



# 用Matplotlib展示数据

北京理工大学计算机学院 高玉金

2019年3月



# 用Matplotlib绘图

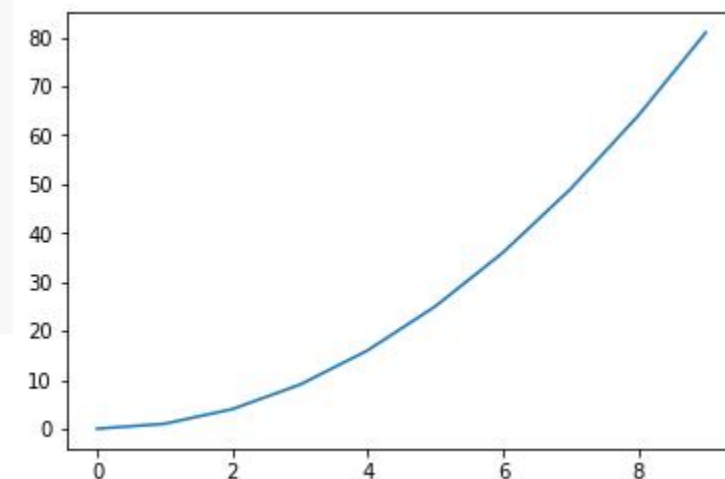
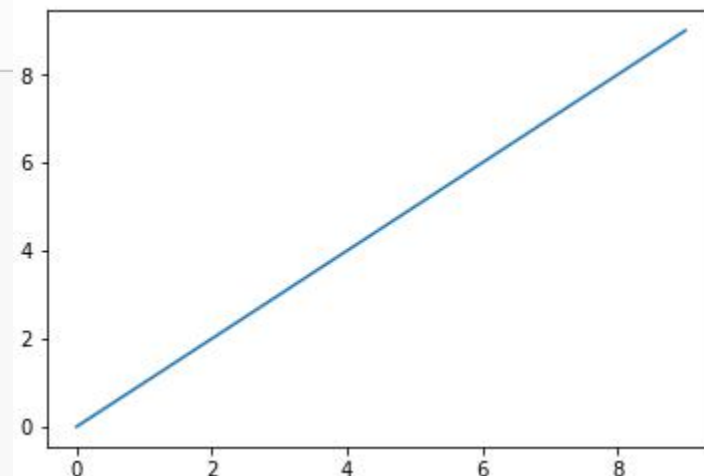
- 在数据分析和处理中，把数据可视化也是一个主要手段，如在发现数据异常值时，通过散点图就可以非常容易的看出哪些数据过于分散
- Matplotlib是和numpy，pandas配合使用的一个非常优秀的三方库
- Matplotlib用于在python环境下进行Matlab风格的绘图
- `import matplotlib.pyplot as plt`



# Matplotlib入门

- 只需要一条语句，即可绘图
- `plt.plot(range(10))`
- `plt.plot([x],y,[format], ...)`
- `x`和`y`为序列类型

```
x=range(10)
y=[k**2 for k in x]
plt.plot(x,y)
```

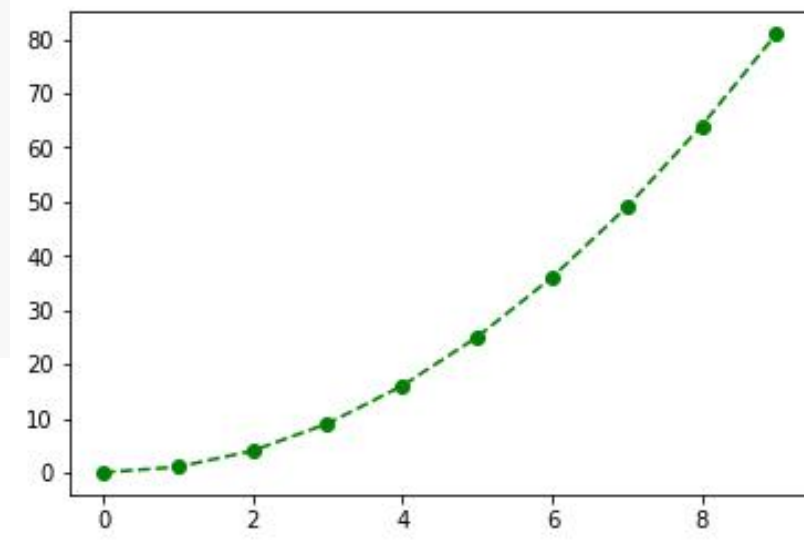




# 控制字符串

- 依次包括：颜色字符、风格字符和标记字符，如 “go--”
  - 颜色：r、g、b等，或 数字格式 '#008080'
  - 风格：-、--、-.、|（实线、虚线、点画线等）
  - 标记：.、,、v、<、>等
- 也可以使用显式表达：
  - color= “g”
  - linestyle= “dashed”
  - marker = “o”

```
x=range(10)
y=[k**2 for k in x]
plt.plot(x,y,'go--')
```

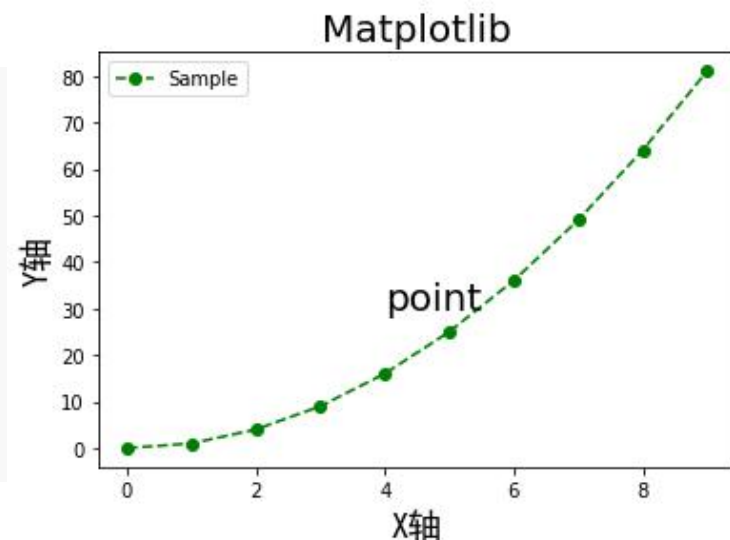




# 图形文本显示

- 设置标题: `title()`
- 图例标注: `label`
- X轴标注: `xlabel()`
- Y轴标注: `ylabel()`
- 文本信息: `text()`
- 显示汉字, 可设置 `fontproperties` 参数

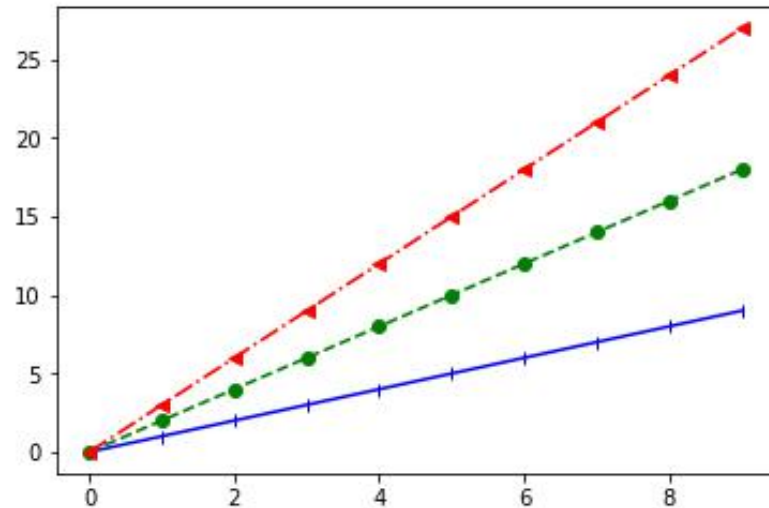
```
x=range(10)
y=[k**2 for k in x]
plt.plot(x,y,'go--',label="Sample")
plt.title("Matplotlib",fontsize=20)
plt.text(4,30,"point",fontsize=20)
plt.xlabel("X轴",fontproperties="SimHei",fontsize=18)
plt.ylabel("Y轴",fontproperties="SimHei",fontsize=18)
plt.legend(loc="upper left")
```





# 多线绘制

- `plot([x], y, [fmt], [x2], y2, [fmt2], ..., **kwargs)`
  - X, x轴数据, 可省略
  - Y, y轴数据, 不可省略
  - fmt 格式控制字符, 可省略
  - 多组数据

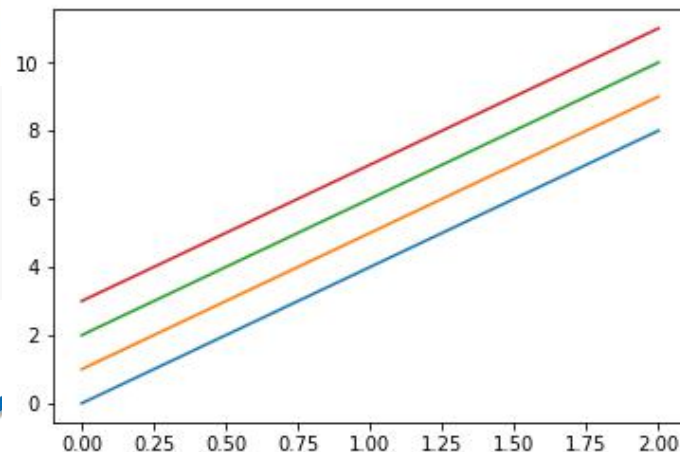


```
x=np.arange(10)
```

```
plt.plot(x,x,'b| - ',x,x*2,'go--',x,x*3,'r<- .')
```

```
d= pd.DataFrame(np.arange(12).reshape(3,4))
```

```
plt.plot(d)
```



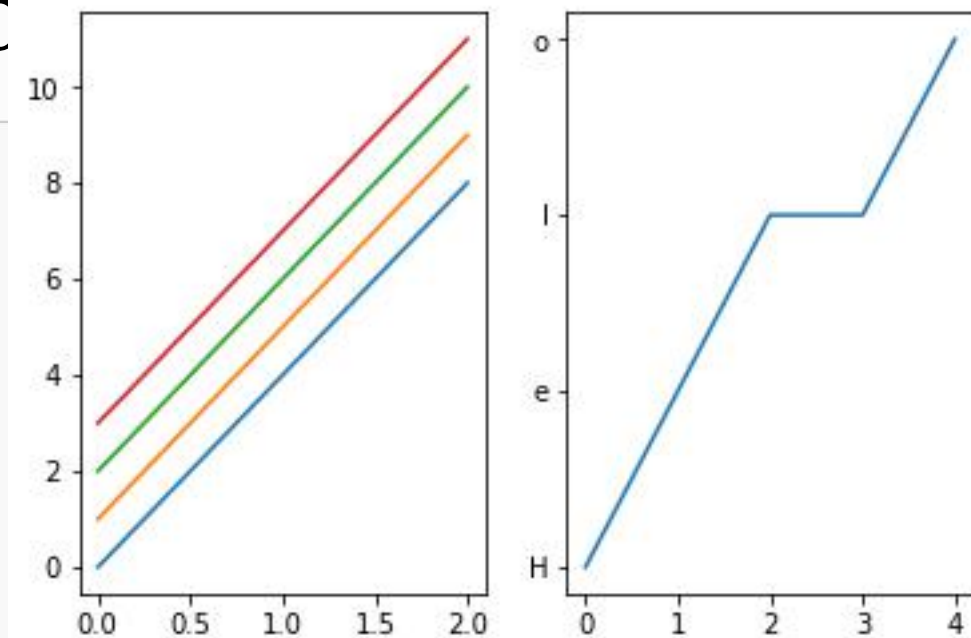




# 多子图绘制subplot

- `plt.subplot(rows, cols, number)`
- 行列和当前正在绘制的编号
- 编号从左到右，从上到下，行优先

```
plt.subplot(1,2,1)
d= pd.DataFrame(np.arange(12).reshape(3,4))
plt.plot(d)
plt.subplot(1,2,2)
s = pd.Series(list("Hello"))
plt.plot(s)
```



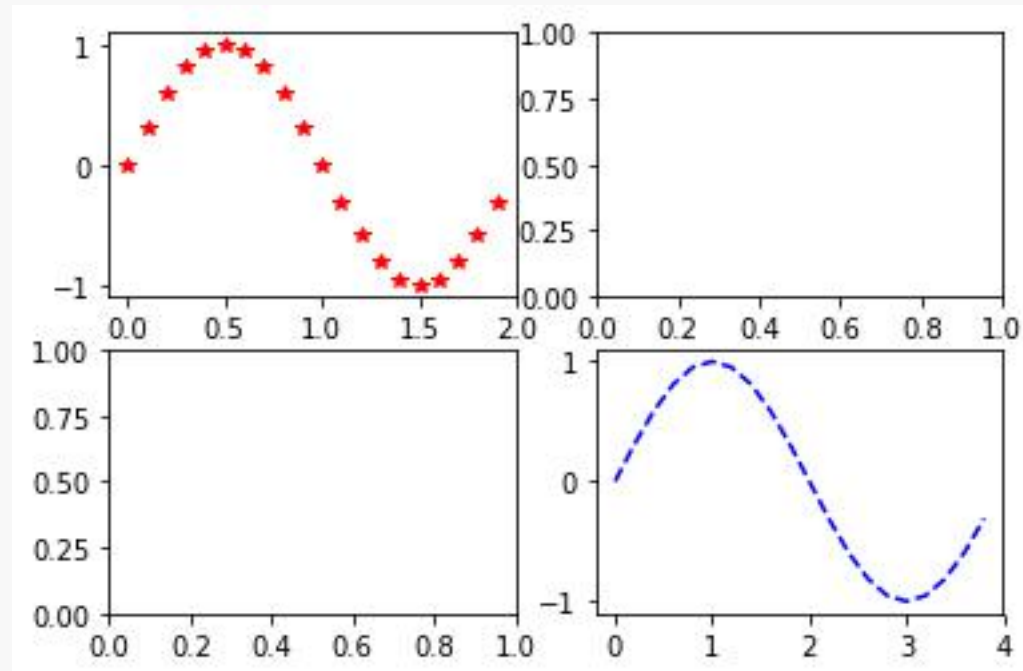


# 多子图绘制subplots

使用subplots进行多图绘制

`fig, ax = subplots(rows, cols)`

```
t=np.arange(0.0,2.0,0.1)
s=np.sin(t*np.pi)
figure,ax=plt.subplots(2,2)
ax[0][0].plot(t,s,'r*')
ax[1][1].plot(t*2,s,'b--')
```







# 不同的绘图风格

- 饼状图、柱状图，散点图……
- 箱线图、直方图、雷达图……

```
labels = 'A', 'B', 'C', 'D'  
fracs = [15, 30.55, 44.44, 10]  
explode = [0, 0.1, 0, 0] # 0.1 凸出部分  
plt.axes(aspect=1) # 轴向比例  
plt.pie(x=fracs, labels=labels, explode=explode,  
        autopct='%3.1f %%',  
        shadow=True, labeldistance=1.1,  
        startangle = 90, pctdistance = 0.6)  
plt.show()
```

