Help on function boxplot in matplotlib.pyplot:

matplotlib.pyplot.boxplot = boxplot(x, notch=None, sym=None, vert=None, whis=None, positions=None, widths=None, patch\_artist=None, bootstrap=None, usermedians=None, conf\_intervals=None, meanline=None, showmeans=None, showcaps=None, showbox=None, showfliers=None, boxprops=None, labels=None, flierprops=None, medianprops=None, meanprops=None, capprops=None, whiskerprops=None, manage\_xticks=True, autorange=False, zorder=None, hold=None, data=None)

Make a box and whisker plot.

Make a box and whisker plot for each column of ``x`` or each

vector in sequence ``x``. The box extends from the lower to

upper quartile values of the data, with a line at the median.

The whiskers extend from the box to show the range of the

data. Flier points are those past the end of the whiskers.

Parameters

----------

x : Array or a sequence of vectors.

The input data.

notch : bool, optional (False)

If `True`, will produce a notched box plot. Otherwise, a

rectangular boxplot is produced. The notches represent the

confidence interval (CI) around the median. See the entry

for the ``bootstrap`` parameter for information regarding

how the locations of the notches are computed.

.. note::

In cases where the values of the CI are less than the

lower quartile or greater than the upper quartile, the

notches will extend beyond the box, giving it a

distinctive "flipped" appearance. This is expected

behavior and consistent with other statistical

visualization packages.

sym : str, optional

The default symbol for flier points. Enter an empty string

('') if you don't want to show fliers. If `None`, then the

fliers default to 'b+' If you want more control use the

flierprops kwarg.

vert : bool, optional (True)

If `True` (default), makes the boxes vertical. If `False`,

everything is drawn horizontally.

whis : float, sequence, or string (default = 1.5)

As a float, determines the reach of the whiskers to the beyond the

first and third quartiles. In other words, where IQR is the

interquartile range (`Q3-Q1`), the upper whisker will extend to

last datum less than `Q3 + whis\*IQR`). Similarly, the lower whisker

will extend to the first datum greater than `Q1 - whis\*IQR`.

Beyond the whiskers, data

are considered outliers and are plotted as individual

points. Set this to an unreasonably high value to force the

whiskers to show the min and max values. Alternatively, set

this to an ascending sequence of percentile (e.g., [5, 95])

to set the whiskers at specific percentiles of the data.

Finally, ``whis`` can be the string ``'range'`` to force the

whiskers to the min and max of the data.

bootstrap : int, optional

Specifies whether to bootstrap the confidence intervals

around the median for notched boxplots. If ``bootstrap`` is

None, no bootstrapping is performed, and notches are

calculated using a Gaussian-based asymptotic approximation

(see McGill, R., Tukey, J.W., and Larsen, W.A., 1978, and

Kendall and Stuart, 1967). Otherwise, bootstrap specifies

the number of times to bootstrap the median to determine its

95% confidence intervals. Values between 1000 and 10000 are

recommended.

usermedians : array-like, optional

An array or sequence whose first dimension (or length) is

compatible with ``x``. This overrides the medians computed

by matplotlib for each element of ``usermedians`` that is not

`None`. When an element of ``usermedians`` is None, the median

will be computed by matplotlib as normal.

conf\_intervals : array-like, optional

Array or sequence whose first dimension (or length) is

compatible with ``x`` and whose second dimension is 2. When

the an element of ``conf\_intervals`` is not None, the

notch locations computed by matplotlib are overridden

(provided ``notch`` is `True`). When an element of

``conf\_intervals`` is `None`, the notches are computed by the

method specified by the other kwargs (e.g., ``bootstrap``).

positions : array-like, optional

Sets the positions of the boxes. The ticks and limits are

automatically set to match the positions. Defaults to

`range(1, N+1)` where N is the number of boxes to be drawn.

widths : scalar or array-like

Sets the width of each box either with a scalar or a

sequence. The default is 0.5, or ``0.15\*(distance between

extreme positions)``, if that is smaller.

patch\_artist : bool, optional (False)

If `False` produces boxes with the Line2D artist. Otherwise,

boxes and drawn with Patch artists.

labels : sequence, optional

Labels for each dataset. Length must be compatible with

dimensions of ``x``.

manage\_xticks : bool, optional (True)

If the function should adjust the xlim and xtick locations.

autorange : bool, optional (False)

When `True` and the data are distributed such that the 25th and

75th percentiles are equal, ``whis`` is set to ``'range'`` such

that the whisker ends are at the minimum and maximum of the

data.

meanline : bool, optional (False)

If `True` (and ``showmeans`` is `True`), will try to render

the mean as a line spanning the full width of the box

according to ``meanprops`` (see below). Not recommended if

``shownotches`` is also True. Otherwise, means will be shown

as points.

zorder : scalar, optional (None)

Sets the zorder of the boxplot.

Other Parameters

----------------

showcaps : bool, optional (True)

Show the caps on the ends of whiskers.

showbox : bool, optional (True)

Show the central box.

showfliers : bool, optional (True)

Show the outliers beyond the caps.

showmeans : bool, optional (False)

Show the arithmetic means.

capprops : dict, optional (None)

Specifies the style of the caps.

boxprops : dict, optional (None)

Specifies the style of the box.

whiskerprops : dict, optional (None)

Specifies the style of the whiskers.

flierprops : dict, optional (None)

Specifies the style of the fliers.

medianprops : dict, optional (None)

Specifies the style of the median.

meanprops : dict, optional (None)

Specifies the style of the mean.

Returns

-------

result : dict

A dictionary mapping each component of the boxplot to a list

of the :class:`matplotlib.lines.Line2D` instances

created. That dictionary has the following keys (assuming

vertical boxplots):

- ``boxes``: the main body of the boxplot showing the

quartiles and the median's confidence intervals if

enabled.

- ``medians``: horizontal lines at the median of each box.

- ``whiskers``: the vertical lines extending to the most

extreme, non-outlier data points.

- ``caps``: the horizontal lines at the ends of the

whiskers.

- ``fliers``: points representing data that extend beyond

the whiskers (fliers).

- ``means``: points or lines representing the means.

Examples

--------

.. plot:: mpl\_examples/statistics/boxplot\_demo.py

.. 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 positional and all keyword arguments.

None