matplotlib.pyplot.pie

```
matplotlib.pyplot.pie(x, explode=None, labels=None,
colors=None, autopct=None, pctdistance=0.6, shadow=False,
labeldistance=1.1, startangle=0, radius=1, counterclock=True,
wedgeprops=None, textprops=None, center=0, 0, frame=False,
rotatelabels=False, *, normalize=None, data=None)
```
[source]

Plot a pie chart.

Make a pie chart of array *x*. The fractional area of each wedge is given by `x/sum(x)`. If `sum(x) < 1`, then the values of *x* give the fractional area directly and the array will not be normalized. The resulting pie will have an empty wedge of size `1 - sum(x)`.

The wedges are plotted counterclockwise, by default starting from the x-axis.

Parameters:

x : 1D array-like

The wedge sizes.

explode : array-like, default: None

If not *None*, is a `len(x)` array which specifies the fraction of the radius with which to offset each wedge.

labels : list, default: None

A sequence of strings providing the labels for each wedge

Table of Contents

Show Page Source

colors : array-like, default: None

> A sequence of colors through which the pie chart will cycle. If *None*, will use the colors in the currently active cycle.

autopct : None or str or callable, default: None

> If not *None*, is a string or function used to label the wedges with their numeric value. The label will be placed inside the wedge. If it is a format string, the label will be `fmt % pct`. If it is a function, it will be called.

pctdistance : float, default: 0.6

> The ratio between the center of each pie slice and the start of the text generated by *autopct*. Ignored if *autopct* is *None*.

shadow : bool, default: False

> Draw a shadow beneath the pie.

normalize : None or bool, default: None

> When *True*, always make a full pie by normalizing x so that `sum(x) == 1`. *False* makes a partial pie if `sum(x) <= 1` and raises a `ValueError` for `sum(x) > 1`.
>
> When *None*, defaults to *True* if `sum(x) >= 1` and *False* if `sum(x) < 1`.

Please note that the previous default value of *None* is now deprecated, and the default will change to *True* in the next release. Please pass `normalize=False` explicitly if you want to draw a partial pie.

labeldistance : float or None, default: 1.1

The radial distance at which the pie labels are drawn. If set to `None`, label are not drawn, but are stored for use in `legend()`

startangle : float, default: 0 degrees

The angle by which the start of the pie is rotated, counterclockwise from the x-axis.

radius : float, default: 1

The radius of the pie.

counterclock : bool, default: True

Specify fractions direction, clockwise or counterclockwise.

wedgeprops : dict, default: None

Dict of arguments passed to the wedge objects making the pie. For example, you can pass in `wedgeprops = {'linewidth': 3}` to set the width of the wedge border lines equal to 3. For more details, look at the doc/arguments of the wedge object. By default `clip_on=False`.

textprops : dict, default: None

Dict of arguments to pass to the text objects.

center : (float, float), default: (0, 0)

The coordinates of the center of the chart.

frame : bool, default: False

Plot Axes frame with the chart if true.

rotatelabels : bool, default: False

Rotate each label to the angle of the corresponding slice if true.

Returns:

patches : list

A sequence of `matplotlib.patches.Wedge` instances

texts : list

A list of the label `Text` instances.

autotexts : list

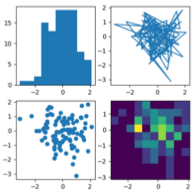
A list of `Text` instances for the numeric labels. This will only be returned if the parameter *autopct* is not *None*.

Notes

The pie chart will probably look best if the figure and Axes are square, or the Axes aspect is equal. This method sets the aspect ratio of the axis to "equal". The Axes aspect ratio can be controlled with `Axes.set_aspect`.

> ### 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 can also be string `s`, which is interpreted as `data[s]` (unless this raises an exception): *x*, *explode*, *labels*, *colors*.
>
> Objects passed as **data** must support item access (`data[s]`) and membership test (`s in data`).

Examples using `matplotlib.pyplot.pie`



Sample plots in Matplotlib

Last updated on May 08, 2021. Created using Sphinx 3.2.1. Doc version v3.4.2-2-gf801f04d09-dirty.