

Informatique 7

fonctions utiles matplotlib

Rémi Emonet - 2017
Université Jean Monnet - Laboratoire Hubert Curien



« TD » de Lundi

- Inscription ?
- Écrire « TD » devant votre nom (sur la feuille d'émargement)

4 / 50 - Rémi Emonet - Informatique 7 fonctions utiles matplotlib



Informatique 7 : Plan

- (re) Numpy
- (re) Bibliothèque Matplotlib : premier contact
- Cas d'utilisation
- Matplotlib : principes appliqués

5 / 50 - Rémi Emonet - Informatique 7 fonctions utiles matplotlib



Déjà Re-Vu : tableaux, *shape*, indices, tranches

6 / 50 - Rémi Emonet - Informatique 7 fonctions utiles matplotlib



Numpy : axes d'un tableau

- Axes d'un tableau ?
 - un axe est une dimension du tableau (numérotés en partant de 0)
 - *shape* correspond donc aux nombres d'éléments selon chaque axe
- Illustration

```
a = np.arange(21).reshape((3, 7))
```

0	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20

```
print(a[2, 0])  
print(a[1:, -1])  
print(a[1, 2:5])
```

```
1 a = np.arange(21).reshape((3, 7))  
2 print(a)  
3  
4 print("Selon l'axe 0, il y a", a.shape[0], "valeurs")  
5 print("Selon l'axe 1, il y a", a.shape[1], "valeurs")
```

7 / 50 - Rémi Emonet - Informatique 7 fonctions utiles matplotlib



Numpy : axes et agrégations

- Les fonctions d'agrégation (min, max, sum, mean, ...) acceptent deux paramètres
 - **axis** : l'axe selon lequel agréger (c'est l'axe qui va « disparaître »), avec `np.sum` et un tableau à 2 dimensions :
 - `axis=0` : sommer chaque colonne
 - `axis=1` : sommer chaque ligne
 - **keepdims** : pour décider si l'axe disparaît ou est réduit à un élément
 - `keepdims=False` (par défaut), l'axe disparaît totalement
 - `keepdims=True`, l'axe est conservé, il n'a alors plus qu'un élément

Exemple

```
1 a = np.ones((3, 7))  
2 print(np.sum(a)) # 21  
3  
4 print(np.sum(a, axis=0).shape) # (7,)  
5 print(np.sum(a, axis=1).shape) # (3,)  
6  
7 print(np.sum(a, axis=0, keepdims=True).shape) # (1, 7)  
8 print(np.sum(a, axis=1, keepdims=True).shape) # (3, 1)  
9 print(np.sum(a, keepdims=True).shape) # (1, 1)
```

8 / 50 - Rémi Emonet - Informatique 7 fonctions utiles matplotlib



Moyenne d'un tableau avec `np.mean`

- Premier paramètre : tableau à traiter
- Paramètre nommé `axis` : axe selon lequel moyenner
- Paramètre nommé `keepdims` : conserve le nombre de dimensions (si `keepdims=True`)

Python Library Documentation: function mean in numpy

```
numpy.mean = mean(a, axis=None, dtype=None, out=None, keepdims=<class 'numpy._globals._NoValue'>)
```

Compute the arithmetic mean along the specified axis.

Returns the average of the array elements. The average is taken over the flattened array by default, otherwise over the specified axis. 'float64' intermediate and return values are used for integer inputs.

Parameters

a : array_like
Array containing numbers whose mean is desired. If 'a' is not an array, a conversion is attempted.
axis : None or int or tuple of ints, optional
Axis or axes along which the means are computed. The default is to compute the mean of the flattened array.



9 / 50 - Rémi Emonet - Informatique 7 fonctions utiles matplotlib

Numpy : lecture de fichier texte

- Fichier de données textuels
 - CSV : *Comma-Separated Values*
 - valeur séparées par des virgules
 - NB: le « . » est utilisé pour les nombres réels en programmation
 - il existe aussi TSV (*Tabulation-Separated Values*)
- Exemple CSV (ici, 3 valeurs par ligne)

```
1 1219,5,89.57
2 1220,9,89.66
3 1221,22,89.68
4 1222,22,90.10
5 1223,8,90.18
6 1224,16,90.34
7 1225,18,90.52
8 1226,13,90.64
9 1227,1,90.65
10 1228,2,90.90
```

- Lecture avec la fonction `genfromtxt` de Numpy

```
1 a = np.genfromtxt('monfichier.csv', delimiter=',')
2
3 print(a.shape) # (10, 3) dans l'exemple
```



18 / 50 - Rémi Emonet - Informatique 7 fonctions utiles matplotlib

Charger des données avec `np.genfromtxt`

- Premier paramètre : nom du fichier
- Paramètre nommé `delimiter` : séparateur de colonnes (ex : `","`)
- Autres paramètres pour les cas plus avancés

Python Library Documentation: function genfromtxt in numpy

```
numpy.genfromtxt = genfromtxt(fname, dtype=<class 'float'>, comments='#', delimiter=None, skip_header=0, skip_footer=0, converters=None, missing_values=None, filling_values=None, usecols=None, names=None, excludelist=None, deletechars=None, replace_space=' ', autostrip=False, case_sensitive=True, defaultfmt='%f', unpack=None, usemask=False, loose=True, invalid_raise=True, max_rows=None)
```

Load data from a text file, with missing values handled as specified.

Each line past the first 'skip_header' lines is split at the 'delimiter' character, and characters following the 'comments' character are discarded.

Parameters

fname : file, str, pathlib.Path, list of str, generator
File, filename, list, or generator to read. If the filename extension is '.gz' or '.bz2', the file is first decompressed. Note that generators must return byte strings in Python 3k. The strings in a list or produced by a generator are treated as lines.
dtype : dtype, optional
Data type of the resulting array.



20 / 50 - Rémi Emonet - Informatique 7 fonctions utiles matplotlib

Informatique 7 : Plan

- (re) Numpy
- (re) Bibliothèque Matplotlib : premier contact
- Cas d'utilisation
- Matplotlib : principes appliqués



20 / 50 - Rémi Emonet - Informatique 7 fonctions utiles matplotlib

Matplotlib : exemple

```
1 from matplotlib import pyplot as plt
2
3 x = np.linspace(0, 10, 2000)
4 y = x**3
5 z = (10+x)**2
6
7 plt.plot(x, y, label="courbe 1")
8 plt.plot(x, z, label="courbe 2")
9 plt.legend()
10
11 #plt.savefig('courbes.pdf')
12 plt.show()
13
```



21 / 50 - Rémi Emonet - Informatique 7 fonctions utiles matplotlib

Informatique 7 : Plan

- (re) Numpy
- (re) Bibliothèque Matplotlib : premier contact
- Cas d'utilisation
- Matplotlib : principes appliqués



22 / 50 - Rémi Emonet - Informatique 7 fonctions utiles matplotlib

Questionnaire QCM



Informatique 7 : Plan

- (re) Numpy
- (re) Bibliothèque Matplotlib : premier contact
- Cas d'utilisation
- **Matplotlib : principes appliqués**



Pyplot

- Matplotlib permet de
 - tracer des figures
 - tracer des courbes, histogrammes, etc
 - sauver dans un fichier image
 - afficher de manière interactive
- Pyplot : `from matplotlib import pyplot as plt`
 - `plt` : pour simplifier les tâches simples
- Quelques fonctions `plt.truc(...)`
 - `plot`, `savefig`, `show`, ...
 - `xlabel`, `ylabel`, `xticks`, `yticks`, `xlim`, `ylim`, `xscale`, `yscale`, `legend`, `grid`, `title`, ...
 - `text`, `annotate`, ...
 - `subplot`
 - `scatter`, `bar`, `barh`, `contour`, `contourf`,
 - `imshow`, `hist`, ...
- http://matplotlib.org/api/pyplot_summary.html



✓ Retour sur π : agrégation, courbes



Illustration de Numpy + Matplotlib : affichage des pas (quand on marche)



Matplotlib : apparence des courbes

- Paramètres de base de `plt.plot()`
 - codes pour la couleurs
 - codes pour le type de trait
 - codes pour le type de marqueurs (points)
- Paramètres nommés
 - pour les éléments ci-dessus (`color`, `linestyle`, `marker`)
 - pour d'autres choses : `markerfacecolor` (couleur des points), `fillstyle` (remplissage sous la courbe), `drawstyle` (comment relier les points : segment oblique, marche...), `markersize` (tailles des points), `label` (nom de la courbe), `linewidth` (épaisseur du trait),



Tracer de courbe avec `plt.plot`

- Plusieurs combinaisons de paramètres possibles
 - 2 listes "x" et "y" donnant les coordonnées des points
 - uniquement "y", un tableau de valeur ("x" est alors considéré comme `range(len(y))`)
 - optionnellement, en plus, un code couleur+trait+marqueur
- + Paramètres nommés

Python Library Documentation: function plot in matplotlib.pyplot

```
matplotlib.pyplot.plot = plot(*args, **kwargs)
```

Plot lines and/or markers to the
:class:`~matplotlib.axes.Axes`. *args* is a variable length
argument, allowing for multiple *x*, *y* pairs with an
optional format string. For example, each of the following is
legal::

```
plot(x, y)          # plot x and y using default line style and color
plot(x, y, 'bo')    # plot x and y using blue circle markers
plot(y)             # plot y using x as index array 0..N-1
plot(y, 'r+')       # ditto, but with red plusses
```



Afficher la figure avec `plt.show`

- Aucun paramètre

Python Library Documentation: function show in matplotlib.pyplot

```
matplotlib.pyplot.show = show(*args, **kw)
```

Display a figure.
When running in ipython with its pylab mode, display all
figures and return to the ipython prompt.

In non-interactive mode, display all figures and block until
the figures have been closed; in interactive mode it has no
effect unless figures were created prior to a change from
non-interactive to interactive mode (not recommended). In
that case it displays the figures but does not block.

A single experimental keyword argument, *block*, may be
set to True or False to override the blocking behavior
described above.

31 / 50 - Rémi Emonet - Informatique 7 fonctions utiles matplotlib



Exporter la figure `plt.savefig`

- Premier paramètre : nom du fichier à exporter
- ^ type en fonction de l'extension `.pdf`, `.png`, `.jpg`, ...

Python Library Documentation: function savefig in matplotlib.pyplot

```
matplotlib.pyplot.savefig = savefig(*args, **kwargs)
```

Save the current figure.

Call signature::

```
savefig(fname, dpi=None, facecolor='w', edgecolor='w',
        orientation='portrait', papertype=None, format=None,
        transparent=False, bbox_inches=None, pad_inches=0.1,
        frameon=None)
```

The output formats available depend on the backend being used.

Arguments:

```
*fname*:
A string containing a path to a filename, or a Python
file-like object, or possibly some backend-dependent object
such as :class:`~matplotlib.backends.backend_pdf.PdfPages`.
```

If *format* is *None* and *fname* is a string, the output



Nom de l'axe horizontal avec `plt.xlabel`

- Un seul paramètre (chaîne de caractères)
- idem avec `ylabel`

Python Library Documentation: function xlabel in matplotlib.pyplot

```
matplotlib.pyplot.xlabel = xlabel(s, *args, **kwargs)
```

Set the *x* axis label of the current axis.

Default override is::

```
override = {
    'fontsize'       : 'small',
    'verticalalignment' : 'top',
    'horizontalalignment' : 'center'
}
```

.. seealso::

```
:func:`~matplotlib.pyplot.text`
For information on how override and the optional args work
```

31 / 50 - Rémi Emonet - Informatique 7 fonctions utiles matplotlib



Graduations horizontales avec `plt.xticks`

- Paramètres
 - soit la liste des valeurs des graduations,
 - soit deux listes (valeurs et texte de la graduation)
- idem avec `yticks`

Python Library Documentation: function xticks in matplotlib.pyplot

```
matplotlib.pyplot.xticks = xticks(*args, **kwargs)
```

Get or set the *x*-limits of the current tick locations and labels.

```
::

# return locs, labels where locs is an array of tick locations and
# labels is an array of tick labels.
locs, labels = xticks()

# set the locations of the ticks
xticks( arange(6) )

# set the locations and labels of the ticks
xticks( arange(5), ('Tom', 'Dick', 'Harry', 'Sally', 'Sue') )
```

The keyword args, if any, are :class:`~matplotlib.text.Text`



Choisir l'intervalle d'affichage avec `plt.xlim`

- Paramètre 1 : valeur minimale selon l'axe x
- Paramètre 2 : valeur maximale selon l'axe x
- idem avec `ylim`

Python Library Documentation: function xlim in matplotlib.pyplot

```
matplotlib.pyplot.xlim = xlim(*args, **kwargs)
```

Get or set the *x* limits of the current axes.

::

```
xmin, xmax = xlim() # return the current xlim
xlim( (xmin, xmax) ) # set the xlim to xmin, xmax
xlim( xmin, xmax )  # set the xlim to xmin, xmax
```

If you do not specify args, you can pass the xmin and xmax as
kwargs, e.g.::

```
xlim(xmax=3) # adjust the max leaving min unchanged
xlim(xmin=1) # adjust the min leaving max unchanged
```

Setting limits turns autoscaling off for the x-axis.

The new axis limits are returned as a length 2 tuple.



plt.xscale

- Un paramètre : type d'échelle parmi 'linear', 'log', 'logit', 'symlog'
- Paramètres nommés selon le type d'échelle
- idem pour `yscale`

Python Library Documentation: function xscale in matplotlib.pyplot

```
matplotlib.pyplot.xscale = xscale(*args,**kwargs)
```

Set the scaling of the *x*-axis.

call signature::

```
xscale(scale, **kwargs)
```

The available scales are: 'linear' | 'log' | 'logit' | 'symlog'

Different keywords may be accepted, depending on the scale:

'linear'

'log'

36 / 50 - Rémi Emonet - Informatique 7 Fonctions utiles matplotlib



Afficher la légende avec plt.legend

- Paramètres nommés intéressants : `loc` (placement de la légende), `ncol` (nombre de colonnes), `fontsize` (taille du texte)

Python Library Documentation: function legend in matplotlib.pyplot

```
matplotlib.pyplot.legend = legend(*args,**kwargs)
```

Places a legend on the axes.

To make a legend for lines which already exist on the axes (via plot for instance), simply call this function with an iterable of strings, one for each legend item. For example::

```
ax.plot([1, 2, 3])
ax.legend(['A simple line'])
```

However, in order to keep the "label" and the legend element instance together, it is preferable to specify the label either at artist creation, or by calling the :meth:`~matplotlib.artist.Artist.set\_label` method on the artist::

```
line, = ax.plot([1, 2, 3], label='Inline label')
# Overwrite the label by calling the method.
line.set_label('Label via method')
ax.legend()
```

37 / 50 - Rémi Emonet - Informatique 7 Fonctions utiles matplotlib



Ajouter une grille avec plt.grid

Python Library Documentation: function grid in matplotlib.pyplot

```
matplotlib.pyplot.grid = grid(b=None, which='major', axis='both', **kwargs)
```

Turn the axes grids on or off.

Call signature::

```
grid(self, b=None, which='major', axis='both', **kwargs)
```

Set the axes grids on or off; *b* is a boolean. (For MATLAB compatibility, *b* may also be a string, 'on' or 'off'.)

If *b* is *None* and ``len(kwargs)==0``, toggle the grid state. If *kwargs* are supplied, it is assumed that you want a grid and *b* is thus set to *True*.

which can be 'major' (default), 'minor', or 'both' to control whether major tick grids, minor tick grids, or both are affected.

axis can be 'both' (default), 'x', or 'y' to control which set of gridlines are drawn.

kwargs are used to set the grid line properties, e.g.,::

```
ax.grid(color='r', linestyle='-', linewidth=2)
```

Valid :class:`~matplotlib.lines.Line2D` kwargs are

38 / 50 - Rémi Emonet - Informatique 7 Fonctions utiles matplotlib



Ajouter un titre avec plt.title

- ...

Python Library Documentation: function title in matplotlib.pyplot

```
matplotlib.pyplot.title = title(s, *args,**kwargs)
```

Set a title of the current axes.

Set one of the three available axes titles. The available titles are positioned above the axes in the center, flush with the left edge, and flush with the right edge.

.. seealso::

See :func:`~matplotlib.pyplot.text` for adding text to the current axes

Parameters

label : str

Text to use for the title

fontdict : dict

A dictionary controlling the appearance of the title text, the default 'fontdict' is:

```
{'fontsize': rcParams['axes.titlesize'],
 'fontweight': rcParams['axes.titleweight']}
```

39 / 50 - Rémi Emonet - Informatique 7 Fonctions utiles matplotlib



Du texte dans un graphique avec plt.text

- Paramètre 1 : position "x" du texte
- Paramètre 2 : position "y" du texte
- Paramètre 3 : texte à afficher (chaîne de caractères)
- Paramètres nommés intéressants : `fontsize` (taille du texte), `horizontalalignment` (alignement du texte horizontalement, par rapport à x,y), `verticalalignment`

Python Library Documentation: function text in matplotlib.pyplot

```
matplotlib.pyplot.text = text(x, y, s, fontdict=None, withdash=False, **kwargs)
```

Add text to the axes.

Add text in string `s` to axis at location `x`, `y`, data coordinates.

Parameters

x, y : scalars
data coordinates

s : string
text

40 / 50 - Rémi Emonet - Informatique 7 Fonctions utiles matplotlib



Annoter un point avec plt.annotate

- Paramètre 1 : texte à afficher (chaîne de caractères)
- Paramètre 2 : paire (2-uplet) position "x,y" du point à annoter
- Paramètre 3 : paire (2-uplet) position "x,y" du texte
- Paramètres nommés intéressants : `arrowprops=dict(arrowstyle="->")` (pour tracer une flèche), `xycoords` (repère pour les coordonnées du point annoté), `textcoords` (repère pour les coordonnées du texte)

Python Library Documentation: function annotate in matplotlib.pyplot

```
matplotlib.pyplot.annotate = annotate(*args,**kwargs)
```

Create an annotation: a piece of text referring to a data point.

Parameters

s : string
label

xy : (x, y)

position of element to annotate. See *xycoords* to control what

41 / 50 - Rémi Emonet - Informatique 7 Fonctions utiles matplotlib



Faire des sous graphiques avec `plt.subplot`

- Paramètre 1 : nombre de lignes de graphiques
- Paramètre 2 : nombre de colonnes de graphiques
- Paramètre 3 : numéro/position du graphique à tracer avec le prochain `plt.plot` (début à 1, sens de la lecture (lignes d'abord))

Python Library Documentation: function subplot in matplotlib.pyplot

`matplotlib.pyplot.subplot` = subplot(*args,**kwargs)

Return a subplot axes positioned by the given grid definition.

Typical call signature::

```
subplot(nrows, ncols, plot_number)
```

Where `*nrows*` and `*ncols*` are used to notionally split the figure into ```nrows * ncols``` sub-axes, and `*plot_number*` is used to identify the particular subplot that this function is to create within the notional grid. `*plot_number*` starts at 1, increments across rows first and has a maximum of ```nrows * ncols```.

In the case when `*nrows*`, `*ncols*` and `*plot_number*` are all less than 10, a convenience exists, such that the a 3 digit number can be given instead, where the hundreds represent `*nrows*`, the tens represent `*ncols*` and the



Tracer un nuage de points avec `plt.scatter`

- Paramètre 1 : liste/tableau des positions horizontales
- Paramètre 2 : liste/tableau des positions verticales

Python Library Documentation: function scatter in matplotlib.pyplot

`matplotlib.pyplot.scatter` = scatter(x, y, s=20, c=None, marker='o', cmap=None, norm=None, vmin=None, vmax=None, alpha=None, linewidths=None, verts=None, edgecolors=None, hold=None, data=None, **kwargs)

Make a scatter plot of x vs y, where x and y are sequence like objects of the same lengths.

Parameters

x, y : array_like, shape (n,)
Input data

s : scalar or array_like, shape (n,), optional, default: 20
size in points².

c : color or sequence of color, optional, default : 'b'
'c' can be a single color format string, or a sequence of color specifications of length 'N', or a sequence of 'N' numbers to be mapped to colors using the 'cmap' and 'norm' specified via kwargs (see below). Note that 'c' should not be a single numeric RGB or RGBA sequence because that is indistinguishable from an array of values to be colormapped. 'c' can be a 2-D array in which the



Tracer un diagramme en barres avec `plt.bar`

- Paramètre 1 : liste/tableau des positions horizontales (gauche de chaque barre)
- Paramètre 2 : liste/tableau des hauteurs (valeur verticale)
- Paramètre nommé `width` : largeur de l'ensemble des barres

Python Library Documentation: function bar in matplotlib.pyplot

`matplotlib.pyplot.bar` = bar(left, height, width=0.8, bottom=None, hold=None, data=None, **kwargs)

Make a bar plot.

Make a bar plot with rectangles bounded by:

```
`left`, `left` + `width`, `bottom`, `bottom` + `height`  
(left, right, bottom and top edges)
```

Parameters

left : sequence of scalars
the x coordinates of the left sides of the bars

height : sequence of scalars
the heights of the bars



... barres horizontales avec `plt.barh`

- Paramètre 1 : liste/tableau des positions verticale (bas de chaque barre)
- Paramètre 2 : liste/tableau des largeur (valeur horizontale)
- Paramètre nommé `height` : hauteur de l'ensemble des barres

Python Library Documentation: function barh in matplotlib.pyplot

`matplotlib.pyplot.barh` = barh(bottom, width, height=0.8, left=None, hold=None, **kwargs)

Make a horizontal bar plot.

Make a horizontal bar plot with rectangles bounded by:

```
`left`, `left` + `width`, `bottom`, `bottom` + `height`  
(left, right, bottom and top edges)
```

'bottom', 'width', 'height', and 'left' can be either scalars or sequences

Parameters

bottom : scalar or array-like
the y coordinate(s) of the bars

width : scalar or array-like
the width(s) of the bars



Tr. des lignes de niveau avec `plt.contour`

- Plusieurs combinaisons de paramètres possibles
 - soit un tableau 2D de valeurs "z"
 - soit 3 tableaux 1D/1D/2D (ou 2D/2D/2D) de valeurs "x", "y", "z" (pour donner les positions sur les axes x et y)
 - dans les deux cas, optionnellement en plus, une liste des valeurs des lignes de niveaux
- Paramètres nommés intéressants : `extent` (bornes x et y), `cmap` (couleurs), `alpha` (opacité), `norm` (normalisation des "z")

Python Library Documentation: function contour in matplotlib.pyplot

`matplotlib.pyplot.contour` = contour(*args,**kwargs)

Plot contours.

:func:`~matplotlib.pyplot.contour` and
:func:`~matplotlib.pyplot.contourf` draw contour lines and filled contours, respectively. Except as noted, function signatures and return values are the same for both versions.

:func:`~matplotlib.pyplot.contourf` differs from the MATLAB version in that it does not draw the polygon edges.



... et en remplissant, avec `plt.contourf`

- idem

Python Library Documentation: function contourf in matplotlib.pyplot

`matplotlib.pyplot.contourf` = contourf(*args,**kwargs)

Plot contours.

:func:`~matplotlib.pyplot.contour` and
:func:`~matplotlib.pyplot.contourf` draw contour lines and filled contours, respectively. Except as noted, function signatures and return values are the same for both versions.

:func:`~matplotlib.pyplot.contourf` differs from the MATLAB version in that it does not draw the polygon edges.
To draw edges, add line contours with
calls to :func:`~matplotlib.pyplot.contour`.

Call signatures::

```
contour(Z)
```

make a contour plot of an array *Z*. The level values are chosen automatically.

::



Afficher une « image » avec plt.imshow

- Premier paramètre : tableau 2D de valeurs (comme "z" pour contour)
- Paramètres nommés intéressants : interpolation ("nearest" pour éviter le flou), extent (bornes x et y), cmap (couleurs), alpha (opacité), norm (normalisation des "z")

Python Library Documentation: function imshow in matplotlib.pyplot

matplotlib.pyplot.imshow = imshow(X, cmap=None, norm=None, aspect=None, interpolation=None, alpha=None, vmin=None, vmax=None, origin=None, extent=None, shape=None, filternorm=1, filterrad=4.0, imlim=None, resample=None, url=None, hold=None, data=None, **kwargs)

Display an image on the axes.

Parameters

X : array_like, shape (n, m) or (n, m, 3) or (n, m, 4)
Display the image in 'X' to current axes. 'X' may be a float array, a uint8 array or a PIL image. If 'X' is an array, it can have the following shapes:

- MxN -- luminance (grayscale, float array only)
- MxNx3 -- RGB (float or uint8 array)
- MxNx4 -- RGBA (float or uint8 array)

The value for each component of MxNx3 and MxNx4 float arrays



Calculer un histogramme avec plt.hist

- Paramètre 1 : tableau de valeurs dont on veut l'histogramme
- Paramètre 2 (nommé bins) : nombre de cases dans l'histogramme
- ...

Python Library Documentation: function hist in matplotlib.pyplot

matplotlib.pyplot.hist = hist(x, bins=10, range=None, normed=False, weights=None, cumulative=False, bottom=None, histtype='bar', align='mid', orientation='vertical', rwidth=None, log=False, color=None, label=None, stacked=False, hold=None, data=None, **kwargs)

Plot a histogram.

Compute and draw the histogram of *x*. The return value is a tuple (*n*, *bins*, *patches*) or ([*n0*, *n1*, ...], *bins*, [*patches0*, *patches1*,...]) if the input contains multiple data.

Multiple data can be provided via *x* as a list of datasets of potentially different length ([*x0*, *x1*, ...]), or as a 2-D ndarray in which each column is a dataset. Note that the ndarray form is transposed relative to the list form.

Masked arrays are not supported at present.

Parameters

