

Terms

The terms will be used to refer to:

- df = Pandas DataFrame
- series = Pandas Series
- data = Pandas DataFrame or Series

Plot with Series and DataFrames

- `series.plot()` Series
- `dataframe.plot(x='None', y='None')` DataFrame
- `data.plot.<kind>()` Another method
- 'bar' or 'barh', Kinds
- 'hist', 'box', 'kde' or 'density',
- 'hexbin', 'pie' and 'scatter'

Bar Plot

- `data.plot.bar()` Bar plot/
`.barh()` Horizontal plot
- `data.plot.bar(stacked=True)` Stacked bar plot

Area Plots

- `data.plot.area()` Area plot
- `data.plot.area(stacked=False)` Non-Stacked area plot

Pie plot

- `series.plot.pie()` Pie plot for Series
- `DataFrame.plot.pie(subplots=True)` Pie plot for DataFrame
- `series.plot.pie(labels=['A','B','C'], colors=['r','b','g'], autopct='%2f')` Wedge labels

It's valid:

fontsize and figsize

Scatter plot

- `DataFrame.plot.scatter(x=' ', y=' ')` Scatter plot
- `ax= df.plot.scatter(x='A', y='B', color='None', label='Group1')`
- `df.plot.scatter(x='C', y='D', color='Other', label='Group2', ax= ax)` Multiple plot

Histograms

- `data.plot.hist()` Histogram plot
- `data.plot.hist(stacked=True, bins=10)` Stacked and bins size
- `data.plot.hist(orientation='horizontal', cumulative=True)` Horizontal and cumulative
- `data.diff().hist(color='g', alpha=0.5)` Subplots histograms

Box Plots

- `data.plot.box()` Box plot
 - `dict={'boxes':'', 'whiskers':'', 'medians':'', 'caps':''}` Color of Boxes
 - `data.plot.box(color=dict)`
 - `data.plot.box(vert=False)` Horizontal box plot
 - `df.boxplot(by='column')`
 - `df.boxplot(column=['', ''], by=['', ''])` Groupings
 - `df.groupby('g').boxplot()` for random choice
- The "choice random" is:
- `g=np.random.choice(['A', 'B'], size=50)`

Hexagonal bin plot

- `DataFrame.plot.hexbin(x='None', y='None')` Hexagonal bin plot
- `DataFrame.plot.hexbin(x='None', y='None', C='z', reduce_C_function=np.max)` add column 'z' for the value
- `DataFrame.plot.hexbin(x='None', y='None', gridsize=20)` Gridsize

Density plot

- `data.plot.kde()` Density plot

Plot for data .CSV

- > `data= pd.read_csv('Name or direction of data')`
- Andrews curves
- `pdt.andrews_curves(data, 'column name with class names')`
- Parallel coordinates
- `pdt.parallel_coordinates(data, 'column name with class names')`
- RadViz
- `pdt.radviz(data, 'column name with class name')`

Plotting Tools from Pandas Plotting

- > `import pandas.plotting as pdt`
- Scatter matrix plot
- `pdt.scatter_matrix('frame', 'alpha= 0.5', 'figsize=(6,6)')`
- Lag plot
- `pdt.lag_plot(series)`
- Autocorrelation plot
- `pdt.autocorrelation_plot(series)`
- Bootstrap plot
- `pdt.bootstrap_plot(series, size= 50, samples= 500, color='green')`

Plot formatting

Plot style

- `series.plot(style='k--')`

Controlling the legend

- `DataFrame.plot(legend= False)`

Color map

- `DataFrame.plot(colormap='')`

Scales (logarithmic)

- `data.plot(logy= True)` or `logx`
or `loglog`

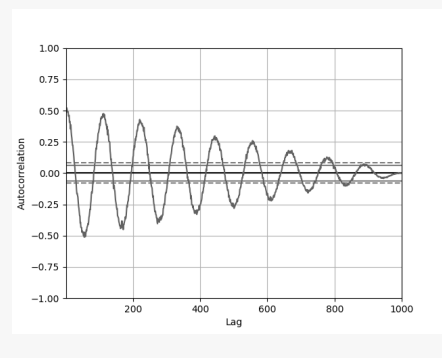
Plotting on a secondary y-axis

- `DataFrame.column1.plot()`
- `DataFrame.column2.plot(secondary_y=True)`

Suppressing tick resolution adjustment

- `data.plot(x_compat= True)`

j



Subplots

- `data.plot(subplots= True)` Subplots

- `data.plot(subplots= True, layout= (2,3))` Multiple axes

It's valid:

`figsize` and `sharex`

Plotting with errors bars

`DataFrame.plot.bar(yerr=df_err, xerr= df1_err, capsize=3)`

`df_err` and `df1_err` are DataFrame of the errors of X and Y

Plotting tables

- `ax.get_xaxis().set_visible(False)`

- `DataFrame.plot(table= True, ax=ax)`

Adds table to:

- `fig, ax= plt.subplots(1,1)`
- `pdt.table(ax, DataFrame, loc='upper right', colWidths= [0.2, 0.2, 0.2])`
- `DataFrame.plot(ax= ax)`

C

By Zayd

cheatography.com/zayd/

Published 5th June, 2020.

Last updated 5th June, 2020.

Page 2 of 2.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>