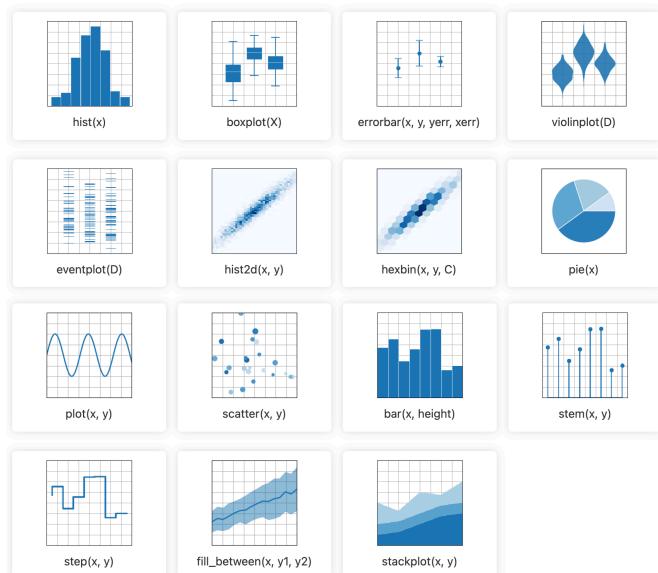


10. Matplotlib 套件

章節內容

- 安裝
- 基本概念
- Pyplot
- Plot 流程
- 個人化風格
- Legend 圖標
- Color/colormap
- 複雜和語義的圖形構成
- Text in Matplotlib
- Toolkits



為什麼要數據可視化？

良好的分析來自文字和可視化

更容易被人類閱讀和理解

在人類記憶中留下更長時間的印象

讓複雜的事情變得簡單

更快的決策

以更好的方式分析數據



Matplotlib 套件

- 用於在 Python 中創建可視化的綜合庫 *Created by John D. Hunter*
- Written in Python, C, and JavaScript with 開源
- 數據分析師使用最廣泛的可視化工具之一 *Latest stable version: 3.9.x*

<https://github.com/matplotlib/matplotlib>

安裝 Install using PIP

- Open terminal (or iterm2, etc), and in ~ root location

`python3 -m install matplotlib`
- This will install matplotlib and its needed dependencies
- Then, launch Jupyter Notebook from terminal

`jupyter notebook`

The screenshot shows the Jupyter Notebook interface. At the top, there's a navigation bar with 'jupyter' logo, 'Files', 'Running', 'Clusters', 'Quit', and 'Logout'. Below the navigation bar is a file list table with columns for 'Name', 'Last Modified', and 'File size'. The file list includes 'Applications', 'Desktop', 'django', 'Documents', and 'Downloads'.

在 Jupyter Notebook

按 New to initiate new notebook, 及 import the matplotlib

```
import matplotlib as plt
```

檢查 version

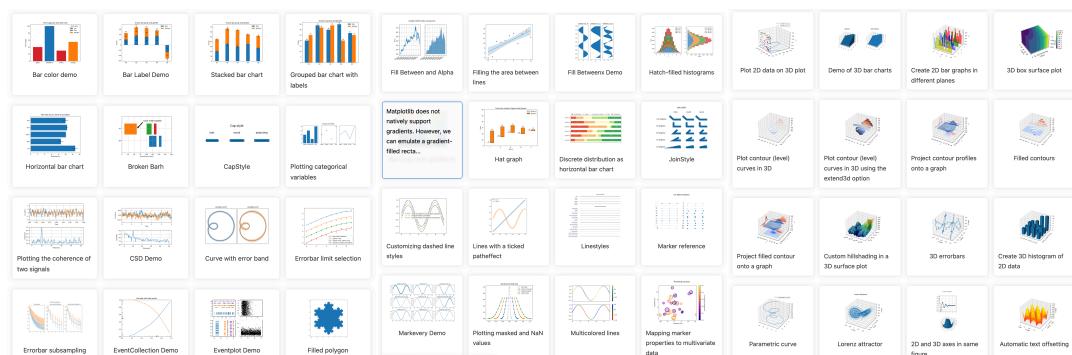
```
plt.__version__
```

```
'3.5.2'
```

The screenshot shows a Jupyter Notebook cell with two code blocks. The first block contains `In [2]: import matplotlib as plt`. The second block contains `In [3]: plt.__version__`, which has an output of `Out[3]: '3.5.2'`. Below the cell is an empty input field labeled `In []:`.

圖庫中有多種類型和不同的圖表。比想像的要多。

然而，普通和流行的總是給讀者留下深刻印象。



Matplotlib Interfaces 兩大介面

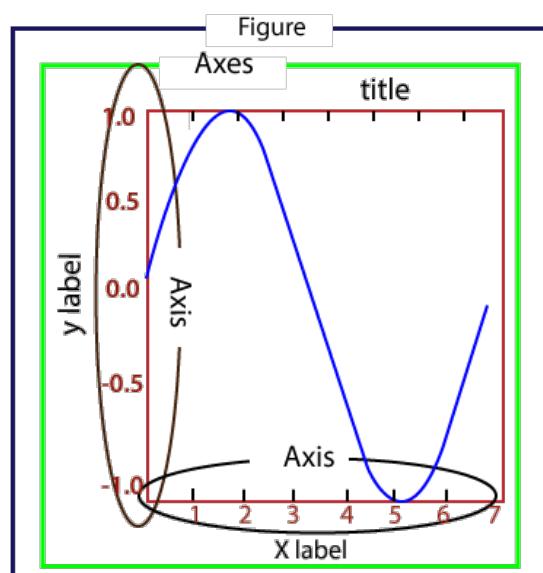
Matplotlib 具有兩個主要的應用程式介面，或使用庫的樣式：

- 顯式(explicit)"axes" 使用圖形或軸物件上的方法創建其他 Artists, 並逐步構建可視化。
- 隱式(implicit)"pyplot" 跟蹤最後一個的介面 Figure and Axes 創建，並添加 Artists 到它認為使用者想要的物件。

Matplotlib 架構

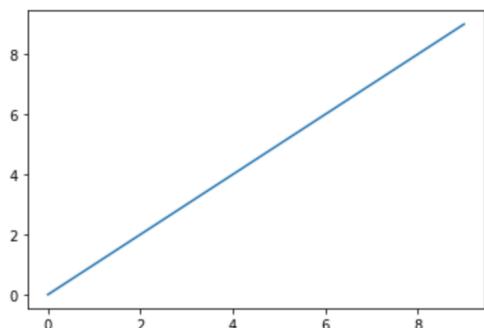
該圖分為幾個部分

- Figure: a canvas to draw plots, can contain multiple subplot
- Axes: can contain several Axes, and each of them contain their own x-label and y-label
- Axis: number of line or object for graph limit
- x-label, y-label
- title



Matplotlib 技巧 簡易操作

```
import matplotlib.pyplot as plt  
  
data = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
  
plt.plot(data)  
  
[<matplotlib.lines.Line2D at 0x10ea5f850>]
```



圖型類別-1

ax.bar()	Vertical rectangles
ax.bart()	Horizontal rectangles
ax.axhline()	Horizontal line across axes
ax.vline()	Vertical line across axes
ax.fill()	Filled polygons
ax.fill_between()	Fill between y-values and 0
ax.stackplot()	Stack plot

圖型類別-2

ax.arrow()	Arrow
ax.quiver()	2D field of arrows
ax.streamplot()	2D vector fields
ax.hist()	Histogram
ax.boxplot()	Boxplot
ax.violinplot()	Violinplot

圖型類別-3

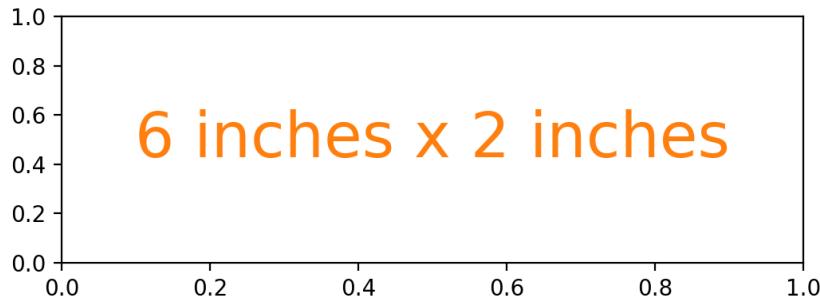
ax.pcolor()	Pseudocolor plot
ax.pcolormesh()	Pseudocolor plot
ax.contour()	Contour plot
ax.contourf()	Filled contour plot
ax.clabel()	Labeled contour plot

圖形尺寸（以英寸為單位）（預設）

```
import matplotlib.pyplot as plt

text_kwargs = dict(ha='center', va='center', fontsize=28, color='C1')

plt.subplots(figsize=(6, 2))
plt.text(0.5, 0.5, '6 inches x 2 inches', **text_kwargs)
plt.show()
```



設置 title, axes label, and legend

更改繪圖 title and axes labels, 您可以遵循以下方法之一，具體取決於要使用的容器:

- 正確設置這些事情的最簡單方法是使用 `ax.set(title="A title", xlabel="x", ylabel="y")` or `ax.set_xlim(), ax.set_ylim()` or `ax.set_title()`.
- 如果你想使用這個數位，你也可以求助於 `fig.suptitle()` to add a title to your plot.
- 如果使用包必須提供的預設設置，則可能需要使用 `plt.title()`, `plt.xlabel()`, `plt.ylabel()`
- 用 `ax.legend()` to 用顏色表示專案，顯示圖示

Legend 圖示

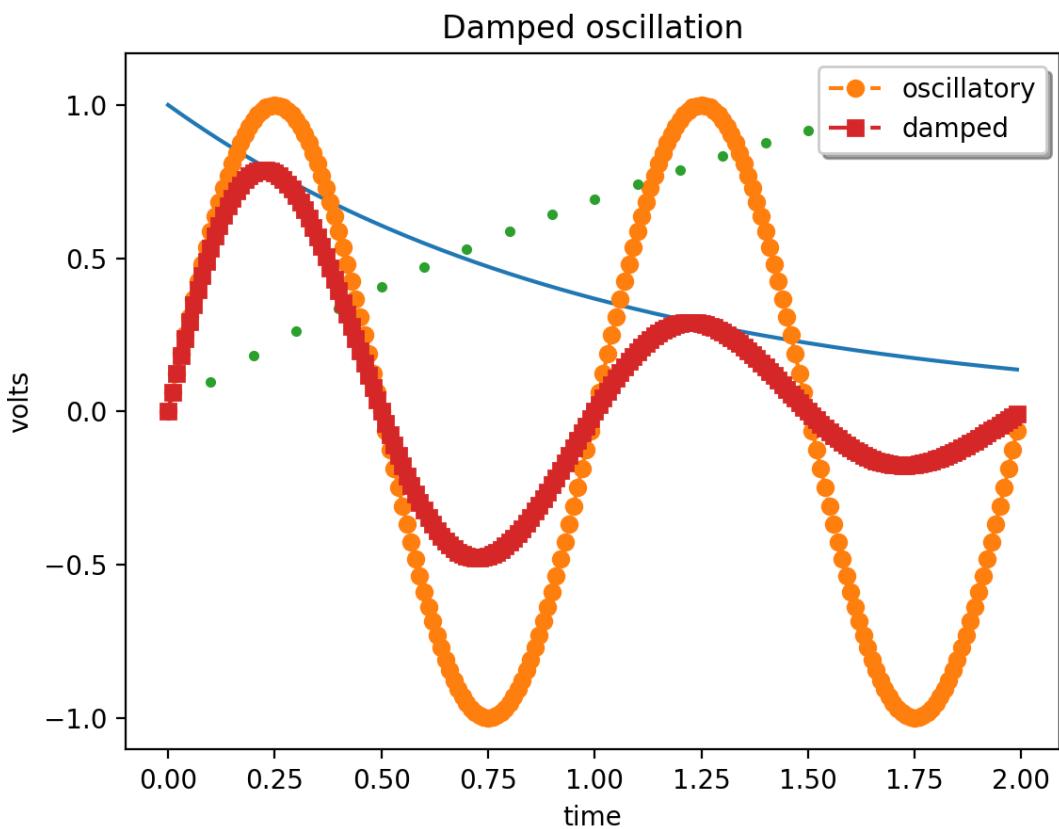
```
import matplotlib.pyplot as plt
import matplotlib.collections as mcol
from matplotlib.legend_handler import HandlerLineCollection, HandlerTuple
from matplotlib.lines import Line2D
import numpy as np

t1 = np.arange(0.0, 2.0, 0.1)
t2 = np.arange(0.0, 2.0, 0.01)

fig, ax = plt.subplots(figsize=(8, 6))

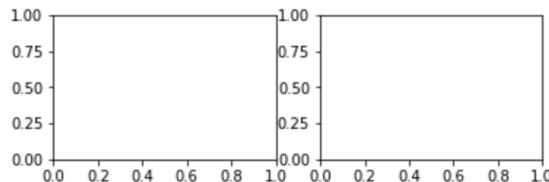
# note that plot returns a list of lines. The "l1, = plot" usage
# extracts the first element of the list into l1 using tuple
# unpacking. So l1 is a Line2D instance, not a sequence of lines
l1, = ax.plot(t2, np.exp(-t2))
l2, l3 = ax.plot(t2, np.sin(2 * np.pi * t2), '--o', t1, np.log(1 + t1), '.')
l4, = ax.plot(t2, np.exp(-t2) * np.sin(2 * np.pi * t2), 's-')

ax.legend((l2, l4), ('oscillatory', 'damped'), loc='upper right', shadow=True)
ax.set_xlabel('time')
ax.set_ylabel('volts')
ax.set_title('Damped oscillation')
plt.show()
```



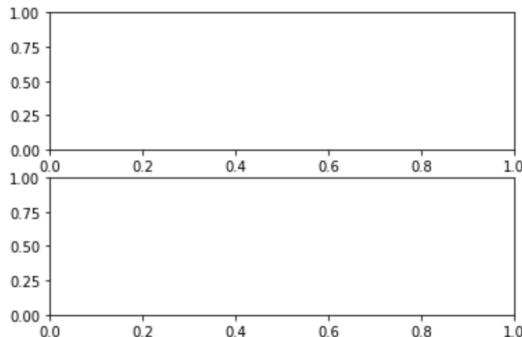
圖和子圖概念 - 1

```
import matplotlib.pyplot as plt
fig = plt.figure()
ax1 = fig.add_subplot(2, 2, 1)
ax2 = fig.add_subplot(2, 2, 2)
```



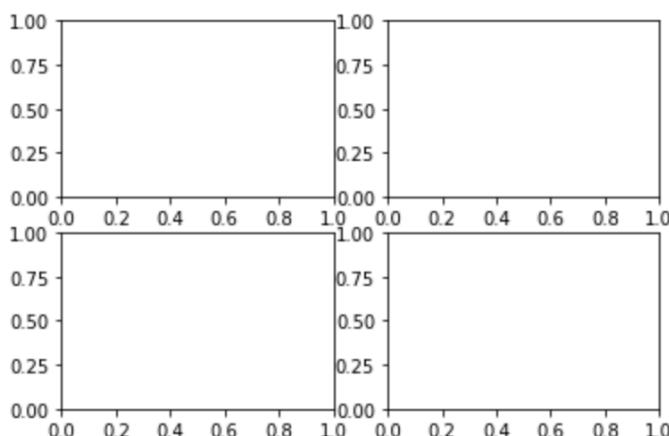
圖和子圖概念 - 2

```
import matplotlib.pyplot as plt
fig = plt.figure()
ax1 = fig.add_subplot(2, 1, 1)
ax2 = fig.add_subplot(2, 1, 2)
```



圖和子圖概念 - 3

```
import matplotlib.pyplot as plt
fig = plt.figure()
ax1 = fig.add_subplot(2, 2, 1)
ax2 = fig.add_subplot(2, 2, 2)
ax3 = fig.add_subplot(2, 2, 3)
ax4 = fig.add_subplot(2, 2, 4)
```



Pyplot.subplots 選擇

Argument	Description
nrows	Number of rows of subplots
ncols	Number of columns of subplots
sharex	All subplots should use the same x-axis ticks (adjusting the <code>xlim</code> will affect all subplots)
sharey	All subplots should use the same y-axis ticks (adjusting the <code>ylim</code> will affect all subplots)
subplot_kw	Dict of keywords passed to <code>add_subplot</code> call used to create each subplot
**fig_kw	Additional keywords to <code>subplots</code> are used when creating the figure, such as <code>plt.subplots(2, 2, figsize=(8, 6))</code>

帶子圖的折線圖

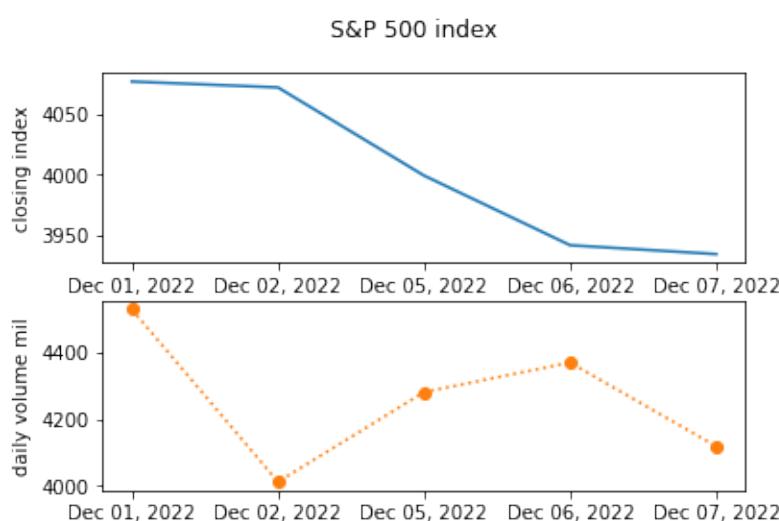
```
import matplotlib.pyplot as plt

dates = ['Dec 01, 2022', 'Dec 02, 2022', 'Dec 05, 2022',
         'Dec 06, 2022', 'Dec 07, 2022']
idx = [4076.57, 4071.70, 3998.84, 3941.26, 3933.92]
vol = [4527, 4012, 4280, 4368, 4118]

plt.figure()
plt.subplot(211)
plt.plot(dates, idx)
plt.ylabel('closing index')

plt.subplot(212)
plt.plot(dates, vol, 'tab:orange',
          marker = 'o', linestyle = 'dotted')
plt.ylabel('daily volume mil')

plt.suptitle('S&P 500 index')
plt.show()
```



折線圖並用 alpha 填充

```
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.cbook as cbook

# load up some sample financial data
r = (cbook.get_sample_data('goog.npz', np_load=True)['price_data']
     .view(np.recarray))
# create two subplots with the shared x and y axes
fig, (ax1, ax2) = plt.subplots(1, 2, sharex=True, sharey=True)

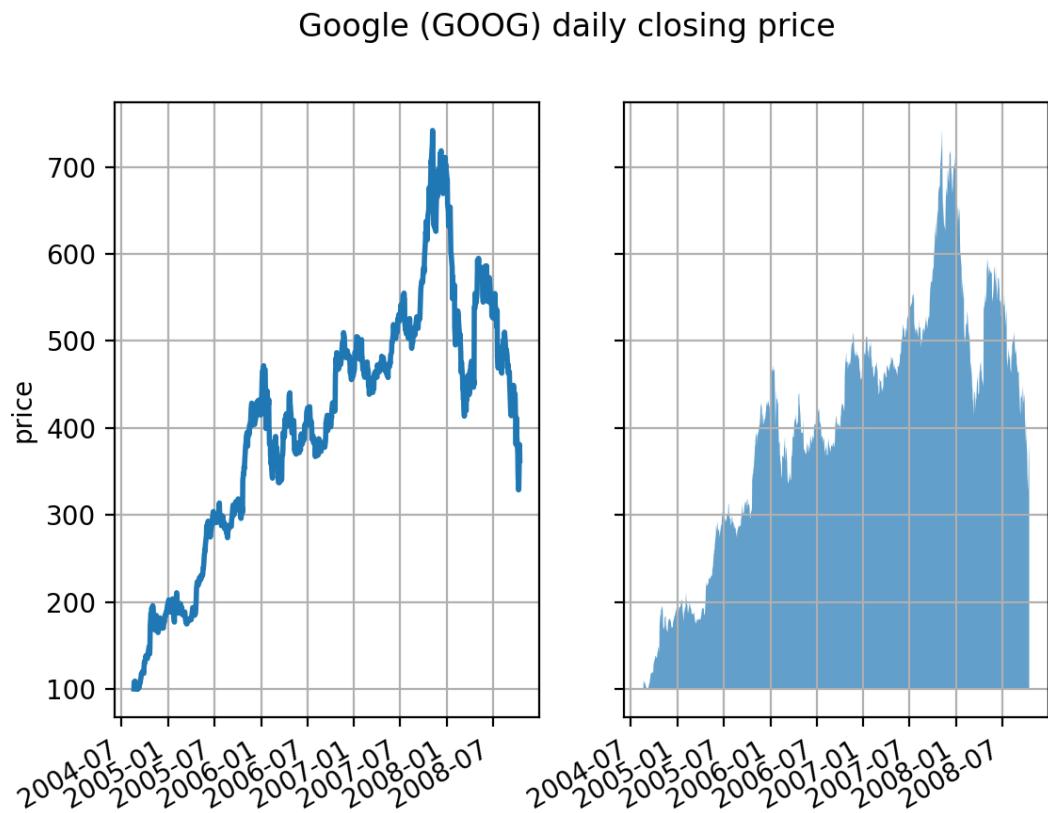
pricemin = r.close.min()

ax1.plot(r.date, r.close, lw=2)
ax2.fill_between(r.date, pricemin, r.close, alpha=0.7)

for ax in ax1, ax2:
    ax.grid(True)
    ax.label_outer()

ax1.set_ylabel('price')

fig.suptitle('Google (GOOG) daily closing price')
fig.autofmt_xdate()
```



圖標示 Figure labels : suptitle, supxlabel, supylabel

```
from matplotlib.cbook import get_sample_data
import matplotlib.pyplot as plt

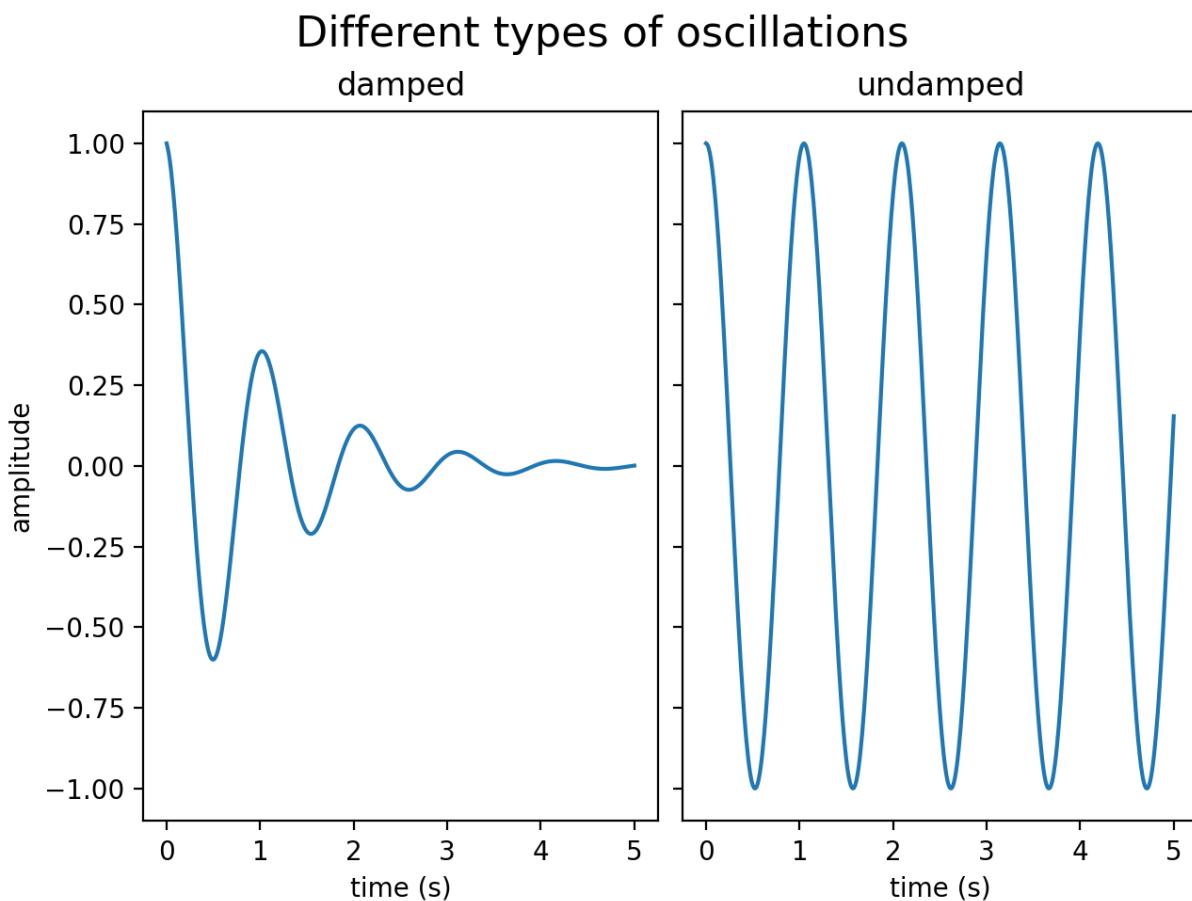
import numpy as np

x = np.linspace(0.0, 5.0, 501)

fig, (ax1, ax2) = plt.subplots(1, 2, constrained_layout=True,
                               sharey=True, figsize=(8, 6))
ax1.plot(x, np.cos(6*x) * np.exp(-x))
ax1.set_title('damped')
ax1.set_xlabel('time (s)')
ax1.set_ylabel('amplitude')

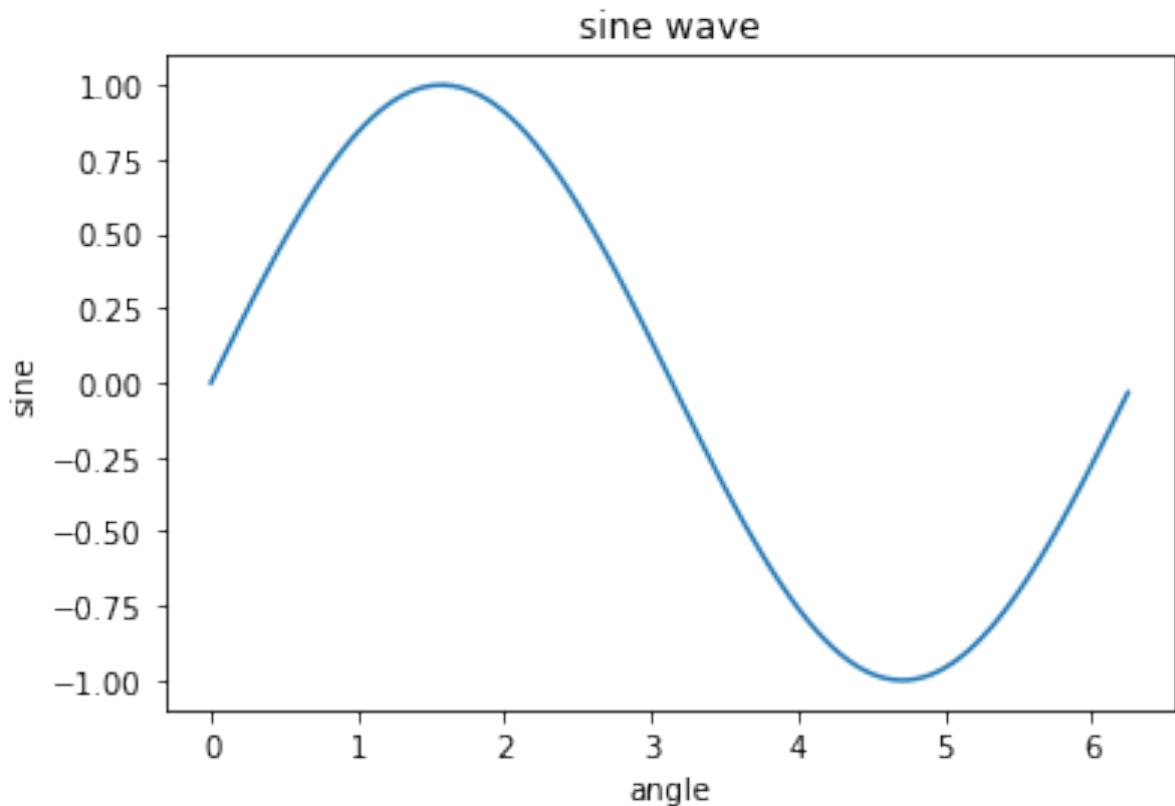
ax2.plot(x, np.cos(6*x))
ax2.set_xlabel('time (s)')
ax2.set_title('undamped')

fig.suptitle('Different types of oscillations', fontsize=16 )
```



繪製正弦波

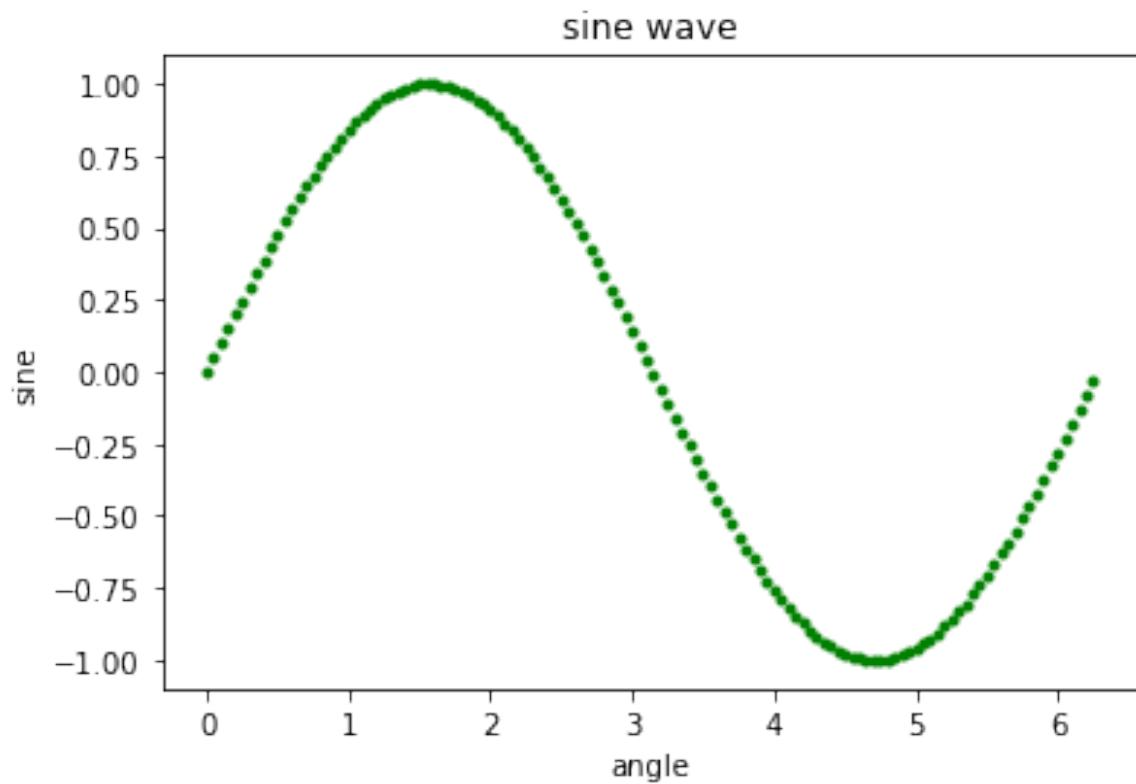
```
import numpy as np
import math #needed for definition of pi
x = np.arange(0, math.pi*2, 0.05)
y = np.sin(x)
plt.plot(x,y)
plt.xlabel("angle")
plt.ylabel("sine")
plt.title('sine wave')
plt.show()
```



若要繪製符號而不是線條，可提供額外的字串參數。

symbols	- , - , - , o , ^ , v , < , > , s , + , x , D , d , 1 , 2 , 3 , 4 , h , H , p , , _
colors	b, g, r, c, m, y, k, w

```
import numpy as np
import math #needed for definition of pi
x = np.arange(0, math.pi*2, 0.05)
y = np.sin(x)
plt.plot(x,y, "g.")    ←
plt.xlabel("angle")
plt.ylabel("sine")
plt.title('sine wave')
plt.show()
```



條形圖

```
import matplotlib.pyplot as plt

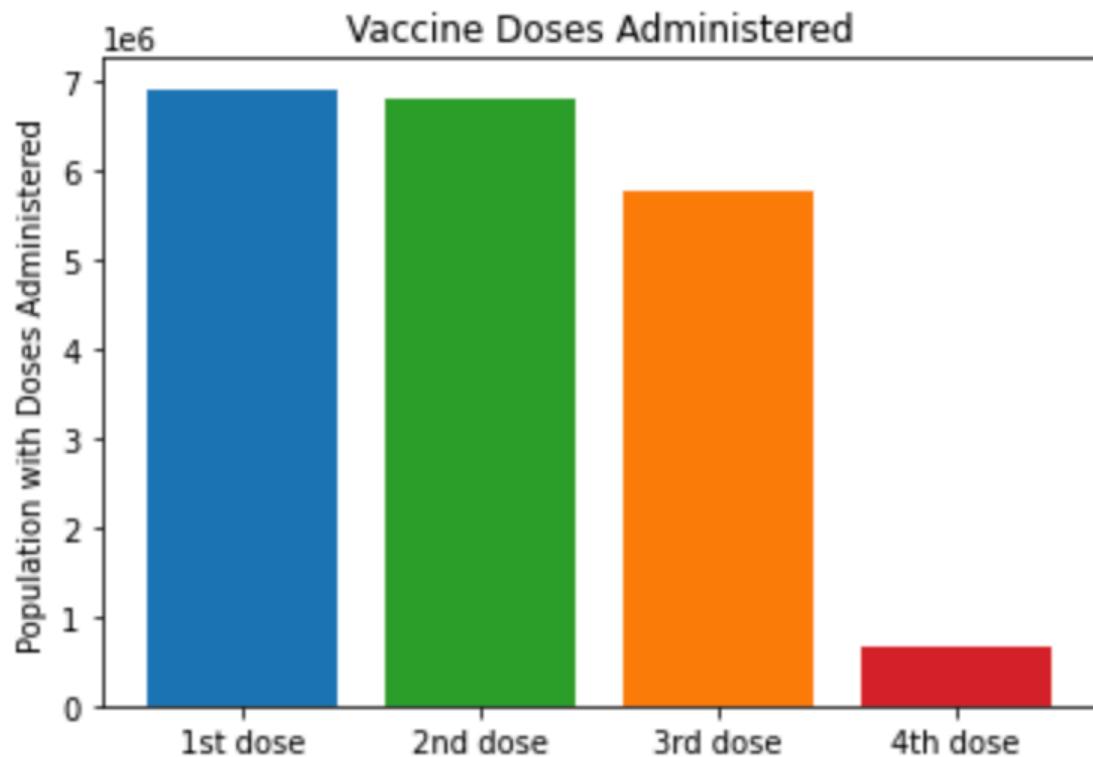
fig, ax = plt.subplots()

vaccine = ['1st dose', '2nd dose', '3rd dose', '4th dose']
counts = [6904243, 6779247, 5764953, 675118]
bar_colors = ['tab:blue', 'tab:green', 'tab:orange', 'tab:red']

ax.bar(vaccine, counts, label=bar_labels, color=bar_colors)

ax.set_ylabel('Population with Doses Administered')
ax.set_title('Vaccine Doses Administered')

plt.show()
```



帶標籤和圖例的分組條形圖

```
import matplotlib.pyplot as plt
import numpy as np

labels = ['Sep', 'Oct', 'Nov', 'Dec']
sales_2021 = [20, 34, 30, 35]
sales_2022 = [25, 32, 34, 20]

x = np.arange(len(labels)) # the label locations
width = 0.35 # the width of the bars

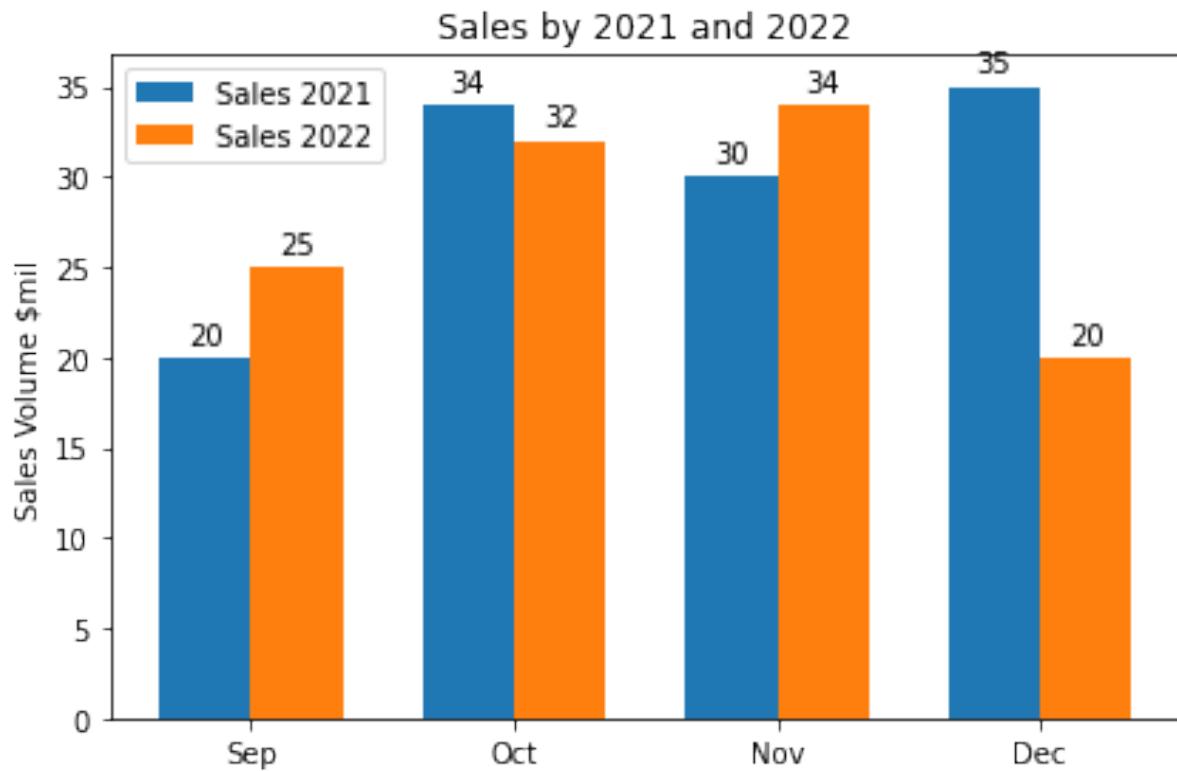
fig, ax = plt.subplots()
rects1 = ax.bar(x - width/2, sales_2021, width, label='Sales 2021')
rects2 = ax.bar(x + width/2, sales_2022, width, label='Sales 2022')

# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('Sales Volume $mil')
ax.set_title('Sales by 2021 and 2022')
ax.set_xticks(x, labels)
ax.legend()

ax.bar_label(rects1, padding=3)
ax.bar_label(rects2, padding=3)

fig.tight_layout()

plt.show()
```



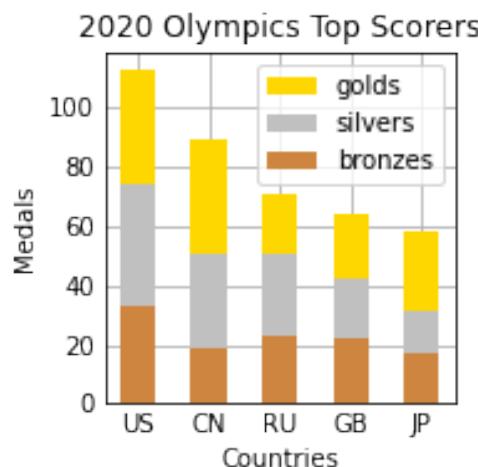
堆積條形圖 Stacked Bar Chart

```
from matplotlib import pyplot as plt
import numpy as np

countries = ['US', 'CN', 'RU', 'GB', 'JP']
bronzes = np.array([33, 19, 23, 22, 17])
silvers = np.array([41, 32, 28, 20, 14])
golds = np.array([39, 38, 20, 22, 27])
ind = [x for x, _ in enumerate(countries)]

plt.bar(ind, golds, width=0.5, label='golds', color='gold', bottom=silvers+bronzes)
plt.bar(ind, silvers, width=0.5, label='silvers', color='silver', bottom=bronzes)
plt.bar(ind, bronzes, width=0.5, label='bronzes', color="#CD853F")

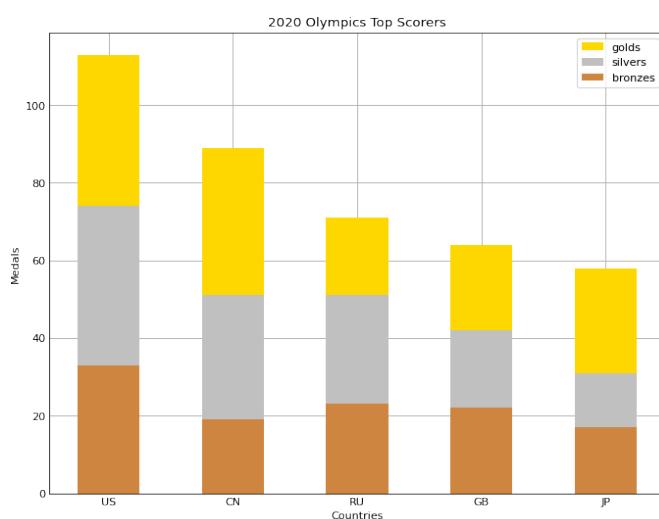
plt.xticks(ind, countries)
plt.ylabel("Medals")
plt.xlabel("Countries")
plt.legend(loc="upper right")
plt.title("2020 Olympics Top Scorers")
```



更改圖形大小和 dpi

- 指定圖形大小和 dpi
- 在代碼頂部添加這兩行

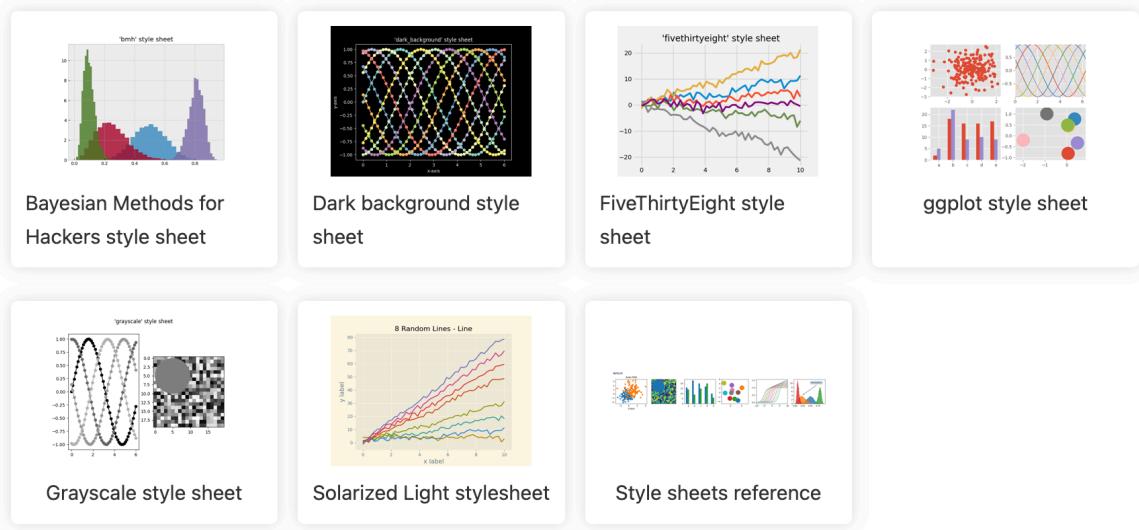
```
from matplotlib.pyplot import figure
figure(figsize=(8, 6), dpi=80)
```



使用預設樣式(preset style)表列印

- 有許多預定義的樣式表在 Matplotlib
- plt.style.use()

Style sheets



```
print(plt.style.available)
```

```
['Solarize_Light2', '_classic_test_patch', '_mpl-gallery', '_mpl-gallery-nogrid', 'bmh', 'classic', 'dark_background', 'fast', 'fivethirtyeight', 'ggplot', 'grayscale', 'seaborn', 'seaborn-bright', 'seaborn-colorblind', 'seaborn-dark', 'seaborn-dark-palette', 'seaborn-darkgrid', 'seaborn-deep', 'seaborn-muted', 'seaborn-notebook', 'seaborn-paper', 'seaborn-pastel', 'seaborn-poster', 'seaborn-talk', 'seaborn-ticks', 'seaborn-white', 'seaborn-whitegrid', 'tableau-colorblind10']
```

Plotting Histogram

```
import matplotlib.pyplot as plt
import numpy as np

plt.style.use('_mpl-gallery')

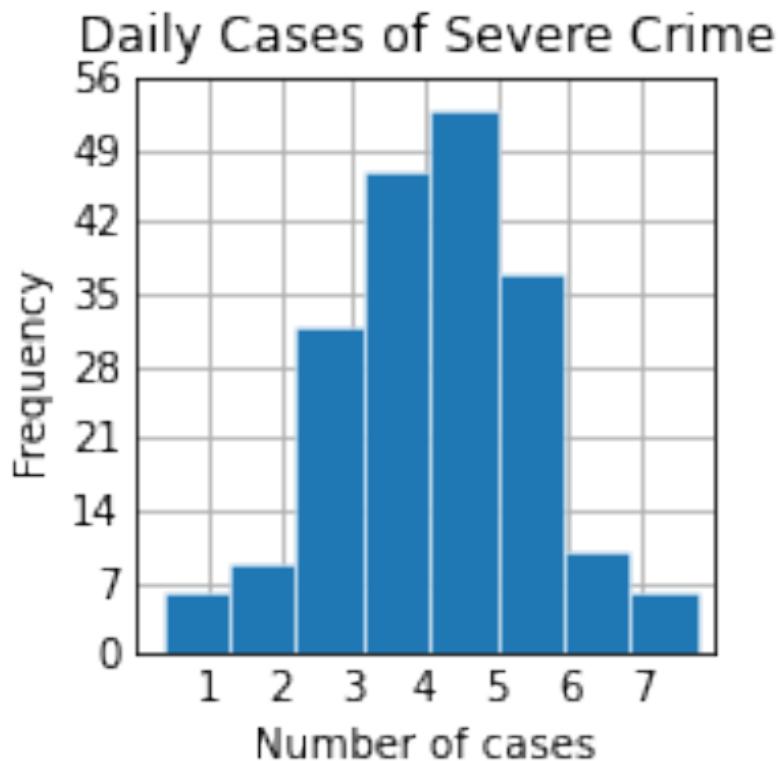
# make data
np.random.seed(1)
x = 4 + np.random.normal(0, 1.5, 200)

# plot:
fig, ax = plt.subplots()

ax.hist(x, bins=8, linewidth=0.5, edgecolor="white")

ax.set(xlim=(0, 8), xticks=np.arange(1, 8),
       ylim=(0, 56), yticks=np.linspace(0, 56, 9))

ax.set_title('Daily Cases of Severe Crime')
ax.set_xlabel('Number of cases')
ax.set_ylabel('Frequency')
plt.show()
```



散點直方圖（可定位軸） Scatter Histogram (Locatable Axes)

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.axes_grid1 import make_axes_locatable

# Fixing random state for reproducibility
np.random.seed(19680801)

# the random data
x = np.random.randn(1000)
y = np.random.randn(1000)

fig, ax = plt.subplots(figsize=(5.5, 5.5))

# the scatter plot:
ax.scatter(x, y)

# Set aspect of the main axes.
ax.set_aspect(1.)

# create new axes on the right and on the top of the current axes
divider = make_axes_locatable(ax)
# below height and pad are in inches
ax_histx = divider.append_axes("top", 1.2, pad=0.1, sharex=ax)
ax_histy = divider.append_axes("right", 1.2, pad=0.1, sharey=ax)

# make some labels invisible
ax_histx.xaxis.set_tick_params(labelbottom=False)
ax_histy.yaxis.set_tick_params(labelleft=False)

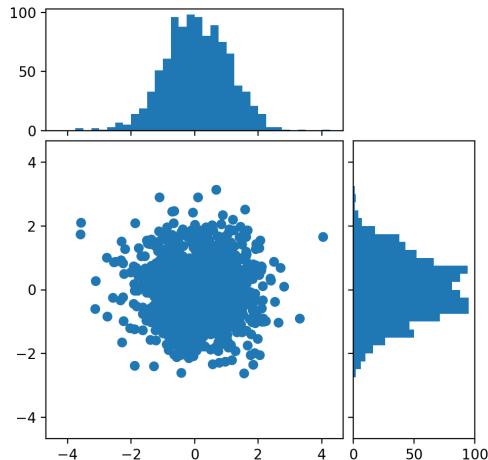
# now determine nice limits by hand:
binwidth = 0.25
xymax = max(np.max(np.abs(x)), np.max(np.abs(y)))
lim = (int(xymax/binwidth) + 1)*binwidth

bins = np.arange(-lim, lim + binwidth, binwidth)
ax_histx.hist(x, bins=bins)
ax_histy.hist(y, bins=bins, orientation='horizontal')

# the xaxis of ax_histx and yaxis of ax_histy are shared with ax,
# thus there is no need to manually adjust the xlim and ylim of these
# axis.

ax_histx.set_yticks([0, 50, 100])
ax_histy.set_xticks([0, 50, 100])

plt.show()
```



Box Plot and Violin Plot

- 除了直方圖，一些數據科學家還會使用 Box Plot or Violin Plot for 揭示一小群數據的分佈
- i.e. 不同年齡組的工資分佈

```
import matplotlib.pyplot as plt
import numpy as np

fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(9, 4))

# Fixing random state for reproducibility
np.random.seed(19680801)

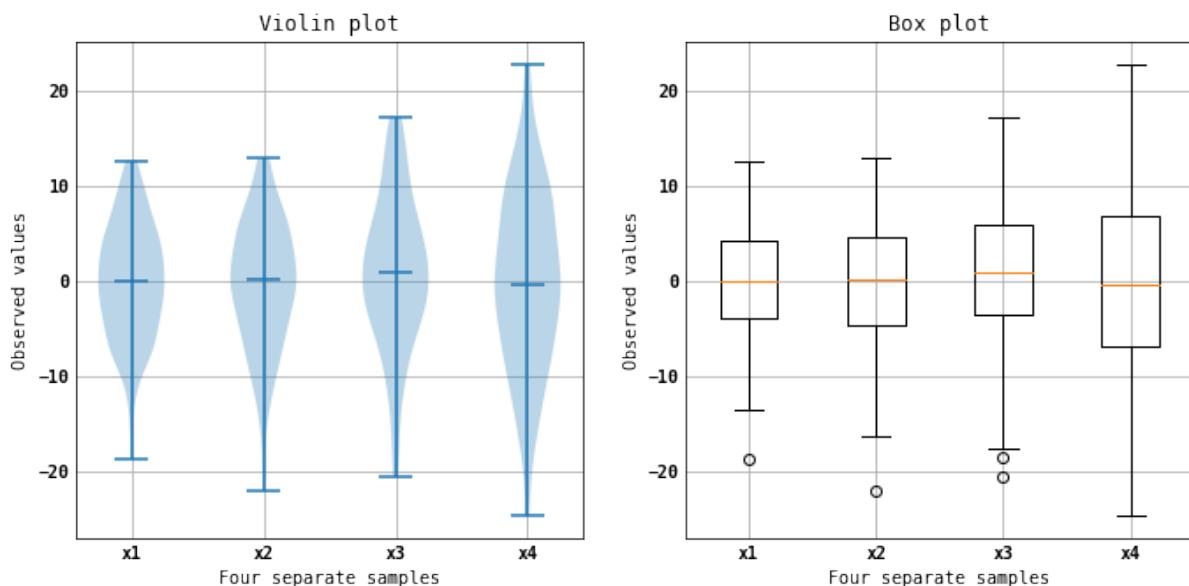
# generate some random test data
all_data = [np.random.normal(0, std, 100) for std in range(6, 10)]

# plot violin plot
axs[0].violinplot(all_data,
                  showmeans=False,
                  showmedians=True)
axs[0].set_title('Violin plot')

# plot box plot
axs[1].boxplot(all_data)
axs[1].set_title('Box plot')

# adding horizontal grid lines
for ax in axs:
    ax.yaxis.grid(True)
    ax.set_xticks([y + 1 for y in range(len(all_data))],
                 labels=['x1', 'x2', 'x3', 'x4'])
    ax.set_xlabel('Four separate samples')
    ax.set_ylabel('Observed values')

plt.show()
```



餅形圖 Pie Chart

```
import numpy as np
import matplotlib.pyplot as plt

fig, ax = plt.subplots(figsize=(8, 6), subplot_kw=dict(aspect="equal"))

recipe = ["375 g flour",
          "75 g sugar",
          "250 g butter",
          "300 g berries"]

data = [float(x.split()[0]) for x in recipe]
ingredients = [x.split()[-1] for x in recipe]

def func(pct, allvals):
    absolute = int(np.round(pct/100.*np.sum(allvals)))
    return "{:.1f}%\n({:d} g)".format(pct, absolute)

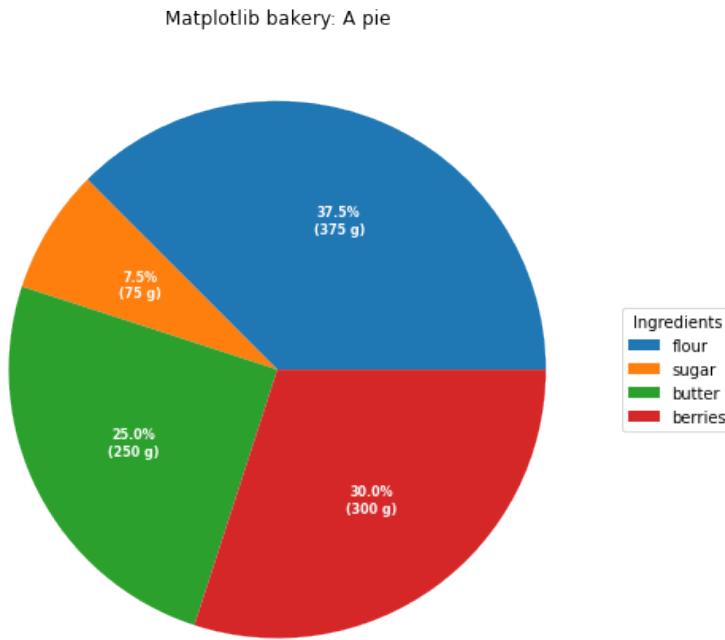
wedges, texts, autotexts = ax.pie(data, autopct=lambda pct: func(pct, data),
                                   textprops=dict(color="w"))

ax.legend(wedges, ingredients,
          title="Ingredients",
          loc="center left",
          bbox_to_anchor=(1, 0, 0.5, 1))

plt.setp(autotexts, size=8, weight="bold")

ax.set_title("Matplotlib bakery: A pie")

plt.show()
```



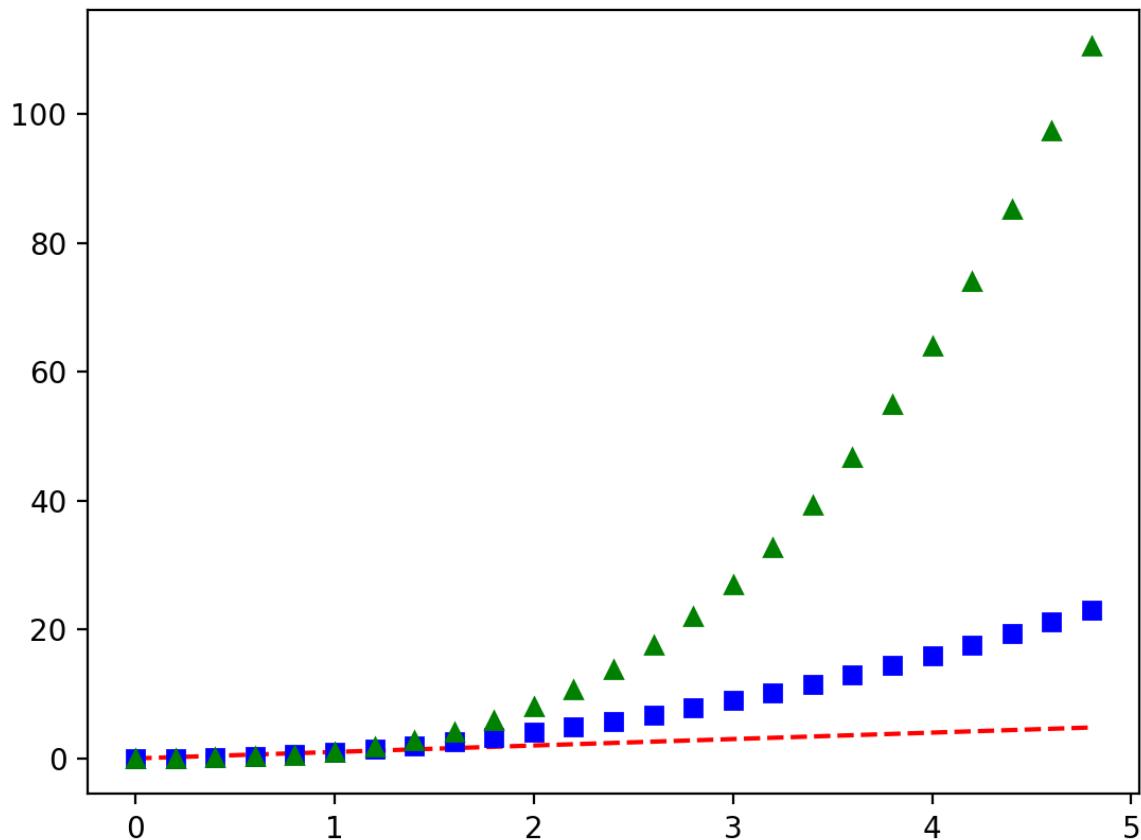
在圖形中以不同顏色繪製 3 條線

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure

figure(figsize=(8, 6), dpi=80)

# evenly sampled time at 200ms intervals
t = np.arange(0., 5., 0.2)

# red dashes, blue squares and green triangles
plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
plt.show()
```



日期刻度標籤

```
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import matplotlib.cbook as cbook

# Load a numpy record array from yahoo csv data with fields date, open, high,
# low, close, volume, adj_close from the mpl-data/sample_data directory. The
# record array stores the date as an np.datetime64 with a day unit ('D') in
# the date column.
data = cbook.get_sample_data('goog.npz', np_load=True)['price_data']

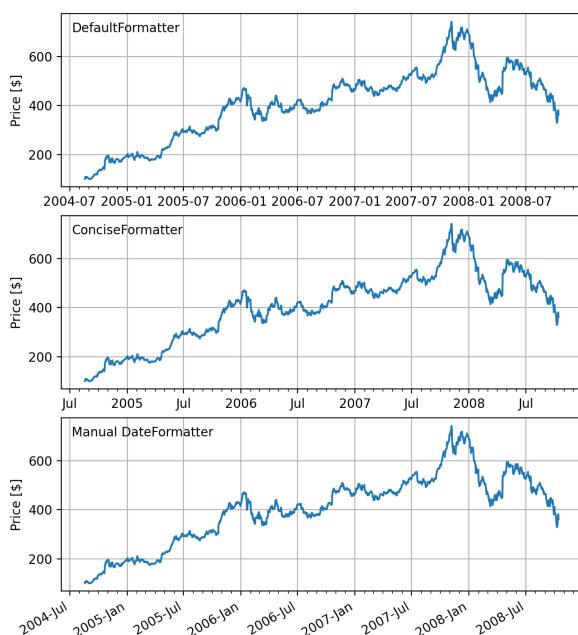
fig, axs = plt.subplots(3, 1, figsize=(6.4, 7), constrained_layout=True)
# common to all three:
for ax in axs:
    ax.plot('date', 'adj_close', data=data)
    # Major ticks every half year, minor ticks every month,
    ax.xaxis.set_major_locator(mdates.MonthLocator(bymonth=(1, 7)))
    ax.xaxis.set_minor_locator(mdates.MonthLocator())
    ax.grid(True)
    ax.set_ylabel(r'Price [\$\']')

# different formats:
ax = axs[0]
ax.set_title('DefaultFormatter', loc='left', y=0.85, x=0.02, fontsize='medium')

ax = axs[1]
ax.set_title('ConciseFormatter', loc='left', y=0.85, x=0.02, fontsize='medium')
ax.xaxis.set_major_formatter(
    mdates.ConciseDateFormatter(ax.xaxis.get_major_locator()))

ax = axs[2]
ax.set_title('Manual DateFormatter', loc='left', y=0.85, x=0.02,
             fontsize='medium')
# Text in the x axis will be displayed in 'YYYY-mm' format.
ax.xaxis.set_major_formatter(mdates.DateFormatter('%Y-%b'))
# Rotates and right-aligns the x labels so they don't crowd each other.
for label in ax.get_xticklabels(which='major'):
    label.set(rotation=30, horizontalalignment='right')

plt.show()
```



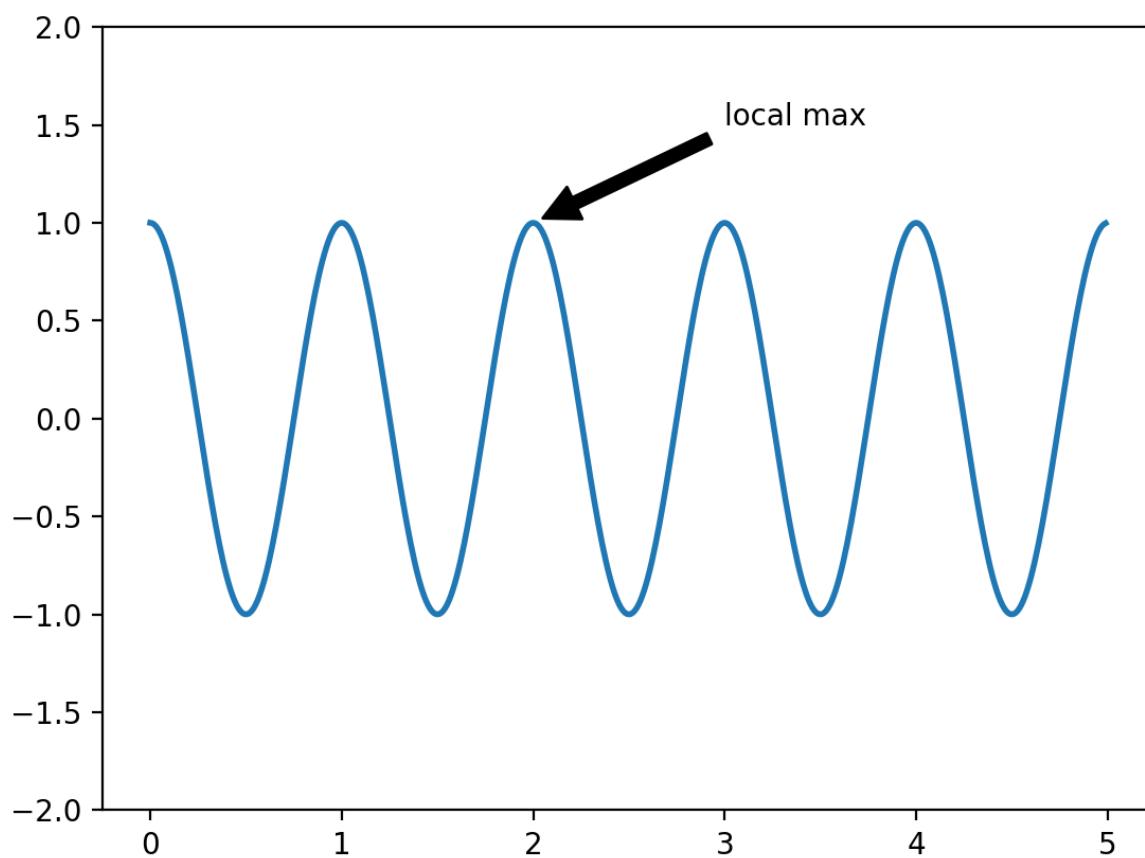
Annotating a plot 圖上加批註

```
import numpy as np
import matplotlib.pyplot as plt

fig, ax = plt.subplots()

t = np.arange(0.0, 5.0, 0.01)
s = np.cos(2*np.pi*t)
line, = ax.plot(t, s, lw=2)

ax.annotate('local max', xy=(2, 1), xytext=(3, 1.5),
            arrowprops=dict(facecolor='black', shrink=0.05),
            )
ax.set_xlim(-2, 2)
plt.show()
```



Text Command (繪製許多不同類型的文本)

```
import matplotlib.pyplot as plt

fig = plt.figure()
fig.suptitle('bold figure suptitle', fontsize=14, fontweight='bold')

ax = fig.add_subplot()
fig.subplots_adjust(top=0.85)
ax.set_title('axes title')

ax.set_xlabel(' xlabel ')
ax.set_ylabel(' ylabel ')

ax.text(3, 8, 'boxed italics text in data coords', style='italic',
       bbox={'facecolor': 'red', 'alpha': 0.5, 'pad': 10})

ax.text(2, 6, r'an equation: $E=mc^2$', fontsize=15)

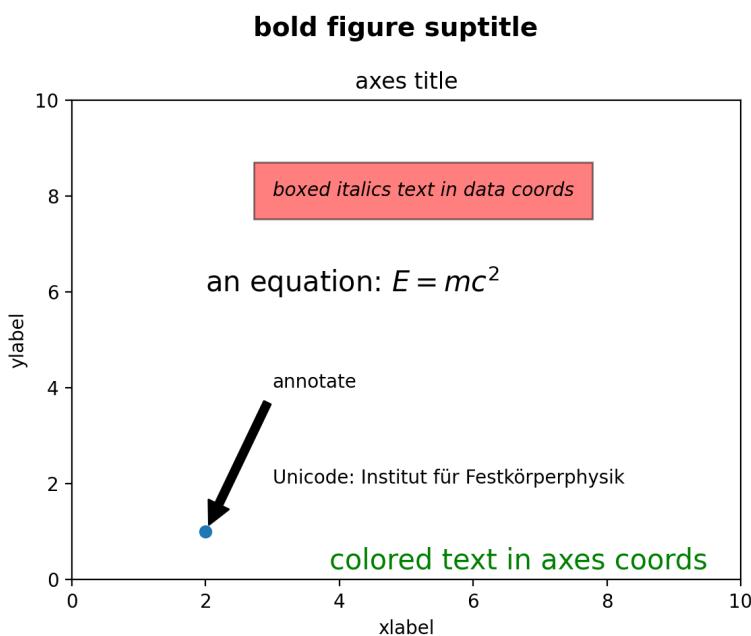
ax.text(3, 2, 'Unicode: Institut f\u00fcr Festk\u00f6rperphysik')

ax.text(0.95, 0.01, 'colored text in axes coords',
       verticalalignment='bottom', horizontalalignment='right',
       transform=ax.transAxes,
       color='green', fontsize=15)

ax.plot([2], [1], 'o')
ax.annotate('annotate', xy=(2, 1), xytext=(3, 4),
            arrowprops=dict(facecolor='black', shrink=0.05))

ax.set(xlim=(0, 10), ylim=(0, 10))

plt.show()
```



Text Watermark 水印

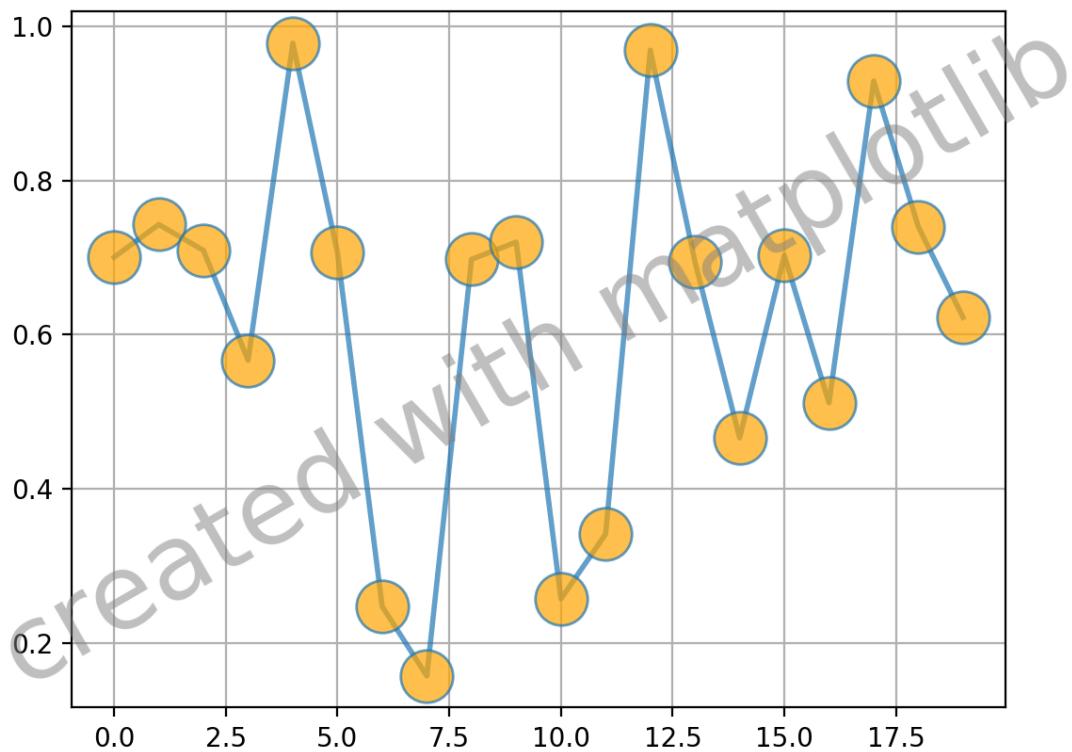
```
import numpy as np
import matplotlib.pyplot as plt

# Fixing random state for reproducibility
np.random.seed(19680801)

fig, ax = plt.subplots()
ax.plot(np.random.rand(20), '-o', ms=20, lw=2, alpha=0.7, mfc='orange')
ax.grid()

ax.text(0.5, 0.5, 'created with matplotlib', transform=ax.transAxes,
        fontsize=40, color='gray', alpha=0.5,
        ha='center', va='center', rotation=30)

plt.show()
```



Simple categorical heatmap 簡單的分類熱圖

```
import numpy as np
import matplotlib
import matplotlib as mpl
import matplotlib.pyplot as plt

vegetables = ["cucumber", "tomato", "lettuce", "asparagus",
              "potato", "wheat", "barley"]
farmers = ["Farmer Joe", "Upland Bros.", "Smith Gardening",
           "Agrifun", "Organiculture", "BioGoods Ltd.", "Cornylee Corp."]

harvest = np.array([[0.8, 2.4, 2.5, 3.9, 0.0, 4.0, 0.0],
                   [2.4, 0.0, 4.0, 1.0, 2.7, 0.0, 0.0],
                   [1.1, 2.4, 0.8, 4.3, 1.9, 4.4, 0.0],
                   [0.6, 0.0, 0.3, 0.0, 3.1, 0.0, 0.0],
                   [0.7, 1.7, 0.6, 2.6, 2.2, 6.2, 0.0],
                   [1.3, 1.2, 0.0, 0.0, 0.0, 3.2, 5.1],
                   [0.1, 2.0, 0.0, 1.4, 0.0, 1.9, 6.3]])

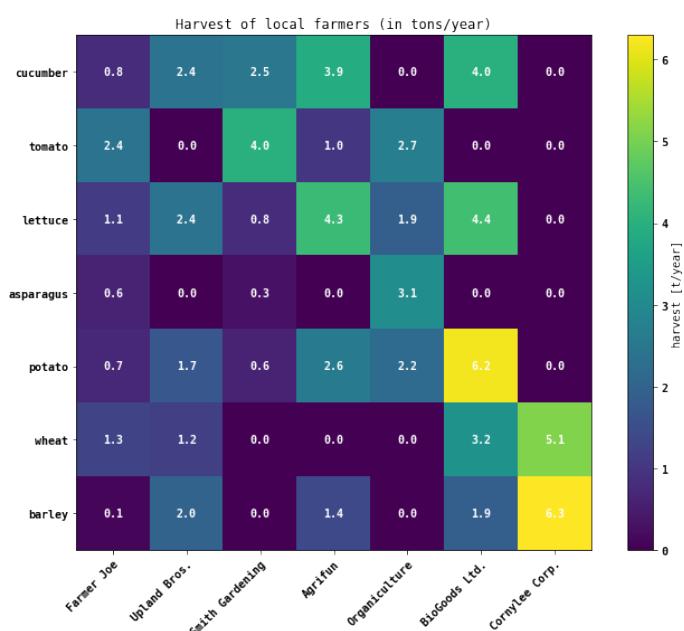
fig, ax = plt.subplots(figsize=(10, 8))
im = ax.imshow(harvest)

# Show all ticks and label them with the respective list entries
ax.set_xticks(np.arange(len(farmers)), labels=farmers)
ax.set_yticks(np.arange(len(vegetables)), labels=vegetables)

# Rotate the tick labels and set their alignment.
plt.setp(ax.get_xticklabels(), rotation=45, ha="right",
         rotation_mode="anchor")

# Loop over data dimensions and create text annotations.
for i in range(len(vegetables)):
    for j in range(len(farmers)):
        text = ax.text(j, i, harvest[i, j],
                      ha="center", va="center", color="w")

ax.set_title("Harvest of local farmers (in tons/year)")
fig.colorbar(im, ax=ax, label='harvest [t/year]')
fig.tight_layout()
plt.show()
```



Toolkits

- Toolkits 是特定於應用程式的函數的集合
- 擴展 Matplotlib
- 通用功能
 - `mpl_toolkits.mplot3d`
 - `mpl_toolkits.axes_grid1`
 - `mpl_toolkits.axisartist`

3D Plotting

```
from mpl_toolkits.mplot3d import axes3d
import matplotlib.pyplot as plt

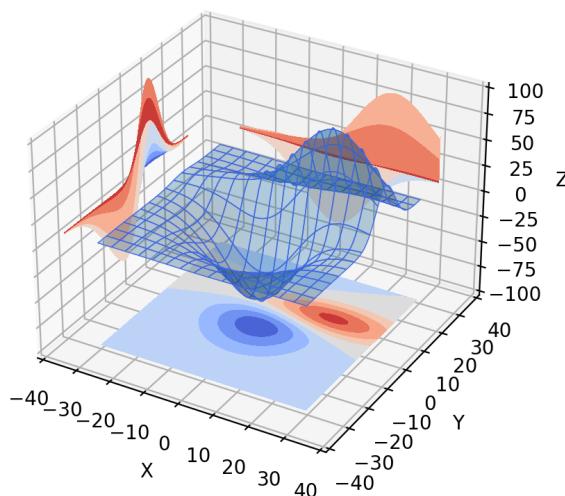
ax = plt.figure(figsize=(8,8)).add_subplot(projection='3d')
X, Y, Z = axes3d.get_test_data(0.05)

# Plot the 3D surface
ax.plot_surface(X, Y, Z, edgecolor='royalblue', lw=0.5, rstride=8, cstride=8,
                alpha=0.3)

# Plot projections of the contours for each dimension. By choosing offsets
# that match the appropriate axes limits, the projected contours will sit on
# the 'walls' of the graph
ax.contourf(X, Y, Z, zdir='z', offset=-100, cmap='coolwarm')
ax.contourf(X, Y, Z, zdir='x', offset=-40, cmap='coolwarm')
ax.contourf(X, Y, Z, zdir='y', offset=40, cmap='coolwarm')

ax.set(xlim=(-40, 40), ylim=(-40, 40), zlim=(-100, 100),
       xlabel='X', ylabel='Y', zlabel='Z')

plt.show()
```



總結

- Matplotlib 是一個大型可視化庫，應對多種科學領域
- 它正在由專業人士發展和擴展功能
- 在其網站上查找更多繪圖、資訊和文檔

matplotlib.org



Appendix

Types of Plot		Axis Functions		Figures Function	
	Function & Description		Function & Description		Function & Description
1	Bar Make a bar plot.	1	Axes Add axes to the figure.	1	Figtext Add text to figure.
2	Barh Make a horizontal bar plot.	2	Text Add text to the axes.	2	Figure Creates a new figure.
3	Boxplot Make a box and whisker plot.	3	Title Set a title of the current axes.	3	Show Display a figure.
4	Hist Plot a histogram.	4	Xlabel Set the x axis label of the current axis.	4	Savefig Save the current figure.
5	hist2d Make a 2D histogram plot.	5	Xlim Get or set the x limits of the current axes.	5	Close Close a figure window.
6	Pie Plot a pie chart.	6	Xscale .		
7	Plot Plot lines and/or markers to the Axes.	7	Xticks Get or set the x-limits of the current tick locations and labels.		
8	Polar Make a polar plot..	8	Ylabel Set the y axis label of the current axis.		
9	Scatter Make a scatter plot of x vs y.	9	Ylim Get or set the y-limits of the current axes.		
10	Stackplot Draws a stacked area plot.	10	Yscale Set the scaling of the y-axis.		
11	Stem Create a stem plot.	11	Yticks Get or set the y-limits of the current tick locations and labels.		
12	Step Make a step plot.				
13	Quiver Plot a 2-D field of arrows.				

Color Code	
Character	Colour
'b'	Blue
'g'	Green
'r'	Red
'b'	Blue
'c'	Cyan
'm'	Magenta
'y'	Yellow
'k'	Black
'b'	Blue
'w'	White

Line Style	
Character	Description
'—'	Solid line
'—'	Dashed line
'-. '	Dash-dot line
'.'	Dotted line
'H'	Hexagon marker

Marker Code	
Character	Description
'.'	Point marker
'o'	Circle marker
'x'	X marker
'D'	Diamond marker
'H'	Hexagon marker
's'	Square marker
'+'	Plus marker