




Python 数据科学 速查表

Matplotlib

Matplotlib

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。



1 准备数据

参阅 列表与 NumPy

一维数据

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

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

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

画布

```
>>> fig = plt.figure()
>>> fig2 = plt.figure(figsize=plt.figaspect(2.0))
```

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 绘图例程

一维数据

```
>>> fig, ax = plt.subplots()
>>> lines = ax.plot(x,y)
>>> ax.scatter(x,y)
>>> axes[0,0].bar([1,2,3],[3,4,5])
>>> axes[1,0].barh([0.5,1,2.5],[0,1,2])
>>> axes[1,1].axhline(0.45)
>>> axes[0,1].axvline(0.65)
>>> ax.fill(x,y,color='blue')
>>> ax.fill_between(x,y,color='yellow')
```

用线或标记连接点
缩放或着色未连接的点
绘制等宽纵向矩形
绘制等高纵向矩形
绘制与轴平行的横线
绘制与轴垂直的竖线
绘制填充多边形
填充y值和0之间

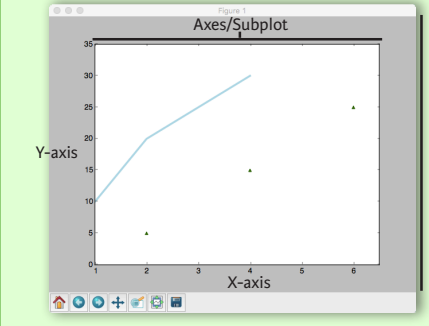
二维数据或图片

```
>>> fig, ax = plt.subplots()
>>> im = ax.imshow(img,
                  cmap='gist_earth',
                  interpolation='nearest',
                  vmin=-2,
                  vmax=2)
```

色彩表或RGB数组

图形解析与 workflow

图形解析



workflow

Matplotlib 绘图的基本步骤:

- 1 准备数据
- 2 创建图形
- 3 绘图
- 4 自定义设置
- 5 保存图形
- 6 显示图形

```
>>> import matplotlib.pyplot as plt
>>> x = [1,2,3,4]
>>> y = [10,20,25,30]
>>> fig = plt.figure()
>>> ax = fig.add_subplot(111)
>>> ax.plot(x, y, color='lightblue', linewidth=3)
>>> ax.scatter([2,4,6],
              [5,15,25],
              color='darkgreen',
              marker='^')
>>> ax.set_xlim(1, 6.5)
>>> plt.savefig('foo.png')
>>> plt.show()
```

4 自定义图形

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')
```

标记

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

线型

```
>>> plt.plot(x,y,linewidth=4.0)
>>> plt.plot(x,y,ls='solid')
>>> plt.plot(x,y,ls='--')
>>> plt.plot(x,y,'--',x**2,y**2,'-.')
>>> plt.setp(lines,color='r',linewidth=4.0)
```

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"),)
```

数学符号

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

尺寸限制、图例和布局

```
>>> ax.margins(x=0.0,y=0.1)
>>> ax.axis('equal')
>>> ax.set(xlim=[0,10.5],ylim=[-1.5,1.5])
>>> ax.set_xlim(0,10.5)
```

```
>>> ax.set(title='An Example Axes',
          ylabel='Y-Axis',
          xlabel='X-Axis')
>>> ax.legend(loc='best')
```

```
>>> ax.xaxis.set(ticks=range(1,5),
               ticklabels=[3,100,-12,"foo"])
>>> ax.tick_params(axis='y',
                  direction='inout',
                  length=10)
```

```
>>> fig3.subplots_adjust(wspace=0.5,
                       hspace=0.3,
                       left=0.125,
                       right=0.9,
                       top=0.9,
                       bottom=0.1)
>>> fig.tight_layout()
```

```
>>> ax1.spines['top'].set_visible(False)
>>> ax1.spines['bottom'].set_position(('outward',10))
```

添加内边距
将图形纵横比设置为1
设置x轴与y轴的限制
设置x轴的限制

设置标题与x、y轴的标签

自动选择最佳的图例位置

手动设置x轴刻度

设置y轴长度与方向

调整子图间距

设置画布的子图布局

隐藏顶部坐标轴
设置底部坐标轴的位置为outward

5 保存

```
>>> plt.savefig('foo.png')
>>> plt.savefig('foo.png', transparent=True)
```

6 显示图形

```
>>> plt.show()
```

关闭与清除

```
>>> plt.cla()
>>> plt.clf()
>>> plt.close()
```

清除坐标轴
清除画布
关闭窗口