

Matplotlib 绘图基本步骤与示例



```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4]          # Step 1 准备数据
y = [10, 20, 25, 30]

fig = plt.figure()        # Step 2 创建图形
ax = fig.add_subplot(111)  # Step 3 绘图
ax.plot(x, y, color='lightblue', \
        linewidth=3)       # Step 4 自定义设置
ax.scatter([2, 4, 6], [5, 15, 25], \
          color='darkgreen', marker='^')
ax.set_xlim(1, 6.5)
plt.savefig('foo.png')    # Step 5 保存图形
plt.show()                # Step 6 显示图形
```

1. 准备数据

1.1 一维数据

```
import numpy as np

x = np.linspace(0, 10, 100)
y = np.cos(x)
z = np.sin(x)
```

1.2 二维数据或图片

```
data = 2 * np.random.random((10, 10))
data2 = 3 * np.random.random((10, 10))
y, X = np.mgrid[-3:3:100j, -3:3:100j]
U = -1 - X**2 + Y
V = 1 + X - Y**2

from matplotlib.cbook import get_sample_data
img = np.load(get_sample_data('axes_grid/bivariate_normal.npy'))
```

2. 绘制图形

2.1 画布

```
import matplotlib.pyplot as plt # 导入库
fig = plt.figure()
fig2 = plt.figure(figsize=plt.figaspect(2.0))
```

2.2 坐标轴

图形是以坐标轴为核心绘制的，大多数情况下子图就可以满足需求。
子图是栅格系统的坐标轴。

```
fig.add_axes()
ax1 = fig.add_subplot(221)      # row-col-num
ax3 = fig.add_subplot(212)
fig3, axes = plt.subplots(nrows=2, ncols=2)
fig4, axes2 = plt.subplots(ncols=3)
```

3. 绘图详解

3.1 一维数据

```
fig, ax = plt.subplots()
lines = ax.plot(x, y) # 用线或标记连接点
ax.scatter(x, y)      # 缩放或着色未连接的点
axes[0, 0].bar([1, 2, 3], [3, 4, 5]) # 绘制柱状图
axes[1, 0].barh([0.5, 1, 2.5], [0, 1, 2]) # 绘制水平柱状图
axes[1, 1].axhline(0.45) # 绘制与轴平行的横线
axes[0, 1].axvline(0.65) # 绘制与轴垂直的竖线
ax.fill(x, y, color='blue') # 绘制填充多边形
ax.fill_between(x, y, color='yellow') # 填充 y 值和 0 之间
```

3.2 Vector Fields – 向量场

```
axes[0, 1].arrow(0, 0, 0.5, 0.5) # 为坐标轴添加箭头
axes[1, 1].quiver(y, z)          # 二维箭头
axes[0, 1].streamplot(X, Y, U, V) # 二维箭头
```

3.3 Data Distributions – 数据分布

```
ax1.hist(y)          # 直方图
ax3.boxplot(y)       # 箱形图
ax3.violinplot(z)    # 小提琴图
```

3.4 二维数据或图片

```
fig, ax = plt.subplots()
im = ax.imshow(img, cmap='gist_earth', \
               interpolation='nearest', vmin=-2, vmax=2)
axes2[0].pcolor(data2) # 二维数组伪彩色图
axes2[0].pcolormesh(data) # 二维数组等高线伪彩色图
cS = plt.contour(Y, X, U) # 等高线图
axes2[2].contourf(data1) # 等高线轮廓图
axes2[2].clabel(cS) # 等高线图标签
```

4. 自定义图形

4.1 颜色、色条与色彩表

```
plt.plot(x, x, x, x**2, x, x**3)
ax.plot(x, y, alpha=0.4)
ax.plot(x, y, c='k')
fig.colorbar(im, orientation='horizontal')
im = ax.imshow(img, cmap='seismic')
```

4.2 标记

```
fig, ax = plt.subplots()
ax.scatter(x, y, marker=".")
ax.plot(x, y, marker="o")
```

4.3 线型

```
plt.plot(x, y, linewidth=4.0)
plt.plot(x, y, ls='solid')
plt.plot(x, y, ls='--')
plt.plot(x, y, '--', x2, y2, '-.')
plt.setp(lines, color='r', linewidth=4.0)
```

4.4 文本与标注

```
ax.text(1, -2.1, 'Example Graph', style='italic')
ax.annotate("Sine", xy=(8, 0), \
            xycoords='data', \
            xytext=(10.5, 0), \
            textcoords='data', \
            arrowprops=dict(arrowstyle="->", \
                            connectionstyle="arc3"), )
```

4.5 数学符号

```
plt.title(r'$\sigma_i=15$', fontsize=20)
```

4.6 尺寸限制、图例和布局

尺寸限制与自动调整

```
ax.margins(x=0.0, y=0.1) # 添加内边距
ax.axis('equal') # 将图形纵横比设置为 1
ax.set(xlim=[0, 10.5], ylim=[-1.5, 1.5]) # 设置 x 轴与 y 轴的限制
ax.set_xlim(0, 10.5) # 设置 x 轴的限制
```

图例

```
ax.set(title='An Example Axes', ylabel='Y-Axis', xlabel='X-Axis') # 设置标题与 x、y 轴的标签
ax.legend(loc='best') # 自动选择最佳的图例位置
```

标记

```
ax.xaxis.set(ticks=range(1, 5), ticklabels=[3, 100, -12, "foo"]) # 手动设置 X 轴刻度
ax.tick_params(axis='y', direction='inout', length=10) # 设置 Y 轴长度与方向
```

子图间距

```
fig3.subplots_adjust(wspace=0.5, hspace=0.3, left=0.125, right=0.9, top=0.9, bottom=0.1) # 调整子图间距
fig.tight_layout() # 设置画布的子图布局
```

坐标轴边线

```
ax1.spines['top'].set_visible(False) # 隐藏顶部坐标轴线
ax1.spines['bottom'].set_position(('outward', 10)) # 设置底部边线的位置为 outward
```

5. 保存

savefig 函数

```
plt.savefig('foo.png') # 保存画布
plt.savefig('foo.png', transparent=True) # 透明画布
```

6. 显示图形

show 函数
`plt.show()`

7. 关闭与清除

绘图清除与关闭
`plt.cla()` # 清除坐标轴
`plt.clf()` # 清除画布
`plt.close()` # 关闭窗口



Matplotlib 是使用 Python 进行数据可视化最重要的基础工具库，Matplotlib 可以很好地和 Numpy 和 Pandas 进行兼容，方便在数据计算和探索分析的同时，以图示的方式呈现信息。



扫码回复“数据科学”
下载最新全套速查表

Matplotlib 速查表

获取最新版 | <http://www.showmeai.tech/>

作者 | 韩信子 @ShowMeAI

设计 | 南乔 @ShowMeAI

参考 | DataCamp Cheatsheet

数据科学工具库速查表



NumPy 是 Python 数据科学计算的核心库，提供了高性能多维数组对象及处理数组的工具。使用以下语句导入 NumPy 库：

```
import numpy as np
```



SciPy 是基于 NumPy 创建的 Python 科学计算核心库，提供了众多数学算法与函数。



Pandas 是基于 NumPy 创建的 Python 库，为 Python 提供了易于使用的的数据结构和数据分析工具。使用以下语句导入：

```
import pandas as pd
```



Matplotlib 是 Python 的二维绘图库，用于生成符合出版质量或跨平台交互环境的各类图形。

```
import matplotlib.pyplot as plt
```



Seaborn 是基于 matplotlib 开发的高阶 Python 数据可视图库，用于绘制优雅、美观的统计图形。使用下列别名导入该库：

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```



Bokeh 是 Python 的交互式可视图库，用于生成在浏览器里显示的大规模数据集高性能可视图。Bokeh 的中间层通用 **bokeh.plotting** 界面主要为两个组件：数据与图示例。

```
from bokeh.plotting import figure
```

```
from bokeh.io import output_file, show
```

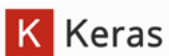


PySpark 是 Spark 的 Python API，允许 Python 调用 Spark 编程模型。Spark SQL 是 Apache Spark 处理结构化数据模块。

AI 垂直领域工具库速查表



Scikit-learn 是开源的 Python 库，通过统一的界面实现机器学习、预处理、交叉验证及可视化算法。



Keras 是强大、易用的深度学习库，基于 Theano 和 TensorFlow 提供了高阶神经网络 API，用于开发和评估深度学习模型。



“TensorFlow™ is an open source software library for numerical computation using data flow graphs.” **TensorFlow** 是 Google 公司开发的机器学习架构，兼顾灵活性和扩展性，既适合用于工业生产也适合用于科学研究。



PyTorch 是 Facebook 团队 2017 年初发布的深度学习框架，有利于研究人员、爱好者、小规模项目等快速搞出原型。**PyTorch** 也是 Python 程序员最容易上手的深度学习框架。



Hugging Face 以开源的 NLP 预训练模型库 **Transformers** 而广为人知，目前 GitHub Star 已超过 54000+。**Transformers** 提供 100+ 种语言的 32 种预训练语言模型，简单，强大，高性能，是新手入门的不二选择。



OpenCV 是一个跨平台计算机视觉库，由 C 函数 /C++ 类构成，提供了 Python、MATLAB 等语言的接口。**OpenCV** 实现了图像处理和计算机视觉领域的很多通用算法。

编程语言速查表



SQL 是管理关系数据库的结构化查询语言，包括数据的增删查改等。作为数据分析的必备技能、岗位 JD 的重要关键词，SQL 是技术及相关岗位同学一定要掌握的语言。



Python 编程语言简洁快速、入门简单且功能强大，拥有丰富的第三方库，已经成为大数据和人工智能领域的主流编程语言。

More...

AI 知识技能速查表



Jupyter Notebook 交互式计算环境，支持运行 40+ 种编程语言，可以用来编写漂亮的交互式文档。这个教程把常用的基础功能讲解得很清楚，对新手非常友好。



正则表达式 非常强大，能匹配很多规则的文本，常用于文本提取和爬虫处理。这也是一门令人难以捉摸的语言，字母、数字和符号堆在一起，像极了“火星文”。

More...



ShowMeAI 速查表 (©2021)

获取最新版 | <http://www.showmeai.tech/>

作者 | 韩信子 @ShowMeAI

设计 | 南乔 @ShowMeAI

数据科学工具库速查表

扫码回复“数据科学”

获取最新全套速查表

AI 垂直领域工具库速查表

扫码回复“工具库”

获取最新全套速查表

编程语言速查表

扫码回复“编程语言”

获取最新全套速查表

AI 知识技能速查表

扫码回复“知识技能”

获取最新全套速查表