

Python程序设计

第八讲 科学计算、数据分析与可视化 Matplotlib



张 华
WHU

Matplotlib

Matplotlib简介

- ✱ **Matplotlib**是Python最著名的绘图库之一，提供了一整套和**MATLAB**相似的命令**API**，既适合交互式地进行制图，也可以作为绘图控件方便地嵌入到**GUI**应用程序中。
- ✱ **Matplotlib**的**pyplot**子库提供了和**MATLAB**类似的绘图**API**，方便用户快速绘制**2D**图表，包括直方图、饼图、散点图等。
- ✱ **Matplotlib**配合**NumPy**模块使用，可以实现科学计算结果的可视化显示。

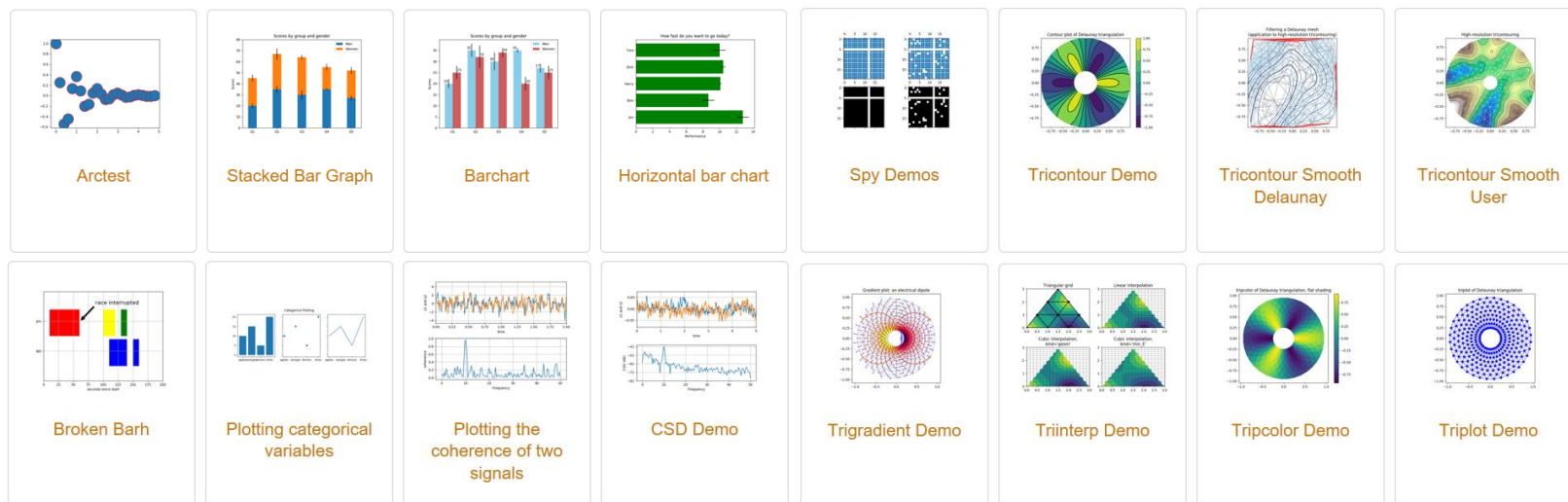
导入matplotlib.pyplot

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

Matplotlib

Matplotlib的绘图效果

 <https://matplotlib.org/gallery>



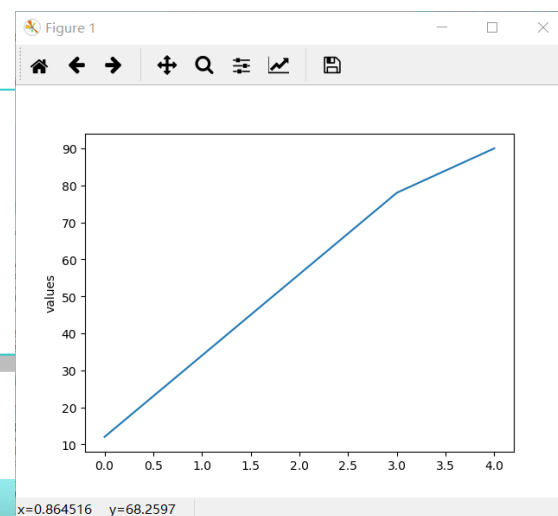
选取恰当的图形展示数据含义

Matplotlib的简单应用

使用Matplotlib绘图

- 使用Matplotlib模块绘图，主要使用了Matplotlib.pyplot工具包。
- Matplotlib是一套面向对象的绘图库，其绘制的图表中的每个绘图元素（例如线条、文字、刻度等）都是对象。
- 示例
 - `plt.plot()` 只有一个输入列表或数组时，参数被当作Y轴，X轴以索引自动生成。

```
>>> import matplotlib.pyplot as plt
>>> alist = [12, 34, 56, 78, 90]
>>> plt.plot(alist)
>>> plt.ylabel('values')
>>> plt.show()
```




Matplotlib的简单应用

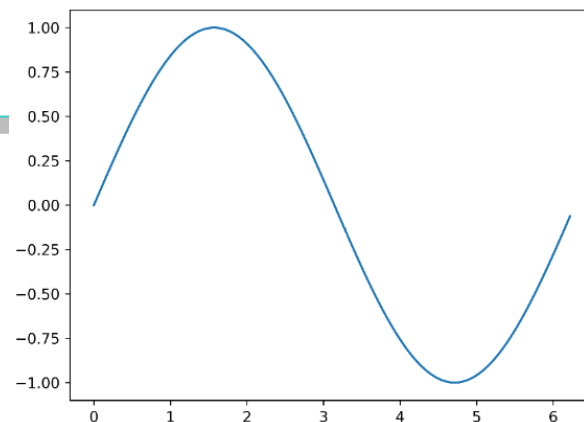
使用Matplotlib绘图

✿ 示例：绘制 $y=\sin(x)$ 的函数曲线

- `plt.plot(x,y)`当有两个以上参数时，按照X轴和Y轴顺序绘制数据点。
- `plt.savefig()`将输出图形存储为文件，默认PNG格式，可以通过`dpi`修改输出质量。

```
>>> import matplotlib.pyplot as plt
>>> import math
>>> x = [2*math.pi*i/100 for i in range(100)]
>>> y = [math.sin(i) for i in x]
>>> plt.plot(x, y)
>>> plt.savefig('sinfig', dpi=600)
```

 sinfig.png



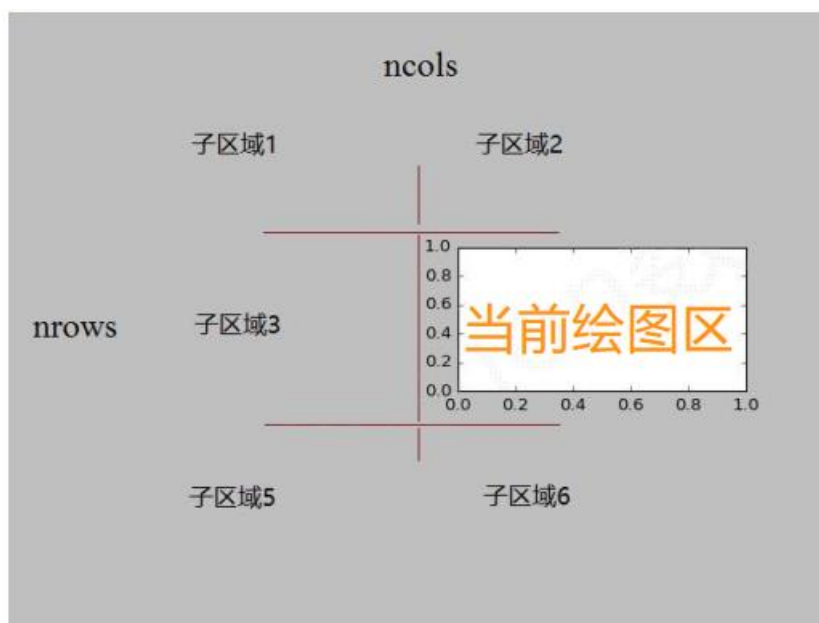
Matplotlib的简单应用

使用Matplotlib绘图

pyplot的绘图区域

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

➤ 在全局绘图区域中创建一个分区体系，并定位到一个子绘图区域。



```
plt.subplot(3,2,4)
```

或

```
plt.subplot(324)
```

Matplotlib的简单应用

使用Matplotlib绘图

✿ 示例：绘图多个子图

➤ 绘制 $y=e^{-t} \times \cos(2\pi x)$ 以及 $y=\cos(2\pi x)$ 的函数曲线。

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

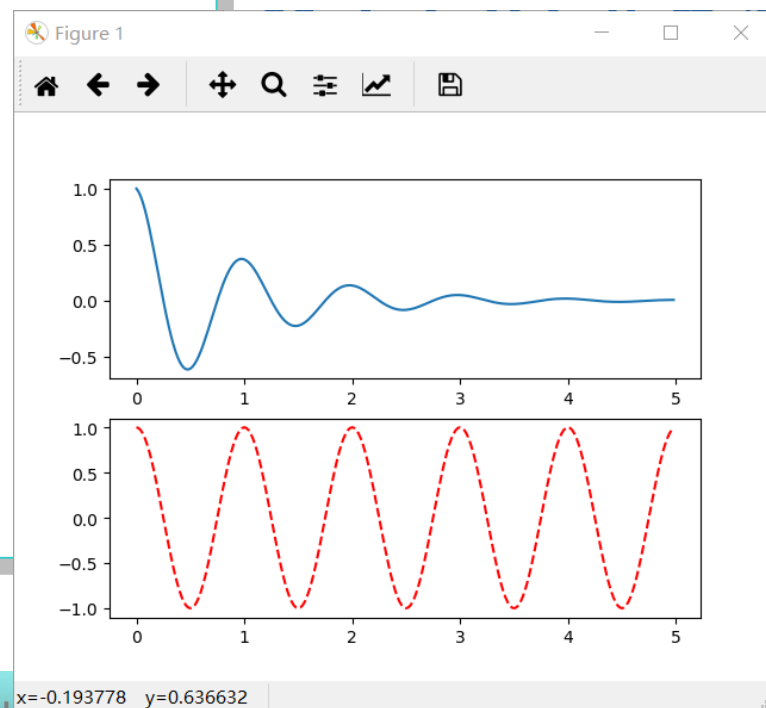
def f(t):
    return np.exp(-t)*np.cos(2*np.pi*t)

x = np.arange(0.0, 5.0, 0.02)

plt.subplot(2,1,1)
plt.plot(x, f(x))

plt.subplot(212)
plt.plot(x, np.cos(2*np.pi*x), 'r--')

plt.show()
```



Matplotlib的简单应用

使用Matplotlib绘图

pyplot的plot()函数

plt.plot(x, y, format_string, **kwargs)

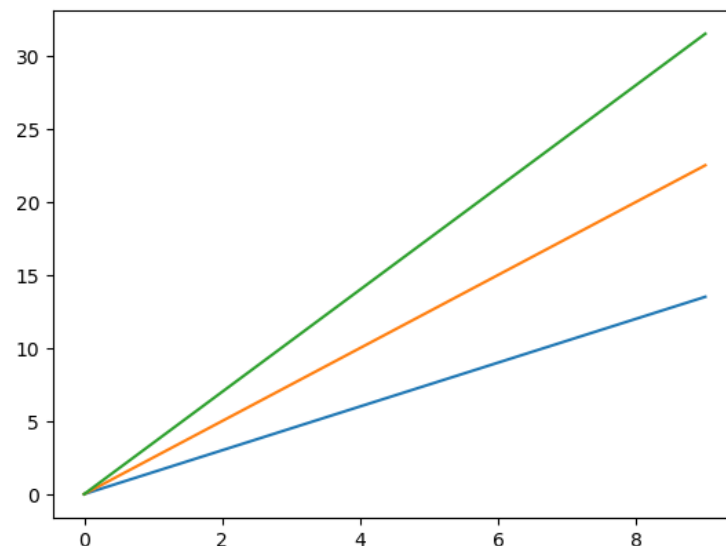
- **x** : X轴数据，列表或数组，可选
- **y** : Y轴数据，列表或数组
- **format_string**: 控制曲线的格式字符串，可选
- ****kwargs** : 第二组或更多(x,y,format_string)
- 当绘制多条曲线时，各条曲线的**x**不能省略。

Matplotlib的简单应用

使用Matplotlib绘图

pyplot的plot()函数

```
>>> import matplotlib.pyplot as plt  
>>> import numpy as np  
>>> x = np.arange(10)  
>>> plt.plot(x, x*1.5, x, x*2.5, x, x*3.5)  
>>> plt.show()
```



Matplotlib的简单应用

使用Matplotlib绘图

pyplot的plot()函数

- **format_string**: 控制曲线的格式字符串，可选
 - 由颜色字符、风格字符和标记字符组成

颜色字符	说明	颜色字符	说明
'b'	蓝色	'm'	洋红色 magenta
'g'	绿色	'y'	黄色
'r'	红色	'k'	黑色
'c'	青绿色 cyan	'w'	白色
'#008000'	RGB某颜色	'0.8'	灰度值字符串

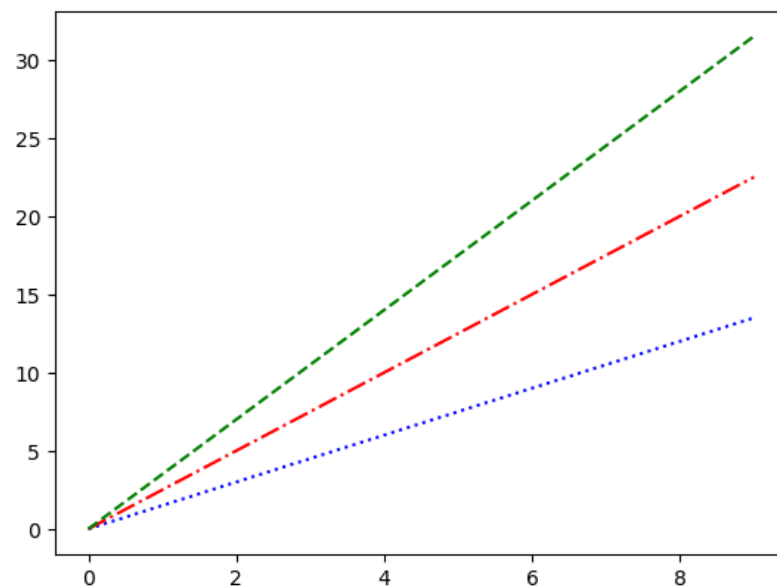
Matplotlib的简单应用

使用Matplotlib绘图

pyplot的plot()函数

- **format_string**: 控制曲线的格式字符串，可选
 - 由颜色字符、风格字符和标记字符组成

风格字符	说明
'—'	实线
'--'	破折线
'-.'	点划线
':'	虚线
''	无线条



```
plt.plot(x, x*1.5, 'b:', x, x*2.5, 'r-.', x, x*3.5, 'g--')
```

Matplotlib的简单应用

使用Matplotlib绘图

pyplot的plot()函数

- **format_string**: 控制曲线的格式字符串，可选
 - 由颜色字符、风格字符和标记字符组成

标记字符	说明	标记字符	说明	标记字符	说明
'.'	点标记	'1'	下花三角标记	'h'	竖六边形标记
','	像素标记(极小点)	'2'	上花三角标记	'H'	横六边形标记
'o'	实心圈标记	'3'	左花三角标记	'+'	十字标记
'v'	倒三角标记	'4'	右花三角标记	'x'	x标记
'^'	上三角标记	's'	实心方形标记	'D'	菱形标记
'>'	右三角标记	'p'	实心五角标记	'd'	瘦菱形标记
'<'	左三角标记	'*'	星形标记	' '	垂直线标记

Matplotlib的简单应用

使用Matplotlib绘图

pyplot的plot()函数

➤ **format_string**: 控制曲线的格式字符串，可选

- **color** : 控制颜色, **color='green'**
- **linestyle** : 线条风格, **linestyle='dashed'**
- **marker** : 标记风格, **marker='o'**
- **markerfacecolor**: 标记颜色, **markerfacecolor='blue'**
- **markersize** : 标记尺寸, **markersize=20**

Matplotlib的简单应用

使用Matplotlib绘图

显示中文

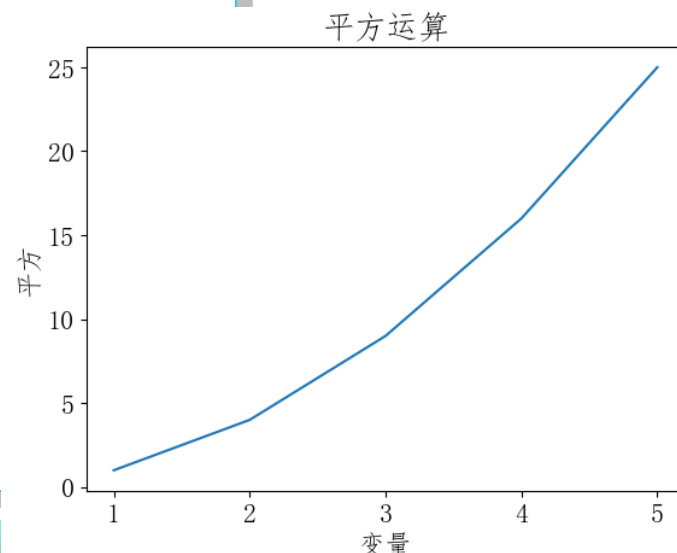
➤ 方法一：使用rcParams修改字体实现

```
import matplotlib.pyplot as plt
import matplotlib

matplotlib.rcParams['font.family']='FangSong'
matplotlib.rcParams['font.style']='italic'
matplotlib.rcParams['font.size']=16

squares = [1,4,9,16,25]
values = [1,2,3,4,5]
plt.plot(values, squares)

plt.ylabel('平方')
plt.xlabel('变量')
plt.title('平方运算')
plt.show()
```



Matplotlib的简单应用

使用Matplotlib绘图

显示中文

➤ 方法一：使用rcParams修改字体实现

属性	说明
'font.family'	用于显示字体的名字
'font.style'	字体风格，正常' <i>normal</i> '或 斜体' <i>italic</i> '
'font.size'	字体大小，整数字号或者' <i>large</i> '、' <i>x-small</i> '

中文字体	说明
'SimHei'	中文黑体
'Kaiti'	中文楷体
'LiSu'	中文隶书
'FangSong'	中文仿宋
'YouYuan'	中文幼圆
'STSong'	华文宋体

Matplotlib的简单应用

使用Matplotlib绘图

显示中文

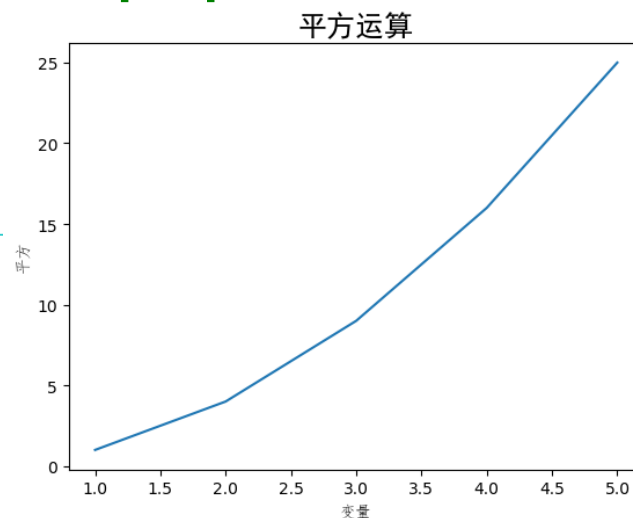
➤ 方法二：在有中文输出的地方，增加属性**fontproperties**

```
import matplotlib.pyplot as plt

squares = [1,4,9,16,25]
values = [1,2,3,4,5]
plt.plot(values, squares)

plt.ylabel('平方', fontproperties='FangSong')
plt.xlabel('变量', fontproperties='FangSong')
plt.title('平方运算', fontproperties='SimHei', fontsize=18)

plt.show()
```



Matplotlib的简单应用

使用Matplotlib绘图

✿ 文本显示

函数	说明
<code>plt.xlabel()</code>	对X轴增加文本标签
<code>plt.ylabel()</code>	对Y轴增加文本标签
<code>plt.title()</code>	对图形整体增加文本标签
<code>plt.text()</code>	在任意位置增加文本
<code>plt.annotate()</code>	在图形中增加带箭头的注解

Matplotlib的简单应用

使用Matplotlib绘图

文本显示示例

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

x = np.arange(0.0, 5.0, 0.02)
plt.plot(x, np.cos(2*np.pi*x), 'r--')

plt.xlabel('时间', fontproperties='FangSong', fontcolor='green')
plt.ylabel('振幅', fontproperties='FangSong', fontsize=16)
plt.title('正弦波实例  $y=\cos(2\pi x)$ ', fontproperties='SimHei', fontsize=18)
plt.annotate(r' $\mu=100$ ', xy=(2,1),
            xytext=(3,1.5), arrowprops=dict(facecolor='black', shrink=0.1,
            width=2))
plt.axis([-1, 6, -2, 2])
plt.grid(True)

plt.show()
```

Latex

