Fork me on GitHub

Version 2.2.2

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*, *verts=None*, *edgecolors=None*, *hold=None*, *data=None*, ***kwargs*) [source]

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

Parameters:

x, y : array_like, shape (n,)

The data positions.

s : scalar or array_like, shape (n,), optional

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

c : color, sequence, or sequence of color, optional, default: ‘b’

The marker color. Possible values:

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

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

Quick search

Go

Table Of Contents

matplotlib.pyplot.scatter

- Examples using `matplotlib.pyplot.scatter`

Related Topics

Documentation overview

- The Matplotlib API
  - matplotlib.pyplot
    - Previous: matplotlib.pyplot.sca
    - Next: matplotlib.pyplot.sci

Show Page Source

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 2-D array with a single row.

marker : `MarkerStyle`, optional, default: 'o'

The marker style. *marker* can be either an instance of the class or the text shorthand for a particular marker. See `markers` for more information marker styles.

cmap : `Colormap`, optional, default: None

A `Colormap` instance or registered colormap name. *cmap* is only used if *c* is an array of floats. If `None`, defaults to rc `image.cmap`.

norm : `Normalize`, optional, default: None

A `Normalize` instance is used to scale luminance data to 0, 1. *norm* is only used if *c* is an array of floats. If *None*, use the default `colors.Normalize`.

vmin, vmax : scalar, optional, default: None

vmin and *vmax* are used in conjunction with *norm* to normalize luminance data. If None, the respective min and max of the color array is used. *vmin* and *vmax* are ignored if you pass a *norm* instance.

alpha : scalar, optional, default: None

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

linewidths : scalar or array_like, optional, default: None

The linewidth of the marker edges. Note: The default *edgecolors* is 'face'. You may want to change this as well. If *None*, defaults to rcParams `lines.linewidth`.

verts : sequence of (x, y), optional

If *marker* is *None*, these vertices will be used to construct the marker. The center of the marker is located at (0, 0) in normalized units. The overall marker is rescaled by *s*.

edgecolors : color or sequence of color, optional, 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 matplotib color.

For non-filled markers, the *edgecolors* kwarg is ignored and forced to 'face' internally.

Returns: **paths** : `PathCollection`

Other Parameters: ****kwargs** : `Collection` properties

See also

`plot`

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

Notes

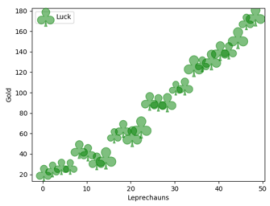
- The `plot` function will be faster for scatterplots where markers don't vary in size or color.
- Any or all of *x*, *y*, *s*, and *c* may be masked arrays, in which case all masks will be combined and only unmasked points will be plotted.
- Fundamentally, scatter works with 1-D arrays; *x*, *y*, *s*, and *c* may be input as 2-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*.

Note

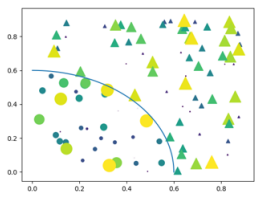
In addition to the above described arguments, this function can take a **data** keyword argument. If such a **data** argument is given, the following arguments are replaced by **data[<arg>]**:

- All arguments with the following names: 'c', 'color', 'edgecolors', 'facecolor', 'facecolors', 'linewidths', 's', 'x', 'y'.

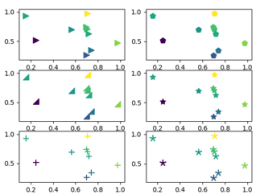
Examples using `matplotlib.pyplot.scatter`

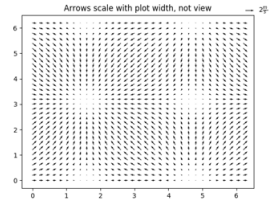


Scatter Symbol



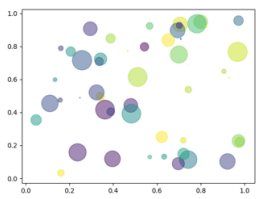
Scatter Masked





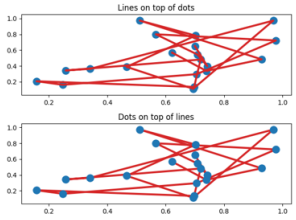
Scatter Star Poly

Demonstration of advanced quiver and quiverkey functions

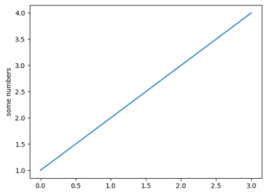


Scatter plot

Hyperlinks



Zorder Demo



Pyplot tutorial

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