

数据分析DAY02

matplotlib概述

matplotlib是python的一个绘图库,使用它可以很方便的绘制出版质量级别的图形.

matplotlib的基本功能

1. 基本绘图

1. 绘制坐标系中连续的线,设置线型/线宽/颜色
2. 设置坐标轴的范围
3. 设置坐标刻度
4. 设置坐标轴
5. 显示图例
6. 绘制特殊点
7. 为图像添加备注.

2. 高级绘图

1. 绘制子图

2. 刻度定位器
3. 刻度网格线
4. 半对数坐标
5. 散点图
6. 图像填充
7. 条形图
8. 饼图

matplotlib基本功能详解

基本绘图

绘制一条线的相关API:

```
import numpy as np
import matplotlib.pyplot as plt
#xarray: 散点的x坐标数组
#yarray: 散点的y坐标数组
plt.plot(xarray, yarray)
plt.show()
```

绘制水平线与垂直线相关API:

#绘制一条垂直x轴的线，需要给x坐标值value,指定y坐标范围

```
mp.vlines(value, ymin, ymax, ..)
```

#绘制一条垂直y轴的线，需要给y坐标值value,指定x坐标范围

```
mp.hlines(value, xmin, xmax, ..)
```

线型/线宽/颜色

#linestyle: 线型

- or solid 直线

-- or dashed 虚线

-. or dashdot 点虚线

: or dot 点线

#linewidth: 线宽 (数字代表n倍线宽)

#color: 英文的颜色单词 或 常见颜色单词的首字母 或

#abcdef 或 (1, 1, 0.7) 或 (1, 1, 1)

#alpha: 设置透明度 0~1 0为完全透明

```
mp.plot(xarray, yarray,
```

```
        linestyle='',          # 线型
        linewidth=1,           # 线宽
        color='',              # 颜色
        alpha=0.5               # 透明度
    )
```

设置坐标轴范围

设置图像的可视区域.

```
#x_lim_min: 可视区域x的最小值
#x_lim_max: 可视区域x的最大值
mp.xlim(x_lim_min, x_lim_max)
#同上
mp.ylim(y_lim_min, y_lim_max)
```

设置坐标刻度

```
#设置x轴的坐标刻度
#x_val_list: 坐标值列表
#x_text_list: 坐标刻度列表
mp.xticks(x_val_list, x_text_list)
mp.yticks(y_val_list, y_text_list)
```

刻度文本的特殊语法 -- LaTeX排版语法规范(参考附录)

\$

$x^2 + y^2 = z^2, -\frac{\pi}{2}$

\$

$$x^2 + y^2 = z^2, -\frac{\pi}{2}$$

设置坐标轴

坐标轴包含四个: left / right / bottom / top

```
# getCurrentAxis 获取当前坐标轴对象
ax = mp.gca()
axl = ax.spines['left']
axr = ax.spines['right']
...
# 设置坐标轴的颜色
axl.set_color()
# 设置坐标轴的位置
# ('data', 0) 以坐标值作为定位参考, 设置坐标轴到0位置
axl.set_position((type, val))
```

显示图例

```
# 自动在窗口中某个位置添加图例
# 添加图例需要在调用mp.plot()绘制曲线时设置
label参数
mp.plot(..., label='y=sin(x)')
# 通过loc参数设置图例的位置

# =====
# Location String      Location Code
# =====
# 'best'              0
# 'upper right'       1
# 'upper left'        2
# 'lower left'        3
# 'lower right'       4
# 'right'             5
# 'center left'       6
# 'center right'      7
# 'lower center'      8
# 'upper center'      9
# 'center'            10

# =====
mp.legend(loc='')

```

绘制特殊点

```
# xarray与yarray是坐标序列
mp.scatter(xarray, yarray,
           marker='',      #点型
           s=3,            #大小
           edgecolor='',   #边缘色
           facecolor='',   #填充色
           zorder=3        #绘制顺序
)
```

marker点型详见附录

添加备注文本

```

mp.annotate(
    '',                                     # 备注内容
    xycoords='',                           # 备注目标点使用
的坐标系
    xy=(x, y),                             # 备注目标点的坐
标
    textcoords='',                         # 备注文本使用的
坐标系
    xytext=(x, y),                         # 备注文本的坐标
    fontsize=14,                           # 备注文本字体大
小
    arrowprops=dict()                     # 指示箭头的属性
)

```

arrowprops参数使用字典定义指向目标点的箭头样式

```

arrowprops=dict(
    arrowstyle='',                         # 定义箭头样式
    connectionstyle=''                     # 定义连接线的样
式
)

```

arrowstyle 参见 *help(mp.annotate)*

高级图形窗口对象操作

一次绘制两个窗口

```
mp.figure(  
    '', # 窗口标题  
    figsize=(4, 3), # 窗口大小  
    facecolor=' ' # 窗口颜色  
)  
mp.show()
```

mp.figure()方法可以创建多个窗口, 每个窗口的标题不同. 后续调用mp的方法进行绘制时将作用于当前窗口. 如果希望修改以前已经创建过的窗口, 可以通过相同的窗口标题调用mp.figure()方法把该窗口置为当前窗口.

设置当前窗口的常用参数

#设置图表的标题

```
mp.title('', fontsize=18)
```

#设置窗口中x坐标轴的文本即y坐标轴的文本

```
mp.xlabel('', fontsize=12)
```

```
mp.ylabel('', fontsize=12)
```

#设置刻度参数(刻度字体大小)

```
mp.tick_params(labelsize=8)
```

#设置图表网格线

```
mp.grid(linestyle=':')
```

#设置紧凑布局

```
mp.tight_layout()
```

绘制子图

矩阵式布局

```
mp.figure('')
# 开始绘制一个子图
# 通过参数rows与cols拆分当前窗口，每个子窗口
都将分配一
# 个序号，1~x
mp.subplot(rows, cols, 1)
mp.title()
mp.grid()
mp.subplot(2, 3, 2)
mp.subplot(233)
mp.show()
```

案例:绘制九宫格子图, 每个子图写一个文本.

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

mp.figure('Sub Layout',
facecolor='gray')
for i in range(1, 10):
    mp.subplot(3, 3, i)
    mp.text(0.5, 0.5, i, ha='center',
            va='center', size=36,
alpha=0.8)
    mp.xticks([])
    mp.yticks([])
    mp.tight_layout()
mp.show()
```

网格式布局

```
import matplotlib.gridspec as mg
mp.figure('')
# 该方法将会返回子图的二维数组
gs = mg.GridSpec(3, 3)
# 通过subplot对子图进行合并
# gs[0, :2]->合并0行中的0/1列作为1个子图进行绘制
mp.subplot(gs[0, :2])
mp.show()
```

自由布局

```
mp.figure('')
# left_bottom_x:子图左下角点的横坐标
# left_bottom_y:子图左下角点的纵坐标
# w: 宽度
# h: 高度
mp.axes([left_bottom_x,
left_bottom_y,w, h])
mp.show()
```

刻度定位器

```
# 获取当前坐标轴
ax = mp.gca()
# 设置x轴的主刻度定位器为NullLocator()
ax.xaxis.set_major_locator(mp.NullLocator())
# 设置x轴的次刻度定位器为MultipleLocator()
ax.xaxis.set_minor_locator(mp.MultipleLocator())
```

案例: 画个数轴

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

locators = ['mp.NullLocator()',
            'mp.MaxNLocator(nbins=4)',
            'mp.FixedLocator(locs=[0,
2.5, 5, 7.5, 10])',
            'mp.AutoLocator()',
            'mp.MultipleLocator()',
            'mp.LogLocator(base=2)']

mp.figure('Locators',
facecolor='lightgray')
```

```
for i, locator in enumerate(locators):
    mp.subplot(len(locators), 1, i +
1)
    mp.xlim(0, 10)
    mp.ylim(-10, 10)
    mp.yticks([])

    ax = mp.gca()
    ax.spines['top'].set_color('none')

    ax.spines['left'].set_color('none')

    ax.spines['right'].set_color('none')

    ax.spines['bottom'].set_position(('da
ta', 0))

    ax.xaxis.set_major_locator(eval(locat
or))
    ax.xaxis.set_minor_locator(
        mp.MultipleLocator(0.1))
```

```
mp.plot(np.arange(11),  
np.zeros(11), c='none')  
mp.text(5, 0.3, locator,  
        ha='center', size=12)  
mp.tight_layout()  
  
mp.show()
```

刻度网格线

```
ax = mp.gca()  
ax.grid(  
    which='',          #'major' / 'minor'  
    axis='',          #'x' / 'y' /  
    'both'  
    linewidth=1,      # 线宽  
    linestyle='',     # 线型  
    color='',  
    alpha=0.5  
)
```

半对数坐标系

y轴将会以指数方式递增.

```
mp.figure()  
mp.semilogy(x, y, ....)  
mp.show()
```

绘制散点图

```
mp.scatter(  
    xarray,  
    yarray,  
    marker=' ',  
    s=10,  
    color=' ',  
    edgecolor=' ',  
    facecolor=' ',  
    zorder=3  
)
```

使用numpy.random的normal函数生成符合二项分布的随机数.

```
n = 100
# 172: 期望值
# 20: 标准差
# n: 数字生成数量
x = np.random.normal(172, 20, n)
y = np.random.normal(60, 10, n)
```

设置点的颜色

```
# d的值是一个大于0的数
# 若所有点计算出的d处于[0-1000]区间
# 那么绘制该点时所使用的颜色,可以根据d的值去
jet颜色映射
# 表中取值(即如果d取值为500,则使用jet颜色映
射表中最中心
# 的颜色值)
d = (x-172)**2 + (y-60)**2
mp.scatter(x, y, c=d, cmap='jet')
```

附录

	black		bisque		lightgreen		slategrey
	k		darkorange		forestgreen		lightsteelblue
	dimgray		burlywood		limegreen		cornflowerblue
	dimgrey		antiquewhite		darkgreen		royalblue
	grey		tan		green		ghostwhite
	gray		navajowhite		g		lavender
	darkgrey		blanchedalmond		lime		midnightblue
	darkgray		papayawhip		seagreen		navy
	silver		moccasin		mediumseagreen		darkblue
	lightgray		orange		springgreen		mediumblue
	lightgrey		wheat		mintcream		blue
	gainsboro		oldlace		mediumspringgreen		b
	whitesmoke		floralwhite		mediumaquamarine		slateblue
	white		darkgoldenrod		aquamarine		darkslateblue
	w		goldenrod		turquoise		mediumslateblue
	snow		cornsilk		lightseagreen		mediumpurple
	rosybrown		lemonchiffon		mediumturquoise		blueviolet
	lightcoral		khaki		azure		indigo
	indianred		palegoldenrod		lightcyan		darkorchid
	brown		darkkhaki		paleturquoise		darkviolet
	firebrick		ivory		darkslategray		mediumorchid
	maroon		beige		darkslategrey		thistle
	darkred		lightyellow		teal		plum
	red		lightgoldenrodyellow		darkcyan		violet
	r		olive		c		purple
	mistyrose		y		cyan		darkmagenta
	salmon		yellow		aqua		m
	tomato		olivedrab		darkturquoise		fuchsia
	darksalmon		yellowgreen		cadetblue		magenta
	coral		darkolivegreen		powderblue		orchid
	orangered		greenyellow		lightblue		mediumvioletred
	lightsalmon		chartreuse		deepskyblue		deeppink
	sienna		lawngreen		skyblue		hotpink
	seashell		sage		lightskyblue		lavenderblush
	chocolate		lightsage		steelblue		palevioletred
	saddlebrown		darksage		aliceblue		crimson
	sandybrown		honeydew		dodgerblue		pink
	peachpuff		darkseagreen		lightslategrey		lightpink
	peru		palegreen		lightslategray		
	linen				slategray		

LaTeX语法字符串

表 3.1: 数学模式重音符

\hat{a}	<code>\hat{a}</code>	\check{a}	<code>\check{a}</code>	\tilde{a}	<code>\tilde{a}</code>	\acute{a}	<code>\acute{a}</code>
\grave{a}	<code>\grave{a}</code>	\dot{a}	<code>\dot{a}</code>	\ddot{a}	<code>\ddot{a}</code>	\breve{a}	<code>\breve{a}</code>
\bar{a}	<code>\bar{a}</code>	\vec{a}	<code>\vec{a}</code>	\widehat{A}	<code>\widehat{A}</code>	\widetilde{A}	<code>\widetilde{A}</code>

表 3.2: 小写希腊字母

α	<code>\alpha</code>	θ	<code>\theta</code>	o	<code>o</code>	v	<code>\upsilon</code>
β	<code>\beta</code>	ϑ	<code>\vartheta</code>	π	<code>\pi</code>	ϕ	<code>\phi</code>
γ	<code>\gamma</code>	ι	<code>\iota</code>	ϖ	<code>\varpi</code>	φ	<code>\varphi</code>
δ	<code>\delta</code>	κ	<code>\kappa</code>	ρ	<code>\rho</code>	χ	<code>\chi</code>
ϵ	<code>\epsilon</code>	λ	<code>\lambda</code>	ϱ	<code>\varrho</code>	ψ	<code>\psi</code>
ε	<code>\varepsilon</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>	ω	<code>\omega</code>
ζ	<code>\zeta</code>	ν	<code>\nu</code>	ς	<code>\varsigma</code>		
η	<code>\eta</code>	ξ	<code>\xi</code>	τ	<code>\tau</code>		

表 3.3: 大写希腊字母

Γ	<code>\Gamma</code>	Λ	<code>\Lambda</code>	Σ	<code>\Sigma</code>	Ψ	<code>\Psi</code>
Δ	<code>\Delta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>	Ω	<code>\Omega</code>
Θ	<code>\Theta</code>	Π	<code>\Pi</code>	Φ	<code>\Phi</code>		

表 3.4: 二元关系符

下述命令的前面加上 `\not` 来得到其否定形式。

$<$	<code><</code>	$>$	<code>></code>	$=$	<code>=</code>
\leq	<code>\leq</code> or <code>\le</code>	\geq	<code>\geq</code> or <code>\ge</code>	\equiv	<code>\equiv</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\doteq	<code>\doteq</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>	\sim	<code>\sim</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>	\simeq	<code>\simeq</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\approx	<code>\approx</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\cong	<code>\cong</code>
\sqsubset^a	<code>\sqsubset^a</code>	\sqsupset^a	<code>\sqsupset^a</code>	\Join^a	<code>\Join^a</code>
\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\bowtie	<code>\bowtie</code>
\in	<code>\in</code>	\ni	<code>\ni</code> , <code>\owns</code>	\propto	<code>\propto</code>
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>	\models	<code>\models</code>
\mid	<code>\mid</code>	\parallel	<code>\parallel</code>	\perp	<code>\perp</code>
\smile	<code>\smile</code>	\frown	<code>\frown</code>	\asymp	<code>\asymp</code>
$:$	<code>:</code>	\notin	<code>\notin</code>	\neq	<code>\neq</code> or <code>\ne</code>

表 3.5: 二元运算符

$+$	<code>+</code>	$-$	<code>-</code>	\triangleleft	<code>\triangleleft</code>
\pm	<code>\pm</code>	\mp	<code>\mp</code>	\trianglelefteq	<code>\trianglelefteq</code>

\cdot	<code>\caot</code>	\div	<code>\div</code>	\triangleright	<code>\trianglerightright</code>
\times	<code>\times</code>	\setminus	<code>\setminus</code>	\star	<code>\star</code>
\cup	<code>\cup</code>	\cap	<code>\cap</code>	\ast	<code>\ast</code>
\sqcup	<code>\sqcup</code>	\sqcap	<code>\sqcap</code>	\circ	<code>\circ</code>
\vee	<code>\vee</code> , <code>\lor</code>	\wedge	<code>\wedge</code> , <code>\land</code>	\bullet	<code>\bullet</code>
\oplus	<code>\oplus</code>	\ominus	<code>\ominus</code>	\diamond	<code>\diamond</code>
\odot	<code>\odot</code>	\oslash	<code>\oslash</code>	\uplus	<code>\uplus</code>
\otimes	<code>\otimes</code>	\bigcirc	<code>\bigcirc</code>	\amalg	<code>\amalg</code>
\triangleup	<code>\bigtriangleup</code>	\triangledown	<code>\bigtriangledown</code>	\dagger	<code>\dagger</code>
\triangleleft	<code>\lhd</code> ^a	\triangleright	<code>\rhd</code> ^a	\ddagger	<code>\ddagger</code>
\trianglelefteq	<code>\unlhd</code> ^a	\trianglerighteq	<code>\unrhd</code> ^a	\wr	<code>\wr</code>

表 3.8: 定界符

$($	<code>(</code>	$)$	<code>)</code>	\uparrow	<code>\uparrow</code>	\Uparrow	<code>\Uparrow</code>
$[$	<code>[</code> or <code>\lbrack</code>	$]$	<code>] or \rbrack</code>	\downarrow	<code>\downarrow</code>	\Downarrow	<code>\Downarrow</code>
$\{$	<code>\{ or \lbrace</code>	$\}$	<code>\} or \rbrace</code>	\updownarrow	<code>\updownarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\langle	<code>\langle</code>	\rangle	<code>\rangle</code>	$ $	<code> or \vert</code>	$\ $	<code>\ or \Vert</code>
\lfloor	<code>\lfloor</code>	\rfloor	<code>\rfloor</code>	\lceil	<code>\lceil</code>	\rceil	<code>\rceil</code>
$/$	<code>/</code>	\backslash	<code>\backslash</code>	\cdot	<code>\cdot</code> (dual. empty)		

表 3.9: 大尺寸定界符

$\left($	<code>\lgroup</code>	$\right)$	<code>\rgroup</code>	$\left\{$	<code>\lmoustache</code>	$\right\}$	<code>\rmoustache</code>
\uparrow	<code>\arrowvert</code>	\Uparrow	<code>\Arrowvert</code>	$\left $	<code>\bracevert</code>		

`\begin{displaymath}`
`\sum_{i=1}^n \quad`
`\int_0^{\frac{\pi}{2}} \quad`
`\prod_{\epsilon}`
`\end{displaymath}`

$$\sum_{i=1}^n \int_0^{\frac{\pi}{2}} \prod_{\epsilon}$$

`a_{1} \quad x^{2} \quad`
`$e^{-\alpha t}$ \quad`
`a^{3}_{ij}\`
`$e^{x^2} \neq {e^x}^2$`

$$a_1 \qquad x^2 \qquad e^{-\alpha t} \qquad a^3_{ij}$$
$$e^{x^2} \neq e^{x^2}$$

`$_{\sqrt{x}}$ \quad`
`$_{\sqrt{x^2+\sqrt{y}}}$`
`\quad $_{\sqrt[3]{2}}$\\[3pt]`
`$_{\sqrt{x^2+y^2}}$`

$$\sqrt{x} \qquad \sqrt{x^2+\sqrt{y}} \qquad \sqrt[3]{2}$$
$$\sqrt{[x^2+y^2]}$$

`$1\frac{1}{2}$~hours`
`\begin{displaymath}`
`\frac{x^2}{k+1} \quad`
`x^{\frac{2}{k+1}} \quad`
`x^{1/2}`
`\end{displaymath}`

$$1\frac{1}{2} \text{ hours}$$
$$\frac{x^2}{k+1} \qquad x^{\frac{2}{k+1}} \qquad x^{1/2}$$

^ : triangle_up



3 : tri_left



P : plus (filled)



x : x



_ : hline



v : triangle_down



2 : tri_up



p : pentagon



+ : plus



| : vline



o : circle



1 : tri_down



s : square



H : hexagon2



d : thin_diamond



, : pixel



> : triangle_right



8 : octagon



h : hexagon1



D : diamond



. : point



< : triangle_left



4 : tri_right



* : star



X : x (filled)

