python绘图篇

- 1, xlable,ylable设置x, y轴的标题文字。
- 2, title设置标题。
- 3, xlim,ylim设置x,y轴显示范围。

plt.show()显示绘图窗口,通常情况下,show () 会阻碍程序运行,带-wthread等参数的环境下,窗口不会关闭。plt.saveFig () 保存图像。

面向对象绘图

1, 当前图表和子图可以用gcf(),gca()获得。

subplot()绘制包含多个图表的子图。 configure subplots,可调节子图与图表边框距离。

可以通过修改配置文件更改对象属性。

图标显示中文

- 1,在程序中直接指定字体。
- 2, 在程序开始修改配置字典reParams.
- 3, 修改配置文件。

Artist对象

- 1, 图标的绘制领域。
- 2,如何在FigureCanvas对象上绘图。
- 3,如何使用Renderer在FigureCanvas对象上绘图。

FigureCanvas和Render处理底层图像操作, Artist处理高层结构。

分为简单对象和容器对象,简单的Aritist是标准的绘图元件,例如Line 2D,Rectangle,Text,AxesImage等,而容器类型包含许多简单的的Aritist对象,使他们构成一个整体,例如Axis,Axes,Figure等。

直接创建Artist对象进项绘图操作步奏:

- 1, 创建Figure对象(通过figure () 函数, 会进行许多初始化操作, 不建议直接创建。)
- 2,为Figure对象创建一个或多个Axes对象。
- 3,调用Axes对象的方法创建各类简单的Artist对象。

5.2.1 Artist 对象的属性

通过前面的介绍我们已经知道,图表中的每个绘图元素都用一个 Artist 对象来表示,而每个 Artist 对象都有一大堆属性用来控制其显示效果。例如 Figure 对象和 Axes 对象都有 patch 属性作为其背景,它是一个 Rectangle 对象。通过设置它的属性可以修改图表的背景颜色或透明度等属性,下面的例子将图表的背景颜色设置为绿色:

>>> fig = plt.figure()

>>> fig.show()

>>> fig.patch.set_color("g") # 设置背景颜色为绿色

>>> fig.canvas.draw() # 重绘界面

注意:在调用 set_color()设置好背景颜色之后,设置的背景颜色并不会立即在界面上显示出来,还需要调用 fig.canvas.draw()才能更新界面显示。

下面是所有 Artist 对象都拥有的一些属性:

- alpha: 透明度, 值在0到1之间, 0为完全透明, 1为完全不透明。
- animated: 布尔值, 在绘制动画效果时使用。
- axes: 拥有此 Artist 对象的 Axes 对象, 可能为 None。
- clip_box: 对象的裁剪框。
- clip_on: 是否裁剪。
- clip_path: 裁剪的路径。
- contains: 判断指定点是否在对象之上的函数。
- figure: 拥有此 Artist 对象的 Figure 对象,可能为 None。
- label: 文本标签。
- picker: 控制 Artist 对象的选取。
- transform: 控制偏移、旋转、缩放等坐标变换。
- visible: 是否可见。
- zorder: 控制绘图顺序。

Artist 对象的所有属性都可以通过相应的 get_*()和 set_*()方法进行读写,例如下面的语句将新绘制的曲线对象的 alpha 属性设置为 0.5,从而使它变成半透明:

- >>> line = plt.plot([1,2,3,2,1], lw=4)[0]
- >>> line.set_alpha(0.5)

可以使用 set()一次设置多个属性:

>>> line.set(alpha=0.5, zorder=2)

使用前面介绍的 getp()可以方便地输出 Artist 对象的所有属性名及其对应值:

>>> plt.getp(fig.patch)

Figure容器

如何找到指定的Artist对象。

- 1, 可调用add subplot()和add axes()方法向图表添加子图。
- 2, 可使用for循环添加栅格。
- 3,可通过transform修改坐标原点。

Axes容器

- 1, patch修改背景。
- 2, 包含坐标轴, 坐标网格, 刻度标签, 坐标轴标题等内容。
- 3, get ticklabels(),,get-ticklines获得刻度标签和刻度线。
- 1,可对曲线进行插值。
- 2, fill between()绘制交点。
- 3, 坐标变换。
- 4,绘制阴影。
- 5,添加注释。
- 1, 绘制直方图的函数是

hist(x,...)

2,箱线图(Boxplot)也称箱须图(Box-whisker Plot),是利用数据中的五个统计量:最小值、第一四分位数、中位数、第三四分位数与最大值来描述数据的一种方法,它可以粗略地看出数据是否具有对称性以及分

布的分散程度等信息,特别可以用于对几个样本的比较。

3, 饼图就是把一个圆盘按所需表达变量的观察数划分为若干份,每一份的角度(即面积)等价于每个观察值的大小。

groupsizes=c()

labels=()

pie(groupsizes, labels)

- 4.散点图
- 5,QQ图

低层绘图函数

类似于barplot(),dotchart()和plot()这样的函数采用低层的绘图函数来画线和点,来表达它们在页面上放置的位置以及其他各种特征。

在这一节中,我们会描述一些低层的绘图函数,用户也可以调用这些函数用于绘图。首先我们先讲一下R怎么描述一个页面;然后我们讲怎么在页面上添加点,线和文字;最后讲一下怎么修改一些基本的图形。

绘图区域与边界

R在绘图时,将显示区域划分为几个部分。绘制区域显示了根据数据描绘出来的图像,在此区域内R根据数据选择一个坐标系,通过显示出来的坐标轴可以看到R使用的坐标系。在绘制区域之外是边沿区,从底部开始按顺时针方向分别用数字1到4表示。文字和标签通常显示在边沿区域内,按照从内到外的行数先后显示。

添加对象

在绘制的图像上还可以继续添加若干对象,下面是几个有用的函数,以及对其功能的说

明。

- •points(x, y, ...), 添加点
- •lines(x, y, ...), 添加线段
- •text(x, y, labels, ...), 添加文字
- •abline(a, b, ...),添加直线y=a+bx
- •abline(h=y, ...),添加水平线
- •abline(v=x, ...),添加垂直线
- •polygon(x, y, ...),添加一个闭合的多边形
- •segments(x0, y0, x1, y1, ...), 画线段
- •arrows(x0, y0, x1, y1, ...), 画箭头
- •symbols(x, y, ...), 添加各种符号
- •legend(x, y, legend, ...), 添加图列说明