pandas.DataFrame.plot

`DataFrame.`**`plot`**(*x=None*, *y=None*, *kind='line'*, *ax=None*, *subplots=False*, *sharex=None*, *sharey=False*, *layout=None*, *figsize=None*, *use_index=True*, *title=None*, *grid=None*, *legend=True*, *style=None*, *logx=False*, *logy=False*, *loglog=False*, *xticks=None*, *yticks=None*, *xlim=None*, *ylim=None*, *rot=None*, *fontsize=None*, *colormap=None*, *table=False*, *yerr=None*, *xerr=None*, *secondary_y=False*, *sort_columns=False*, ***kwds*) [source]

Make plots of DataFrame using matplotlib / pylab.

New in version 0.17.0: Each plot kind has a corresponding method on the `DataFrame.plot` accessor: `df.plot(kind='line')` is equivalent to `df.plot.line()`.

Parameters:

data : *DataFrame*

x : *label or position, default None*

y : *label or position, default None*

Allows plotting of one column versus another

kind : *str*

- 'line' : line plot (default)
- 'bar' : vertical bar plot
- 'barh' : horizontal bar plot
- 'hist' : histogram
- 'box' : boxplot
- 'kde' : Kernel Density Estimation plot
- 'density' : same as 'kde'
- 'area' : area plot
- 'pie' : pie plot
- 'scatter' : scatter plot
- 'hexbin' : hexbin plot

ax : *matplotlib axes object, default None*

subplots : *boolean, default False*

Make separate subplots for each column

sharex : *boolean, default True if ax is None else False*

In case subplots=True, share x axis and set some x axis labels to invisible; defaults to True if ax is None otherwise False if an ax is passed in; Be aware, that passing in both an ax and sharex=True will alter all x axis labels for all axis in a figure!

sharey : *boolean, default False*

In case subplots=True, share y axis and set some y axis labels to invisible

layout : *tuple (optional)*

(rows, columns) for the layout of subplots

figsize : *a tuple (width, height) in inches*

use_index : *boolean, default True*

Use index as ticks for x axis

title : *string or list*

Title to use for the plot. If a string is passed, print the string at the top of the figure. If a list is passed and *subplots* is True, print each item in the list above the corresponding subplot.

grid *: boolean, default None (matlab style default)*

Axis grid lines

legend *: False/True/'reverse'*

Place legend on axis subplots

style *: list or dict*

matplotlib line style per column

logx *: boolean, default False*

Use log scaling on x axis

logy *: boolean, default False*

Use log scaling on y axis

loglog *: boolean, default False*

Use log scaling on both x and y axes

xticks *: sequence*

Values to use for the xticks

yticks *: sequence*

Values to use for the yticks

xlim *: 2-tuple/list*

ylim *: 2-tuple/list*

rot *: int, default None*

Rotation for ticks (xticks for vertical, yticks for horizontal plots)

fontsize *: int, default None*

Font size for xticks and yticks

colormap *: str or matplotlib colormap object, default None*

Colormap to select colors from. If string, load colormap with that name from matplotlib.

colorbar *: boolean, optional*

If True, plot colorbar (only relevant for 'scatter' and 'hexbin' plots)

position *: float*

Specify relative alignments for bar plot layout. From 0 (left/bottom-end) to 1 (right/top-end). Default is 0.5 (center)

table *: boolean, Series or DataFrame, default False*

If True, draw a table using the data in the DataFrame and the data will be transposed to meet matplotlib's default layout. If a Series or DataFrame is passed, use passed data to draw a table.

yerr *: DataFrame, Series, array-like, dict and str*

See Plotting with Error Bars for detail.

xerr *: same types as yerr.*

stacked *: boolean, default False in line and*

bar plots, and True in area plot. If True, create stacked plot.

sort_columns *: boolean, default False*

Sort column names to determine plot ordering

secondary_y *: boolean or sequence, default False*

Whether to plot on the secondary y-axis If a list/tuple, which columns to plot on secondary y-axis

mark_right *: boolean, default True*

When using a secondary_y axis, automatically mark the column labels with “(right)” in the legend

kwds *: keywords*

Options to pass to matplotlib plotting method

Returns: **axes** *: matplotlib.AxesSubplot or np.array of them*

Notes

- See matplotlib documentation online for more on this subject
- If *kind* = ‘bar’ or ‘barh’, you can specify relative alignments for bar plot layout by *position* keyword. From 0 (left/bottom-end) to 1 (right/top-end). Default is 0.5 (center)
- If *kind* = ‘scatter’ and the argument *c* is the name of a dataframe column, the values of that column are used to color each point.
- If *kind* = ‘hexbin’, you can control the size of the bins with the *gridsize* argument. By default, a histogram of the counts around each *(x, y)* point is computed. You can specify alternative aggregations by passing values to the *C* and *reduce_C_function* arguments. *C* specifies the value at each *(x, y)* point and *reduce_C_function* is a function of one argument that reduces all the values in a bin to a single number (e.g. *mean*, *max*, *sum*, *std*).