



本文紀錄Python繪圖的方法-使用 pandas

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

pandas 繪圖(參數連結)

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, **kws)` [\[source\]](#)

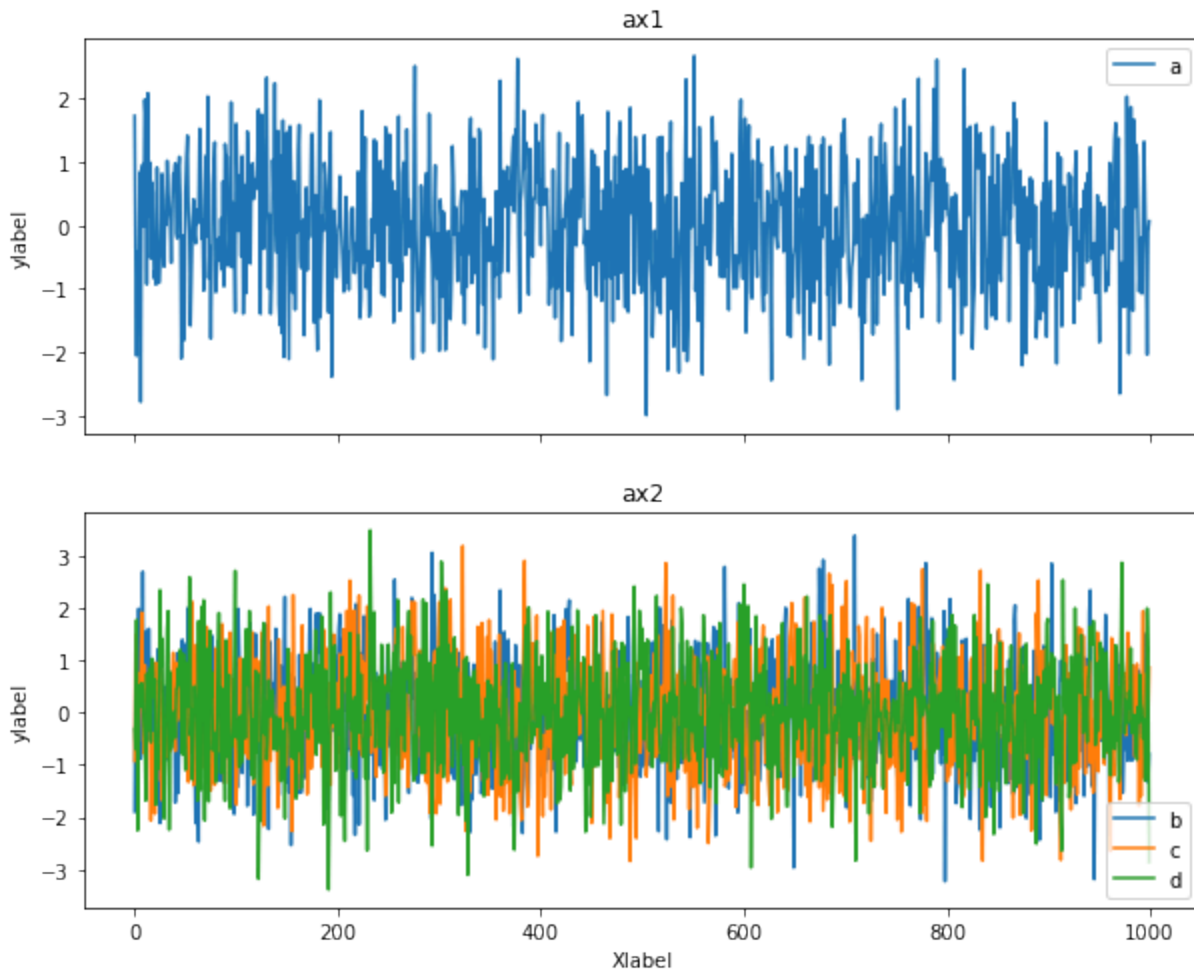
```
data=np.random.randn(1000,4)
df=pd.DataFrame(data=data,index=np.arange(1000),columns=['a','b','c','d'])
```

line plot

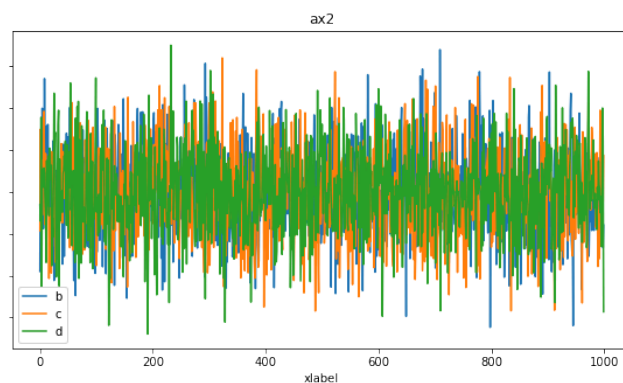
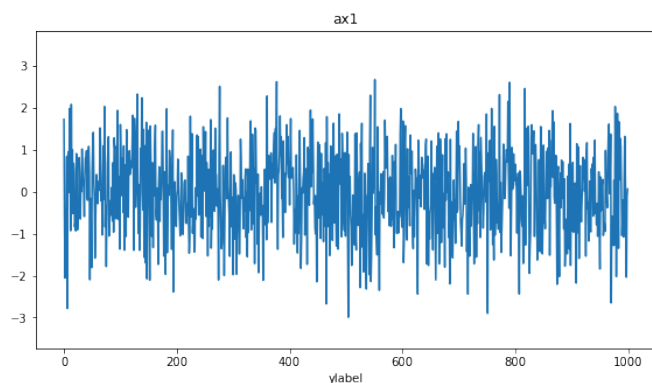
```
fig, axs = plt.subplots(2, 1, sharex=True)
df.plot(y=['a'], kind='line', ax=axs[0], title='ax1')
df.plot(y=['b', 'c', 'd'], kind='line', ax=axs[1], title='ax2', figsize=(10, 8))
axs[0].set_ylabel('ylabel')
axs[1].set_ylabel('ylabel')
axs[1].set_xlabel('Xlabel')
fig.suptitle('This is a somewhat long figure title', fontsize=16)
```

```
Text(0.5, 0.98, 'This is a somewhat long figure title')
```

This is a somewhat long figure title



```
fig, axs = plt.subplots(1, 2, sharey=True)
df.plot(y=['a'], kind='line', ax=axs[0], legend=False)
df.plot(y=['b', 'c', 'd'], kind='line', ax=axs[1], figsize=(20, 5))
#設定title
axs[0].set_title('ax1')
axs[1].set_title('ax2')
#設定label
axs[0].set_xlabel('xlabel')
axs[1].set_xlabel('xlabel')
axs[0].set_ylabel('ylabel')
#調整各個圖的間距
plt.subplots_adjust(hspace=0.5, wspace=0.1)
```



bar chart(參數連結)

pandas.DataFrame.plot.bar

`DataFrame.plot.bar(x=None, y=None, **kwargs)`

[\[source\]](#)

Vertical bar plot.

A bar plot is a plot that presents categorical data with rectangular bars with lengths proportional to the values that they represent. A bar plot shows comparisons among discrete categories. One axis of the plot shows the specific categories being compared, and the other axis represents a measured value.

Parameters:

x : label or position, optional

Allows plotting of one column versus another. If not specified, the index of the DataFrame is used.

y : label or position, optional

Allows plotting of one column versus another. If not specified, all numerical columns are used.

****kwargs**

Additional keyword arguments are documented in `pandas.DataFrame.plot()`.

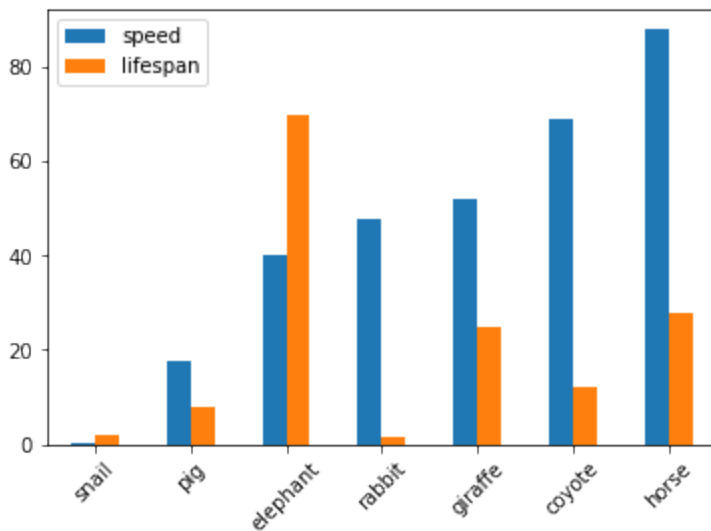
Returns:

axes : `matplotlib.axes.Axes` or `np.ndarray` of them

An ndarray is returned with one `matplotlib.axes.Axes` per column when `subplots=True`.

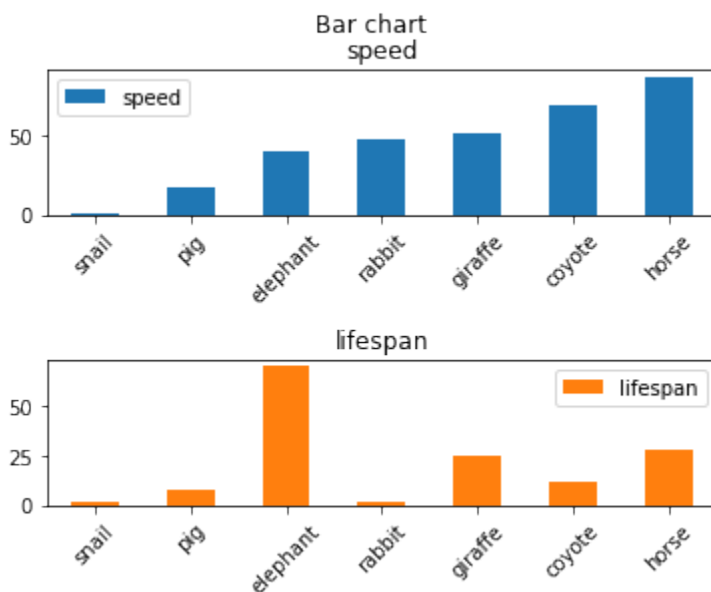
```
speed = [0.1, 17.5, 40, 48, 52, 69, 88]
lifespan = [2, 8, 70, 1.5, 25, 12, 28]
index = ['snail', 'pig', 'elephant', 'rabbit', 'giraffe', 'coyote', 'horse']
df = pd.DataFrame({'speed': speed, 'lifespan': lifespan}, index=index)
ax = df.plot.bar(rot=45) # rot表示xstick旋轉的角度
ax.legend(loc=2) # legend的位置可以用loc調整
```

```
<matplotlib.legend.Legend at 0x1bab28437c0>
```



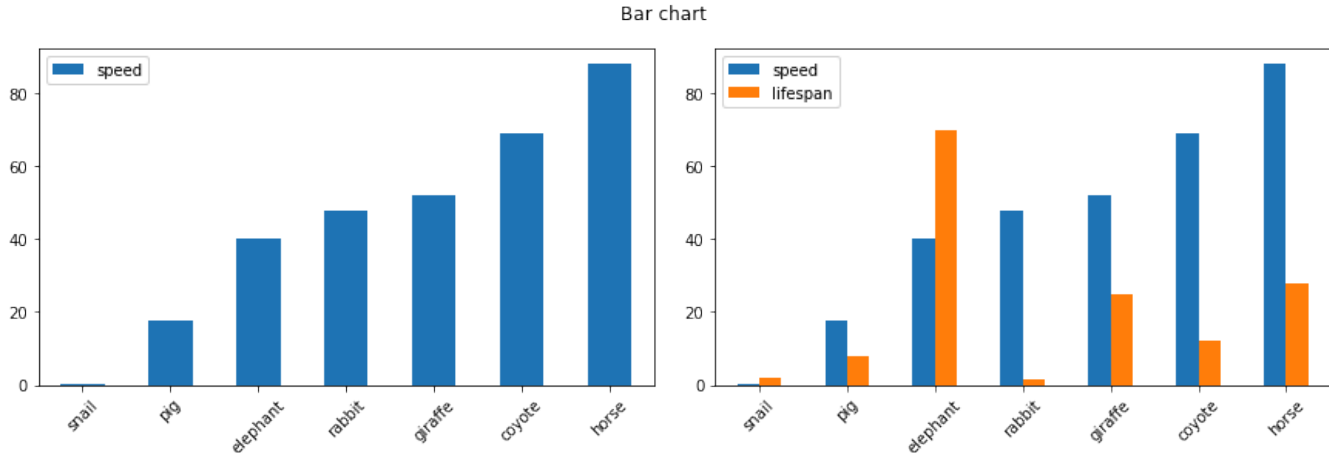
```
axes = df.plot.bar(rot=45, subplots=True, sharex=False)
axes[1].legend(loc=1)
plt.subplots_adjust(hspace=1, wspace=0.5) #調整各個ax間的距離
plt.suptitle('Bar chart')
```

```
Text(0.5, 0.98, 'Bar chart')
```



```
fig, axes = plt.subplots(1, 2, sharey=False, figsize=(15, 4))
df.plot.bar(y='speed', rot=45, ax=axes[0])
df.plot.bar(y=['speed', 'lifespan'], rot=45, ax=axes[1])
plt.subplots_adjust(wspace=0.1)
plt.suptitle('Bar chart')
```

```
Text(0.5, 0.98, 'Bar chart')
```



scatter plot chart([參數連結](#))

pandas.DataFrame.plot.scatter

DataFrame.plot.**scatter**(self, x, y, s=None, c=None, **kwargs)

[source]

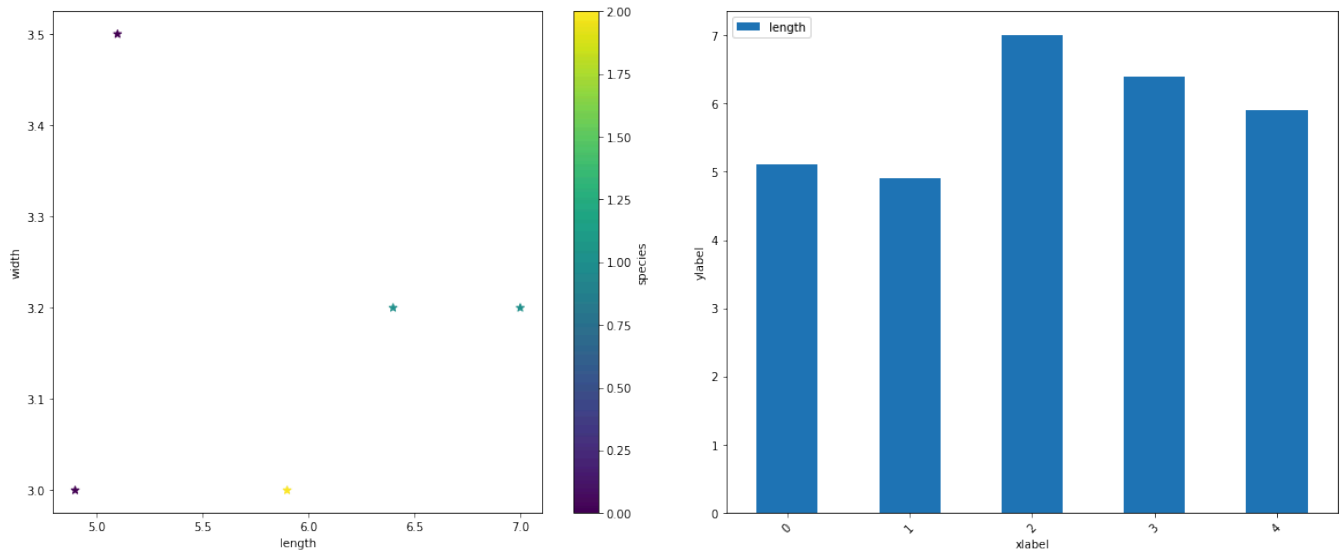
Create a scatter plot with varying marker point size and color.

The coordinates of each point are defined by two dataframe columns and filled circles are used to represent each point. This kind of plot is useful to see complex correlations between two variables. Points could be for instance natural 2D coordinates like longitude and latitude in a map or, in general, any pair of metrics that can be plotted against each other.

```
fig,axs=plt.subplots(1,2,figsize=(20,8),sharey=False)
df = pd.DataFrame([[5.1, 3.5, 0], [4.9, 3.0, 0], [7.0, 3.2, 1],[6.4, 3.2, 1], [5.9, 3.0, 2],
                    columns=['length', 'width', 'species'])
df.plot.scatter(x='length',y='width',s=50,marker='*',c='species',colormap='viridis',ax=axs[0])

df.plot.bar(y=['length'], rot=45,ax=axs[1])
axs[1].set_xlabel('xlabel')
axs[1].set_ylabel('ylabel')
axs[1].legend(loc=2)
plt.suptitle('scatter plot')
plt.subplots_adjust(wspace=0.1)
```

scatter plot



hist plot(參考連結)

pandas.DataFrame.plot.hist

`DataFrame.plot.hist(by=None, bins=10, **kwargs)` ¶

[\[source\]](#)

Draw one histogram of the DataFrame's columns.

A histogram is a representation of the distribution of data. This function groups the values of all given Series in the DataFrame into bins and draws all bins in one `matplotlib.axes.Axes`. This is useful when the DataFrame's Series are in a similar scale.

Parameters:

by : str or sequence, optional

Column in the DataFrame to group by.

bins : int, default 10

Number of histogram bins to be used.

****kwargs**

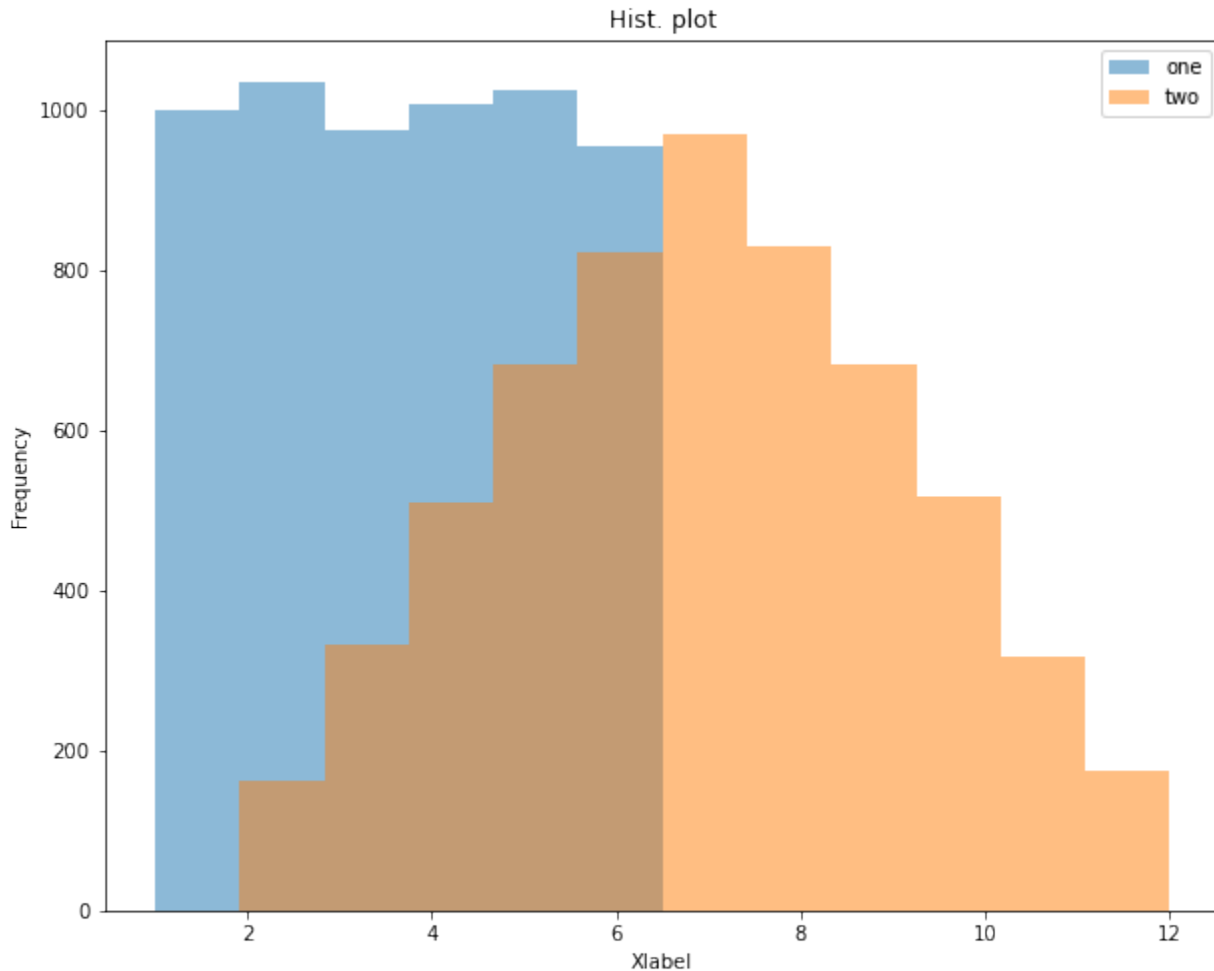
Additional keyword arguments are documented in `pandas.DataFrame.plot()`.

Returns:

axes : `matplotlib.AxesSubplot` histogram.

```
fig,ax=plt.subplots(1,1,figsize=(10,8))
df = pd.DataFrame(np.random.randint(1, 7, 6000),columns = ['one'])
df['two'] = df['one'] + np.random.randint(1, 7, 6000)
df.plot.hist(bins=12, alpha=0.5,ax=ax)
ax.set_title('Hist. plot')
ax.set_xlabel('Xlabel')
```

```
Text(0.5, 0, 'Xlabel')
```



box plot([參考連結](#))

pandas.DataFrame.boxplot

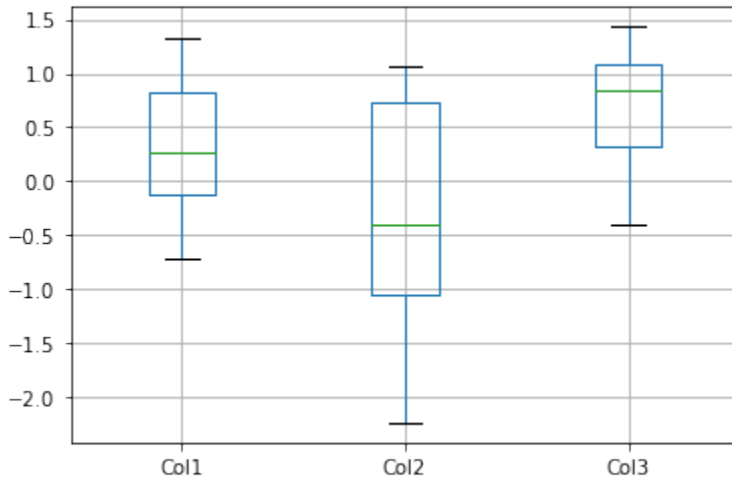
`DataFrame.boxplot(self, column=None, by=None, ax=None, fontsize=None, rot=0, grid=True, figsize=None, layout=None, return_type=None, **kws)` [\[source\]](#)

Make a box plot from DataFrame columns.

Make a box-and-whisker plot from DataFrame columns, optionally grouped by some other columns. A box plot is a method for graphically depicting groups of numerical data through their quartiles. The box extends from the Q1 to Q3 quartile values of the data, with a line at the median (Q2). The whiskers extend from the edges of box to show the range of the data. The position of the whiskers is set by default to $1.5 * IQR$ ($IQR = Q3 - Q1$) from the edges of the box. Outlier points are those past the end of the whiskers.


```
np.random.seed(1234)
df = pd.DataFrame(np.random.randn(10,4),columns=['Col1', 'Col2', 'Col3', 'Col4'])
df.boxplot(column=['Col1', 'Col2', 'Col3'])
```

<AxesSubplot:>



kde plot([參考連結](#))

pandas.DataFrame.plot.kde

DataFrame.plot.kde(*bw_method=None, ind=None, **kwargs*)

[\[source\]](#)

Generate Kernel Density Estimate plot using Gaussian kernels.

In statistics, [kernel density estimation](#) (KDE) is a non-parametric way to estimate the probability density function (PDF) of a random variable. This function uses Gaussian kernels and includes automatic bandwidth determination.

Parameters: **bw_method** : *str, scalar or callable, optional*

The method used to calculate the estimator bandwidth. This can be 'scott', 'silverman', a scalar constant or a callable. If None (default), 'scott' is used. See [scipy.stats.gaussian_kde](#) for more information.

ind : *NumPy array or int, optional*

Evaluation points for the estimated PDF. If None (default), 1000 equally spaced points are used. If *ind* is a NumPy array, the KDE is evaluated at the points passed. If *ind* is an integer, *ind* number of equally spaced points are used.

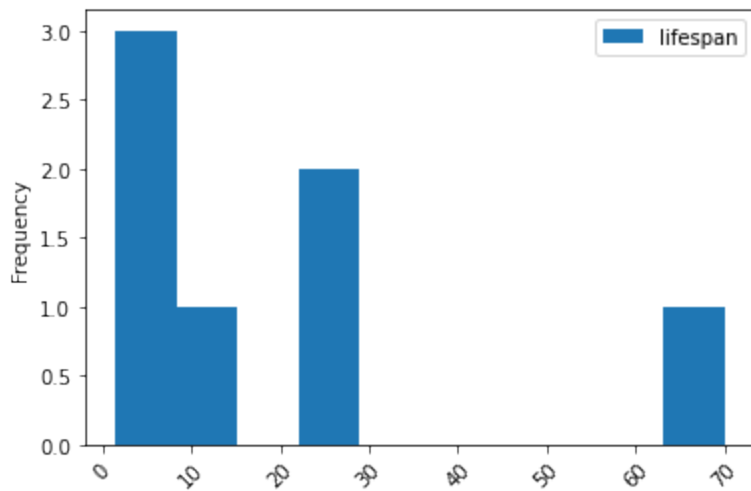
****kwargs**

Additional keyword arguments are documented in [pandas.DataFrame.plot.kde](#).

```

speed = [0.1, 17.5, 40, 48, 52, 69, 88]
lifespan = [2, 8, 70, 1.5, 25, 12, 28]
index = ['snail', 'pig', 'elephant', 'rabbit', 'giraffe', 'coyote', 'horse']
df = pd.DataFrame({'speed': speed, 'lifespan': lifespan}, index=index)
ax = df.plot.hist(y='lifespan', rot=45) #rot表示xstick旋轉的角度

```



```

df = pd.DataFrame({'x': [1, 2, 2.5, 3, 3.5, 4, 5], 'y': [4, 4, 4.5, 5, 5.5, 6, 6]},)
df.plot.kde()

```

<AxesSubplot:ylabel='Density'>

