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=<deprecated parameter>,
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 : scalar or 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 colors, optional

The marker color. Possible values:

- A single color format string.
- A sequence of colors of length n.
- A scalar or 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 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. Otherwise, value-matching will have precedence in case of a size matching with *x* and *y*.

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`, optional

The marker style. *marker* can be either an instance of the class or the text shorthand for a particular marker. Defaults to `None`, in which case it takes the value of `rcParams["scatter.marker"]` (default: 'o') = 'o'. See `markers` for more information about 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"]` (default: 1.5).

edgecolors : {'face', 'none', *None*} or color or sequence of color, optional.

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 Matplotlib color or sequence of color.

Defaults to `None`, in which case it takes the value of `rcParams["scatter.edgecolors"]` (default: 'face') = 'face'.

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

plotnonfinite : boolean, optional, default: False

Set to plot points with nonfinite *c*, in conjunction with `set_bad`.

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 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*.

> ### 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'.
>
> Objects passed as **data** must support item access (`data[<arg>]`) and membership test (`<arg> in data`).

Examples using `matplotlib.pyplot.scatter`

Hyperlinks

Last updated on Apr 08, 2020. Created using Sphinx 3.0.0. Doc version v3.2.1-13-g7a9bc5ebb-dirty.