

数据可视化工具

- 数据 + 思路 + 工具 + 绘制 + 调试 = 绘图

绘图的最终效果跟投入的时间成本成正比，需要好学耐心多尝试，才能做出漂亮的数据图。

Matplotlib 官网：<http://matplotlib.org> Matplotlib 是一个 Python 的 2D 绘图库，通过 Matplotlib，开发者可以仅需要几行代码，便可以生成绘图，直方图，功率谱，条形图，错误图，散点图等。

plotly 官网：<https://plot.ly/> plotly 是开挂的作图神器，可以供 js, python, R, DB 等使用

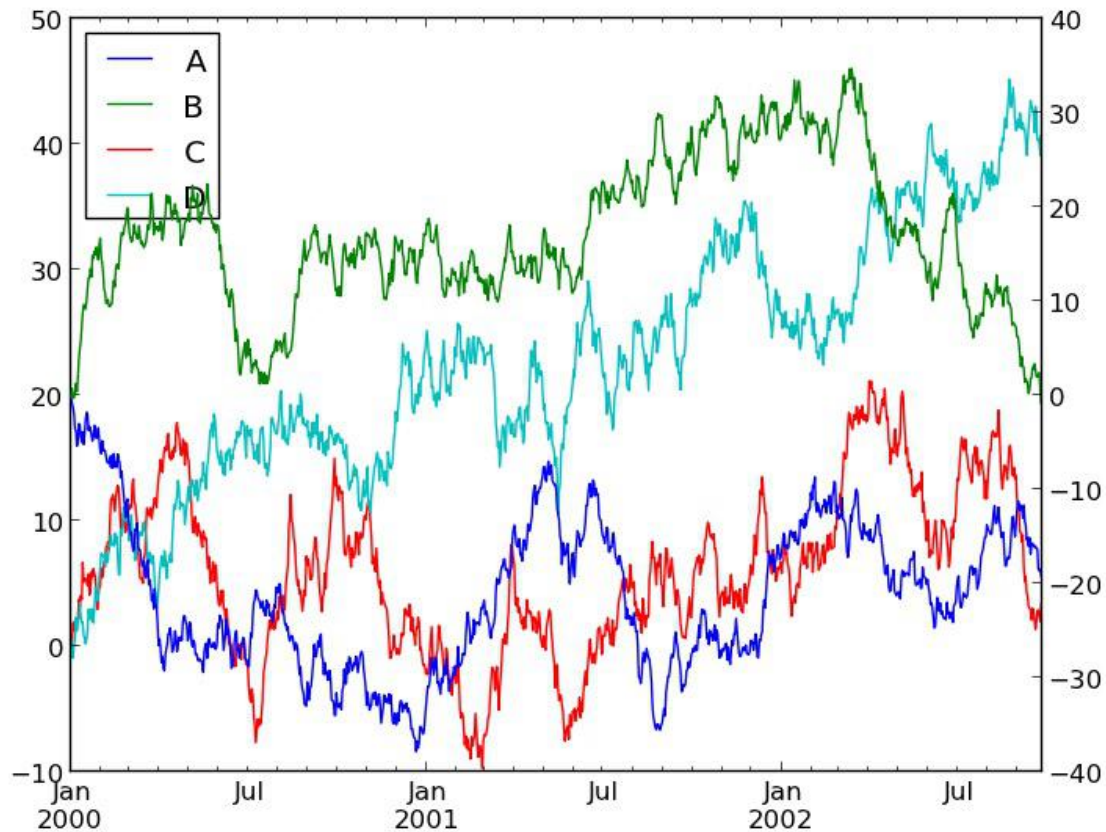
1. Matplotlib

Matplotlib 可能是 Python 2D-绘图领域使用最广泛的套件。它能让使用者很轻松地将数据图形化，并且提供多样化的输出格式

文档地址：<http://matplotlib.org>

- 用于创建出版质量图表的绘图工具库
- 目的是为 Python 构建一个 Matlab 式的绘图接口
- `import matplotlib.pyplot as plt`
- `pyplot` 模块包含了常用的 matplotlib API 函数，承担了大部分的绘图任务。

http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.plot



1.1 figure 对象

在 Matplotlib 中，整个图像为一个 figure 对象

Matplotlib 的图像均位于 figure 对象中

创建 figure : `fig = plt.figure()`

如果不创建 figure 对象，matplotlib 会自动创建一个 figure 对象。

示例代码：

```
# 引入 matplotlib 包
import matplotlib.pyplot as plt
import numpy as np

% matplotlib inline # 在 jupyter notebook 里需要使用这一句命令

# 创建 figure 对象，通过 figsize 参数可以指定绘图对象的宽度和高度，单位为英寸；dpi 参
```

数指定绘图对象的分辨率，即每英寸多少个像素，缺省值为 80。

因此本例中所创建的图表窗口为 640 * 320 像素。

```
plt.figure(figsize=(8, 4))
```

运行结果：

<matplotlib.figure.Figure at 0x11a2dd7b8>

1.2 subplot 分隔区域

- subplot 命令是将图片窗口划分成若干区域,按照一定顺序使得图形在每个小区域内呈现其图形。
- 在 figure 对象中可以包含一个或者多个 Axes 对象。
- 每个 Axes(ax)对象都是一个拥有自己坐标系统的绘图区域
- fig.add_subplot(a, b, c)
 - a, b 表示将 fig 分割成 a * b 的区域
 - c 表示当前选中要操作的区域，
 - 注意：从 1 开始编号（不是从 0 开始）
- plot 绘图的区域是最后一次指定 subplot 的位置 (jupyter notebook 里不能正确显示)

示例代码：

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

fig = plt.figure()
# 指定切分区域的位置
ax1 = fig.add_subplot(2, 2, 1)
ax2 = fig.add_subplot(2, 2, 2)
ax3 = fig.add_subplot(2, 2, 3)
ax4 = fig.add_subplot(2, 2, 4)

# 在 subplot 上作图
random_arr = np.random.randn(100)
```

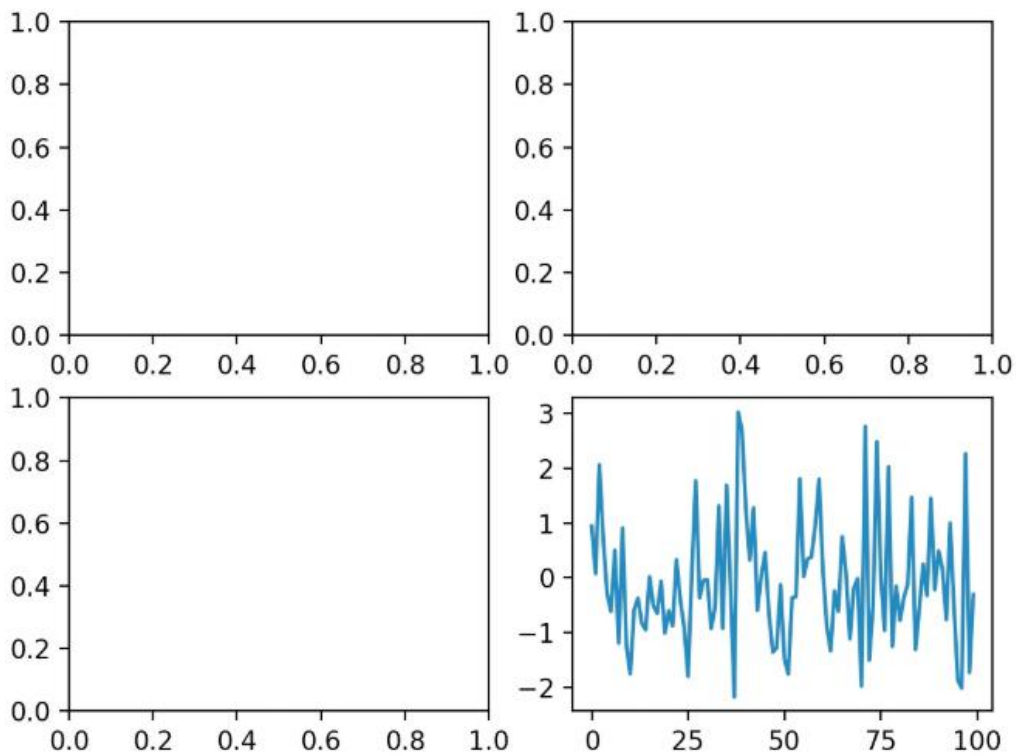
```
# 可以指定在某个或多个 subplot 位置上作图
# ax1.plot(random_arr)
# ax2.plot(random_arr)
# ax3.plot(random_arr)
# ax4.plot(random_arr)

# 如果不指定，则默认是在最后一次使用 subplot 的位置上作图，但是在 jupyter notebook
里可能显示有误
plt.plot(random_arr)

# 保存绘图结果到本地磁盘文件
plt.savefig("random_figure.png")

# 显示绘图结果
plt.show()
```

运行结果：



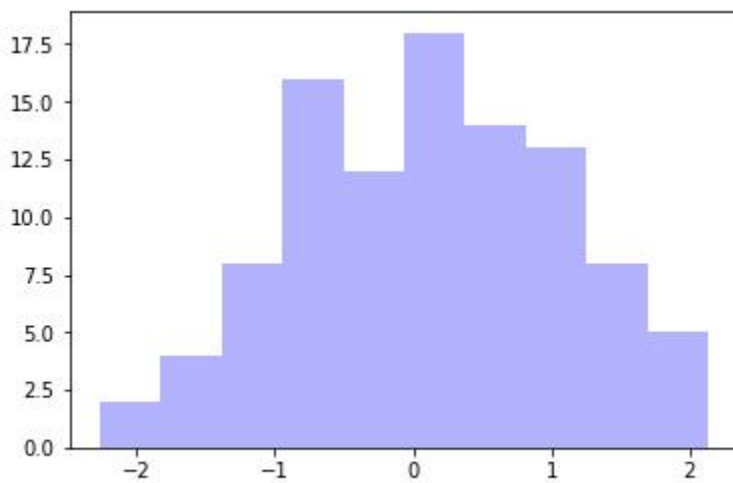
1.3 直方图 : hist

示例代码：

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

plt.hist(np.random.randn(100), bins=10, color='b', alpha=0.3)
plt.show()
```

运行结果：



1.4 散点图 : scatter

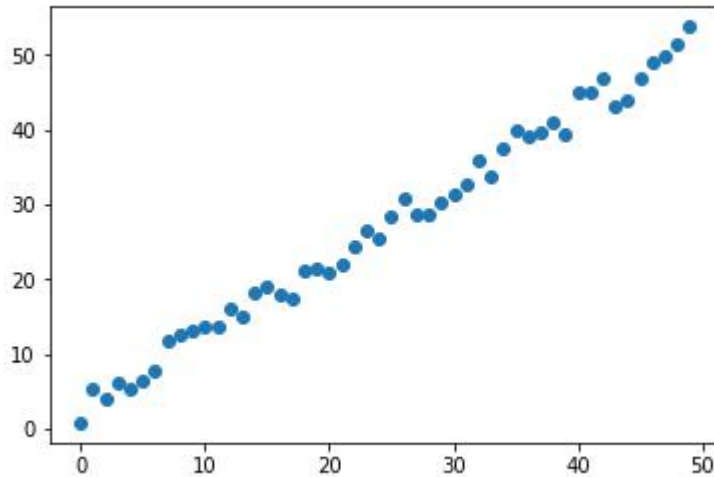
示例代码：

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

# 绘制散点图
# x 表示横坐标, y 表示纵坐标
x = np.arange(50)
y = x + 5 * np.random.rand(50)
plt.scatter(x, y)
```

```
plt.show()
```

运行结果：



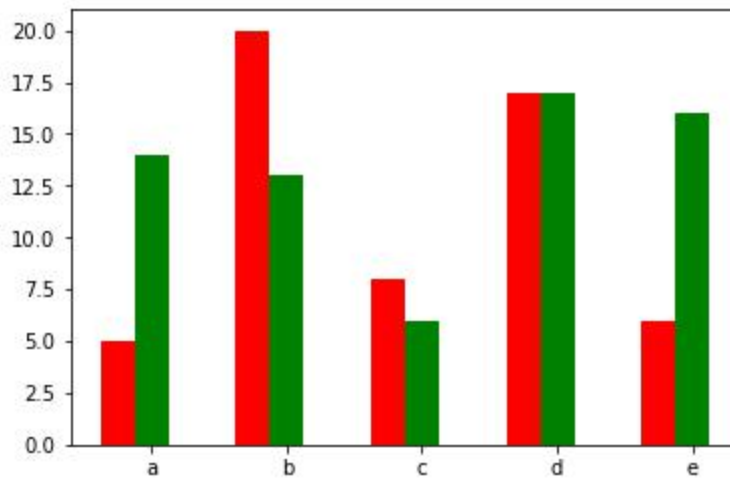
1.5 柱状图：bar

示例代码：

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

# 柱状图
x = np.arange(5)
y1, y2 = np.random.randint(1, 25, size=(2, 5))
width = 0.25
ax = plt.subplot(1, 1, 1)
ax.bar(x, y1, width, color='r')
ax.bar(x + width, y2, width, color='g')
ax.set_xticks(x + width)
ax.set_xticklabels(['a', 'b', 'c', 'd', 'e'])
plt.show()
```

运行结果：



1.6 矩阵绘图：imshow()

- 混淆矩阵，三个维度的关系

示例代码：

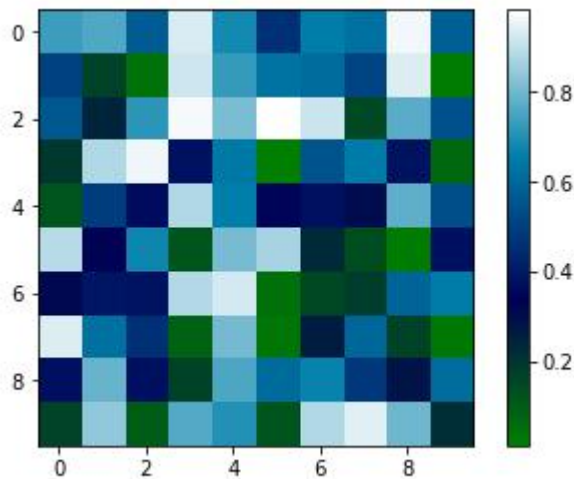
```
import matplotlib.pyplot as plt
import numpy as np

# 矩阵绘图
m = np.random.rand(10, 10)
print(m)
# interpolation 代表的是插值运算，'nearest'只是选取了其中的一种插值方式（邻近算法，了解即可）；
# cmap 表示绘图时的样式，这里选择的是 ocean 主题。
plt.imshow(m, interpolation='nearest', cmap=plt.cm.ocean)
plt.colorbar()
plt.show()
```

运行结果：

```
[[ 0.92859942  0.84162134  0.37814667  0.46401549  0.93935737  0.0344159
    0.56358375  0.75977745  0.87983192  0.22818774]
 [ 0.88216959  0.43369207  0.1303902   0.98446182  0.59474031  0.04414217
    0.86534444  0.34919228  0.53950028  0.89165269]]
```

```
[ 0.52919761  0.87408715  0.097871   0.78348534  0.09354791  0.3186
  0.25978432  0.48340623  0.1107699  0.14065592]
[ 0.90834516  0.42377475  0.73042695  0.51596826  0.14154431  0.22165693
  0.64705882  0.78062873  0.55036304  0.40874584]
[ 0.98853697  0.46762114  0.69973423  0.7910757  0.63700306  0.68793919
  0.28685306  0.3473426  0.17011744  0.18812329]
[ 0.73688943  0.58004874  0.03146167  0.08875797  0.32930191  0.87314734
  0.50757536  0.8667078  0.8423364  0.99079049]
[ 0.37660356  0.63667774  0.78111565  0.25598593  0.38437628  0.95771051
  0.01922366  0.37020219  0.51020305  0.05365718]
[ 0.87588452  0.56494761  0.67320078  0.46870376  0.66139913  0.55072149
  0.51328222  0.64817353  0.198525  0.18105368]
[ 0.86038137  0.55914088  0.55240021  0.15260395  0.4681218  0.28863395
  0.6614597  0.69015592  0.46583629  0.15086562]
[ 0.01373772  0.30514083  0.69804049  0.5014782  0.56855904  0.14889117
  0.87596848  0.29757133  0.76062891  0.03678431]]
```



1.7 plt.subplots()

同时返回新创建的 figure 和 subplot 对象数组

生成 2 行 2 列 subplot: `fig, subplot_arr = plt.subplots(2,2)`

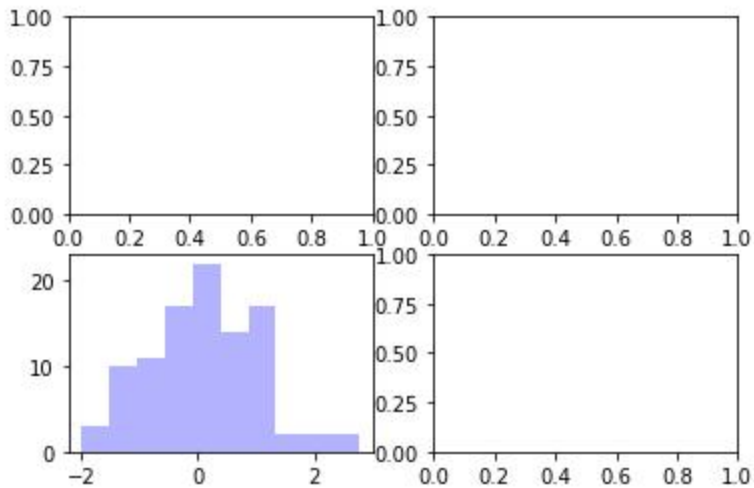
在 jupyter 里可以正常显示，推荐使用这种方式创建多个图表

示例代码：

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

fig, subplot_arr = plt.subplots(2, 2)
# bins 为显示个数，一般小于等于数值个数
subplot_arr[1, 0].hist(np.random.randn(100), bins=10, color='b', alpha=0.3)
plt.show()
```

运行结果：



1.8 颜色、标记、线型

- `ax.plot(x, y, 'r--')`

等价于 `ax.plot(x, y, linestyle='--', color='r')`

示例代码：

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

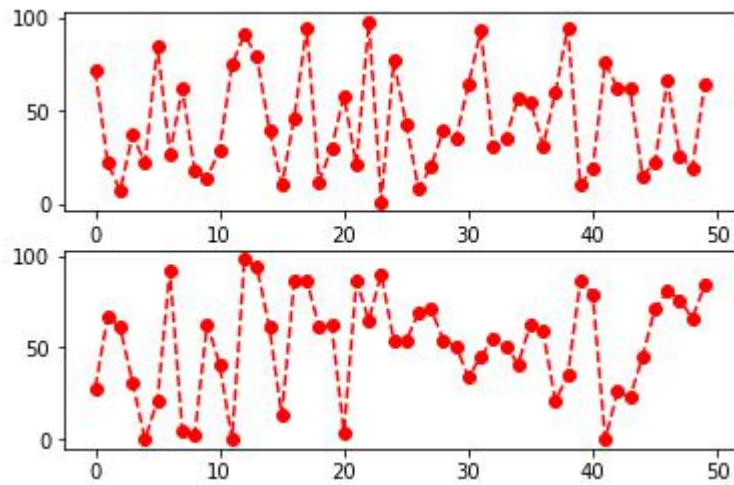
fig, axes = plt.subplots(2)
axes[0].plot(np.random.randint(0, 100, 50), 'ro--')
```

等价

```
axes[1].plot(np.random.randint(0, 100, 50), color='r', marker='o', linestyle='dashed')
```

```
plt.show()
```

运行结果：



- b: blue
- g: green
- r: red
- c: cyan
- m: magenta
- y: yellow
- k: black
- w: white

常用的颜色、标记、线型:

| marker | description |
|--------|---------------|
| "." | point |
| "." | pixel |
| "o" | circle |
| "v" | triangle_down |
| "^" | triangle_up |
| "<" | triangle_left |

| linestyle | description |
|-------------------|------------------|
| '-' or 'solid' | solid line |
| '--' or 'dashed' | dashed line |
| '-.' or 'dashdot' | dash-dotted line |
| ':' or 'dotted' | dotted line |
| 'None' | draw nothing |
| ' ' | draw nothing |
| ' ' | draw nothing |

1.9 刻度、标签、图例

- 设置刻度范围
 - plt.xlim(), plt.ylim()

- `ax.setxlim(), ax.setylim()`
- 设置显示的刻度
 - `plt.xticks(), plt.yticks()`
 - `ax.setxticks(), ax.setyticks()`
- 设置刻度标签
 - `ax.setxticklabels(), ax.setyticklabels()`
- 设置坐标轴标签
 - `ax.setxlabel(), ax.setylabel()`
- 设置标题
 - `ax.set_title()`
- 图例
 - `ax.plot(label= 'legend')`
 - `ax.legend(), plt.legend()`
 - `loc= 'best'` : 自动选择放置图例最佳位置

示例代码：

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

fig, ax = plt.subplots(1)
ax.plot(np.random.randn(1000).cumsum(), label='line0')

# 设置刻度
# plt.xlim([0,500])
ax.set_xlim([0, 800])

# 设置显示的刻度
# plt.xticks([0,500])
```

```
ax.set_xticks(range(0, 500, 100))

# 设置刻度标签
ax.set_yticklabels(['Jan', 'Feb', 'Mar'])

# 设置坐标轴标签
ax.set_xlabel('Number')
ax.set_ylabel('Month')

# 设置标题
ax.set_title('Example')

# 图例
ax.plot(np.random.randn(1000).cumsum(), label='line1')
ax.plot(np.random.randn(1000).cumsum(), label='line2')

ax.legend()
plt.show()
```

运行结果：

Example

