change this as well.

edgecolors : {'face', 'none', None} or color or sequence of color, default: rcParams["scatter.edgecolors"] (default: 'face')

The edge color of the marker. Possible values:

- 'face': The edge color will always be the same as the face color.
- 'none': No patch boundary will be drawn.
- A color or sequence of colors.

For non-filled markers, *edgecolors* is ignored. Instead, the color is determined like with 'face', i.e. from *c*, *colors*, or *facecolors*.

Returns: [PathCollection](#)

Other Parameters: **data** : *indexable object, optional*

© Copyright 2002 - 2012 John Hunter, Darren Dale, Eric Firing, Michael Droettboom and the Matplotlib development team; 2012 - 2021 The Matplotlib development team.

Created using [Sphinx](#) 4.3.0.

If given, the following parameters also accept a string *s*, which is

interpreted as `data[s]` (unless this raises an exception):

x, y, s, linewidths, edgecolors, c, facecolor, facecolors, color

****kwargs** : [Collection properties](#)

See also

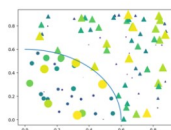
[plot](#)

To plot scatter plots when markers are identical in size and color.

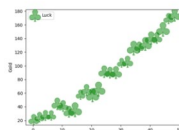
Notes

- The [plot](#) function will be faster for scatterplots where markers don't vary in size or color.
- Any or all of *x*, *y*, *s*, and *c* may be masked arrays, in which case all masks will be combined and only unmasked points will be plotted. **plotnonfinite : bool, default: False**
Whether to plot points with nonfinite *c* (i.e. `inf`, `-inf` or `nan`). If `True` the points are drawn with the `bad` colormap color (see [Colormap.set_bad](#)).
- Fundamentally, scatter works with 1D arrays, *x*, *y*, *s*, and *c* may be input as N-D arrays, but within scatter they will be flattened. The exception is *c*, which will be flattened only if its size matches the size of *x* and *y*.

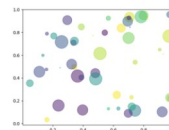
Examples using `matplotlib.pyplot.scatter`



[Scatter Masked](#)



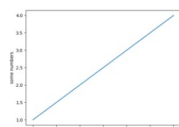
[Scatter Symbol](#)



[Scatter plot](#)



[Hyperlinks](#)



[Pyplot tutorial](#)