Numpy数据分析模块

一、Numpy

- NumPy 是 Numerical **Python** 的简称，是高性能计算和数据分析的基础包。
- NumPy是Python的一个扩充程序库。支持高级大量的维度数组与矩阵运算，此外也针对数组运算提供大量的数学函 数库。Numpy运算效率极好,是大量机器学习框架的基础库。

包括：

1、一个强大的N维数组对象Array；

2、比较成熟的函数库；

3、用于整合C/C++和Fortran代码的工具包；

4、实用的线性代数、傅里叶变换和随机数生成函数。

使用NumPy，可以有利于以下操作：

1. 数组的算数和逻辑运算。
2. 傅立叶变换和用于图形操作的例程。
3. 与线性代数有关的操作

NumPy + Matplotlib绘图库+ SciPy科学计算包一起使用，这种组合渐渐称为 MatLab的替代方案。

二、NumPy Ndarray 对象

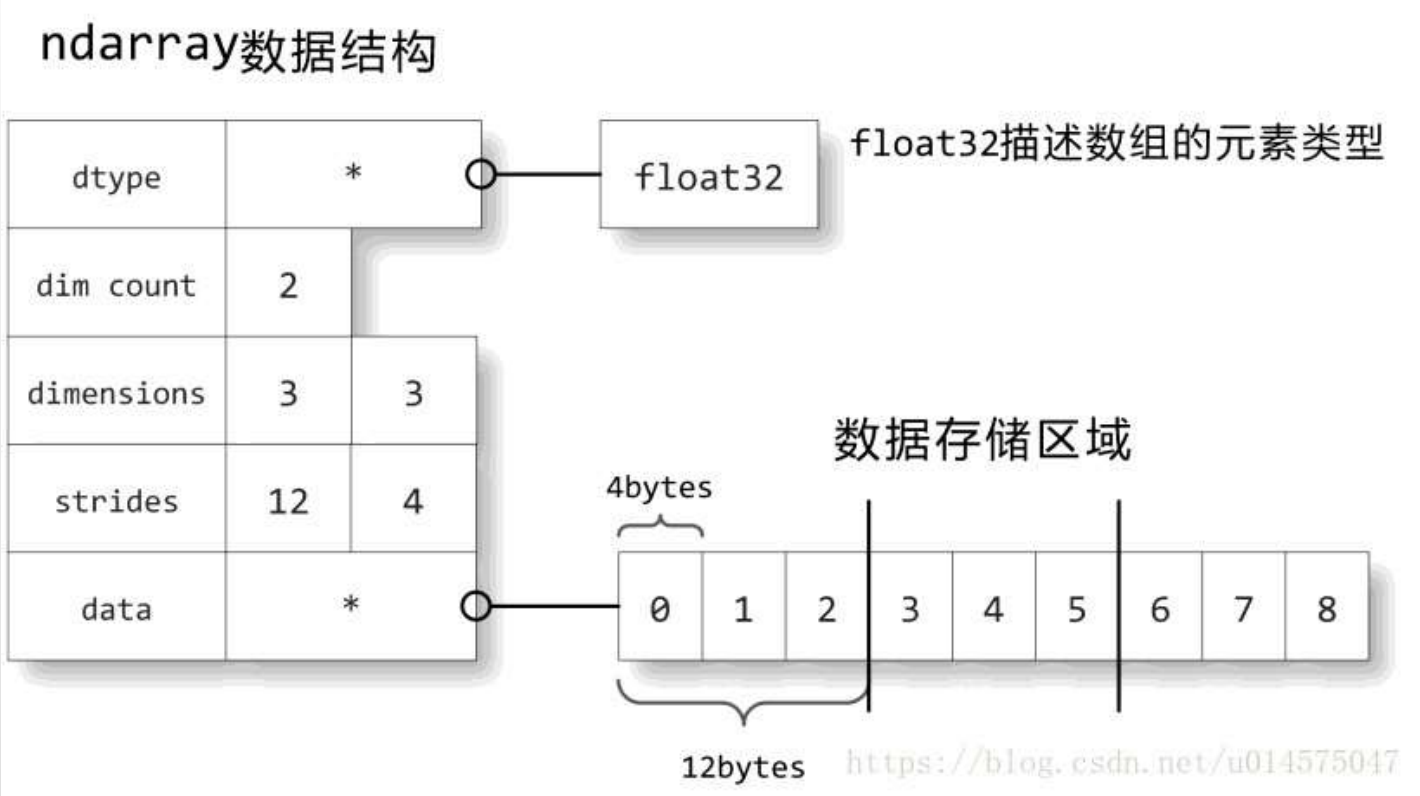
- NumPy 的底层是一个Ndarray结构，该结构可以生成N 维数组对象。
- ndarray 对象是用于存放同类型元素的多维数组。
- ndarray 内部由以下内容组成：

  指针：一个指向数据的指针。 dtype

  ：数据类型。

  Shape：表示各维度大小的元组。

  Stride：一个跨度元组，指为了前进到当前维度下一个元素需要"跨过"的字节数。


ndarray数据结构
dtype
*
float32
float32描述数组的元素类型
dim count
2
dimensions
3
3
strides
12
4
data
*
数据存储区域
4bytes
0
1
2
3
4
5
6
7
8
12bytes
https://blog.csdn.net/u014575047

Ndarray的术语

1、轴（axes）：每一维数组称为一个轴。

2、秩（rank）：秩是描述轴的数量，即数组的维数。一维数组的秩为1，二维数组的秩为2，依此类推。

创建 ndarray

只需调用 NumPy 的 array 函数即可：

numpy.array(object,dtype=None,copy=True,order=None,subok=False,ndmin=0)

参数说明:

名称	描述
object	数组或嵌套的数列
dtype	数组元素的数据类型，可选
copy	对象是否需要复制，可选
order	创建数组的样式，C为行方向，F为列方向
subok	默认返回一个与基类类型一致的数组
ndimin	指定生成数组的最小维度

实验1：创建Ndarray

```
import numpy as np
arr=np.array([[0,1,2],[3,4,5]])
print(arr)
```

```
#显示不同轴的统计结果
print(arr.sum(axis=0))
print(arr.sum(axis=1))
```

- ndarray.shape返回一个元组。这个元组的长度就是维度的数目，即ndim属性(秩)。
- 另外，ndarray.shape 和reshape 函数都可以调整数组大 小。

实验2：调整数组大小

在实验1的基础上，添加代码：

```
reArr = arr.reshape(3,2)
reArr
```

```
array([[0, 1],
       [2, 3],
       [4, 5]])
```

和

```
arr.shape = (3,2)
arr
```

效果是相同的

创建特殊的数组

（1）numpy.empty：

创建一个指定形状（shape）、数据类型（dtype）且未初始化的数组。

例如：numpy.empty(shape, dtype=float, order='C')

参数	描述
shape	数组形状
dtype	数据类型，可选
order	有"C"和"F"两个选项,分别代表，行优先和列优先，在计算机内存中的存储元素的顺序。

（2）numpy.zeros：

创建指定大小的数组，以0填充。

格式：numpy.zeros(shape,dtype=float,order = 'C’)

参数	描述
shape	数组形状
dtype	数据类型，可选
order	'C' 用于C 的行数组，或者 'F' 用于FORTRAN 的列数组

（3）**numpy.ones**：

创建指定形状的数组，数组元素以1来填充。

实验3：创建全0数组

```
y = np.zeros((5,), dtype = np.int)
y
```

（4）能创建序列的函数

——arange函数、linspace函数以及python的range函数

range()函数

函数形式：range(start, stop[, step])，其功能是根据[start,stop)范围以及step设定的步长，生成一个序列。计数不包括stop。

例如：arr1=range(0,5,1)

Numpy .arrange()函数

- 函数形式：arange([start,] stop[, step,], dtype=None)，其功能是根据[start,stop)范围以及step设定的步长，生成一个ndarray。

linspace()函数

- numpy.linspace(start, stop, num=50, endpoint=True, retstep=False, dtype=None)
- scalar是序列的起始点, stop是序列的结束点, num是在[start,stop]范围内生成的样本数。

例如：arr=numpy.linspace(1, 5, 10)

实验4：numpy创建序列

对比

```
import numpy as np
arr=np.arange(1,5,0.2)
arr
```

与

```
arr=np.linspace(1, 5, 10)
arr
```

三、数组的切片、迭代和索引

- ndarray一维数组可以直接索引、切片和迭代，和其它Python序列同样。

二维数组切片操作

- 二维数组的切片操作与一维数组类似，由于多一个轴，所以方括号里有两个值（使用逗号分隔）。可以把逗号的左边和右边当做是一个一维数组，比如：A[0:2,0:2]

实验5：ndarray数组的切片

使用**len(arr)**查看到前面的**arr**数组长度为**10**，在实验**4**的基础上添加两行代码：

```
arr=arr.reshape(5,2)  #这里试试不赋值回去会怎么样?
arr
arr[1:4, 0:3]
```

四、二维数组的的迭代

- 可以使用for循环嵌套，但它通常按照第一条轴对二维数组进行扫描，这里可以使用 apply_along_axis(func,axis,arr)函数设定对每一列或行 进行处理。

另外，NumPy中包含一个迭代器对象numpy.nditer，可以很方便地对数组进行迭代。

实验6：对ndarray数组进行迭代

使用arange()函数创建一个3X4 数组，分别使用for和nditer进行迭代。

```
import numpy as np
a = np.arange(0,60,5)
a = a.reshape(3,4)
print(a)

#两种迭代方式
 for item in a:
    print(item)

for x in np.nditer(a):
     print(x)
```

```
[[ 0  5 10 15]
 [20 25 30 35]
 [40 45 50 55]]
[ 0  5 10 15]
[20 25 30 35]
[40 45 50 55]
0
5
10
15
20
25
30
35
40
45
50
55
```

Pandas数据分析模块

一、Pandas

Pandas （Python Data Analysis Library）是python的一个数据分析包，是基于NumPy的一种工具，为了解决数据分析任务而创建的。

Pandas提供了高效地操作大型数据集所需的工具，是Python高效数据分析的重要因素之一。

Pandas的三种基础数据结构：

1、Series：一维数组，与Numpy中的一维array类似。与Python的数据结构List很相近。不同种数据类型都能保存在Series中。

2、DataFrame：二维数据结构。可以将DataFrame理解为Series的容器。很多数据处理是基于DataFrame结构的。

3、Panel：三维的数组，可以理解为DataFrame的容器。

这些数据结构建在Numpy数组的基础之上，运算速度很快。

实验7：Series一维数组操作

```
#1.一维Series结构
import pandas as pd
s = pd.Series([1,3,5,6])
print(s)
```

接下来修改:

s[1]=80

S

```
s2=pd.Series({'color':1,'size':2,'weight':3})
s2['weight']=200
s2
```

实验8：二维DataFrame数组操作

```
#二维DataFrame结构

import pandas as pd
dict1 = {'col1':[1,2,5,7],'col2':['a','b','c','d']}
df = pd.DataFrame(dict1)
df
```

从数据文件中读取，生成dataframe。

```
import pandas as pd
data = pd.read_csv('dataH.txt',sep = ' ')  #指明分隔符
print(data)
```

切片

```
data[:2]
```

简单过滤

```
data[data['Wt']<60]
```

- Numpy和Pandas都能完成数据分析任务，但求方差的情况比较特殊，比较下面两个例子。

例3：

```
import numpy as np
a = np.arange(0,60,5)
a = a.reshape(3,4)
print(a)
result = np.std(a, axis=0)
print(result)
result = np.std(a, axis=1)
print(result)
```

```
[[ 0  5 10 15]
 [20 25 30 35]
 [40 45 50 55]]
[16.32993162 16.32993162 16.32993162 16.32993162]
[5.59016994 5.59016994 5.59016994]
```

可以看到如下结果：

对同样的数据，如果使用pandas的std函数，运算结果则是不同的：
例4：

```
import numpy as np
import pandas as pd
a = np.arange(0,60,5)
a = a.reshape(3,4)
df = pd.DataFrame(a)
print(df)
print('_____')
print(df.std())
```

```
    0   1   2   3
0   0   5  10  15
1  20  25  30  35
2  40  45  50  55
-----------------
0    20.0
1    20.0
2    20.0
3    20.0
dtype: float64
```

- 原因是numpy 的std()函数 和 pandas 的std() 函数的默认参数ddof是不同的。ddof参数表示标准偏差类型，numpy中ddof默认是0，计算的是总体标准偏差；在pandas中ddof的值默认是1，计算的是样本标准偏差*。
- 注*：标准差也被称为标准偏差(Standard　Deviation)，统计学名 词，描述各数据偏离平均数的距离（离均差）的平均数。标准差能反映一个数据集的离散程度，标准偏差越小，这些值偏离平均值就越少。

Matplotlib绘图库

一、 Matplotlib

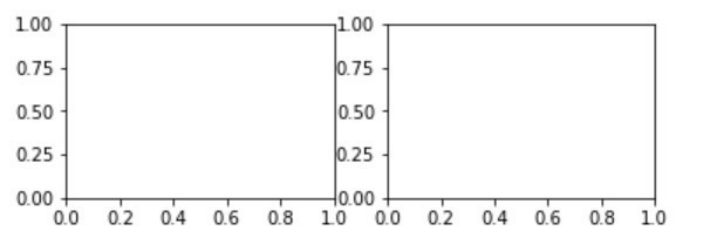
- Matplotlib是一个Python的2D绘图库，它可以生成跨平台的出版质量级别的图形。
- Matplotlib可用于Python、 IPython、 Jupyter notebook、 web应用 服务器等。
- 使用Matplotlib，能让复杂的工作变得容易，可以制图、可以生成直方图、条形图、散点图、曲线图等。
- 它提供了很多参数，可以通过参数控制样式、属性等。

1、 Figure和Subplot

matplotlib的图像都位于Figure对象中，可以用plt.figure创建一个新的Figure（空Figure不能绘图），添加plot用add_subplot。

实验9：绘制简单图表

```
import matplotlib.pyplot as plt
fig=plt.figure()
ax1=fig.add_subplot(2,2,1)
ax2=fig.add_subplot(2,2,2) #修改成（2,2,3)试试
```



2、 plot绘图

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

参数:

x:x轴数据，列表或数组，可选。

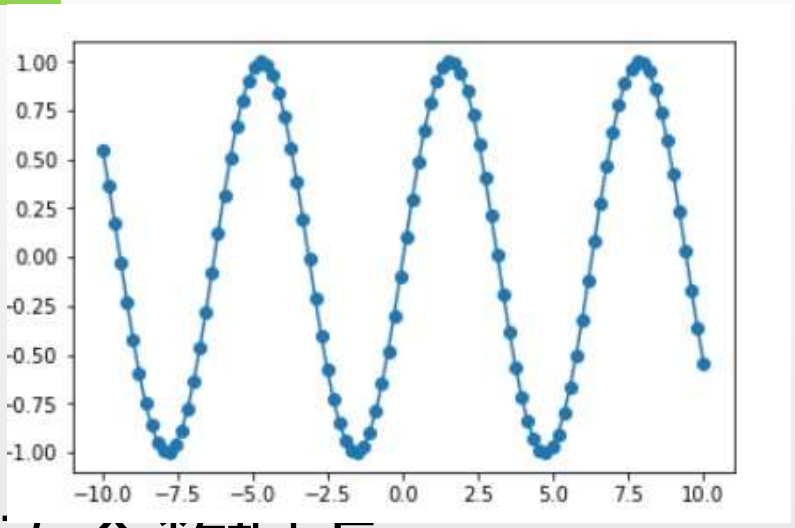
y:y轴数据，列表或数组。

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)。

注：当绘制多条曲线时，各条曲线的x不能省略。

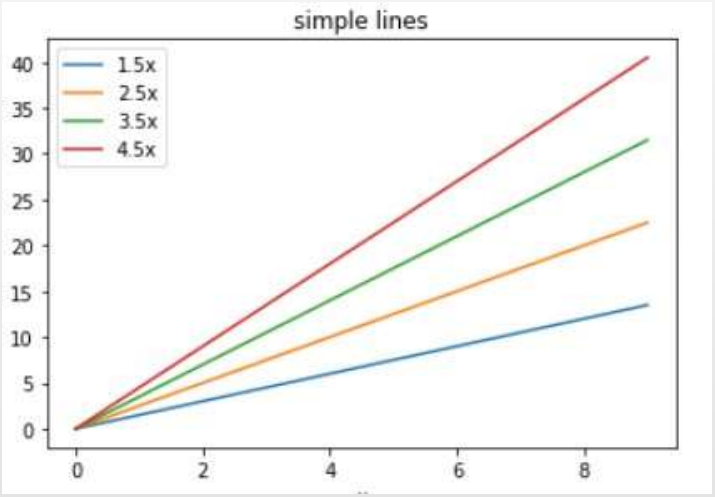
实验10：绘制正弦曲线图形




```
#制sin(x)函数图形

import numpy as np
import matplotlib.pyplot as plt


x = np.linspace(-10, 10, 100)  #列举出一百个数据点
y = np.sin(x)     #计算出对应的y
plt.plot(x, y, marker="o")
```

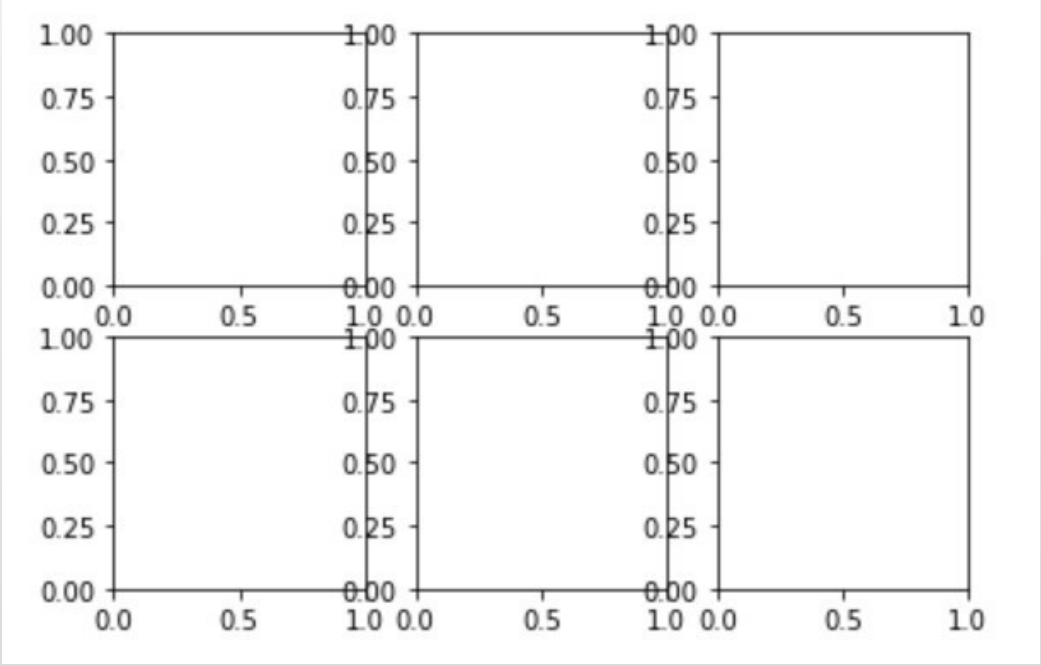
实验11：绘制简单直线，设置标题、坐标轴、图例。

```
import matplotlib.pyplot as plt
import numpy as np
a = np.arange(10)
plt.xlabel('x')
plt.ylabel('y')
plt.plot(a,a*1.5,a,a*2.5,a,a*3.5,a,a*4.5)
plt.legend(['1.5x','2.5x','3.5x','4.5x'])
plt.title('simple lines')
plt.show()
```



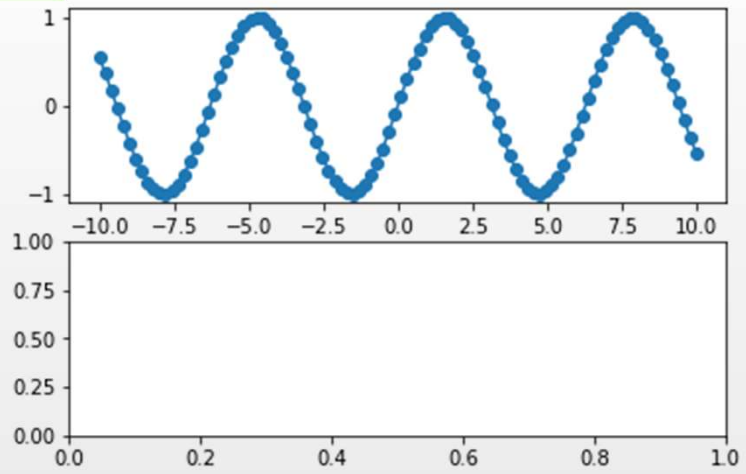
实验12：多个plot的绘制

```
fig,axes=plt.subplots(2,3)
axes
```

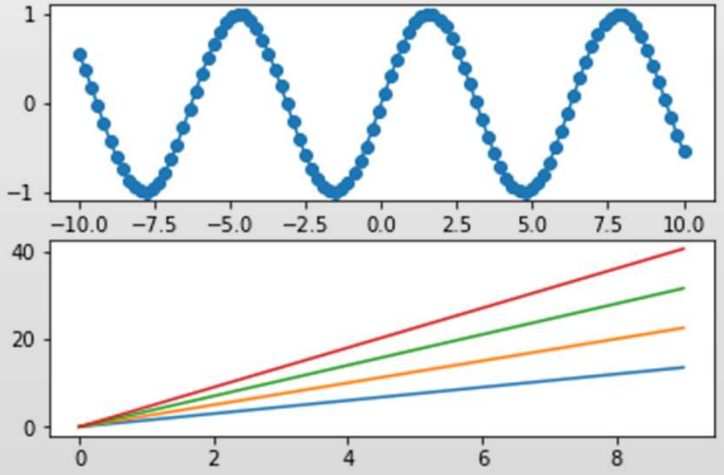


实验13：在多个plot上绘图

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

fig,axes=plt.subplots(2,1)


plt.subplot(2,1,1)

x = np.linspace(-10, 10, 100)  #列举出一百个数据点
y = np.sin(x)    #计算出对应的y
plt.plot(x, y, marker="o")
```

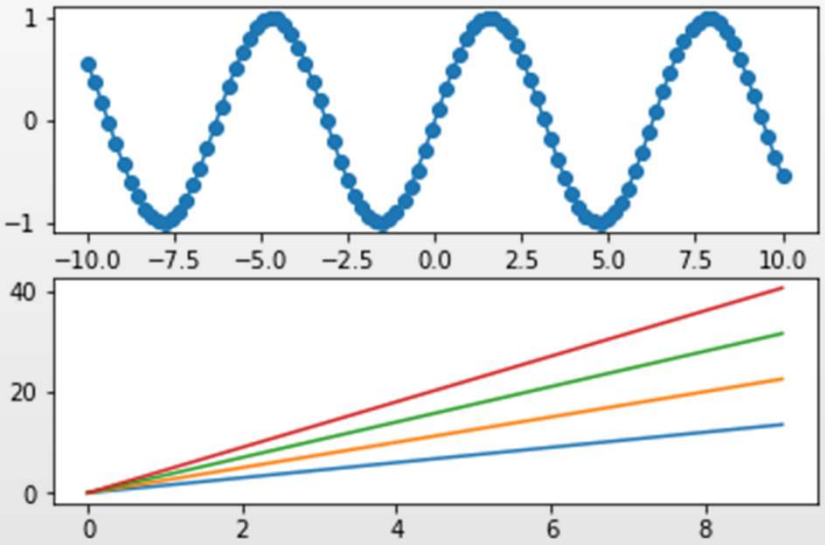


请继续修改程序，做成右边这张图的效果->



实验13参考

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

fig,axes=plt.subplots(2,1)

plt.subplot(2,1,1)
x = np.linspace(-10, 10, 100)  #列举出一百个数据点
y = np.sin(x)    #计算出对应的y
plt.plot(x, y, marker="o")
```



plt.subplot(2,1,2)

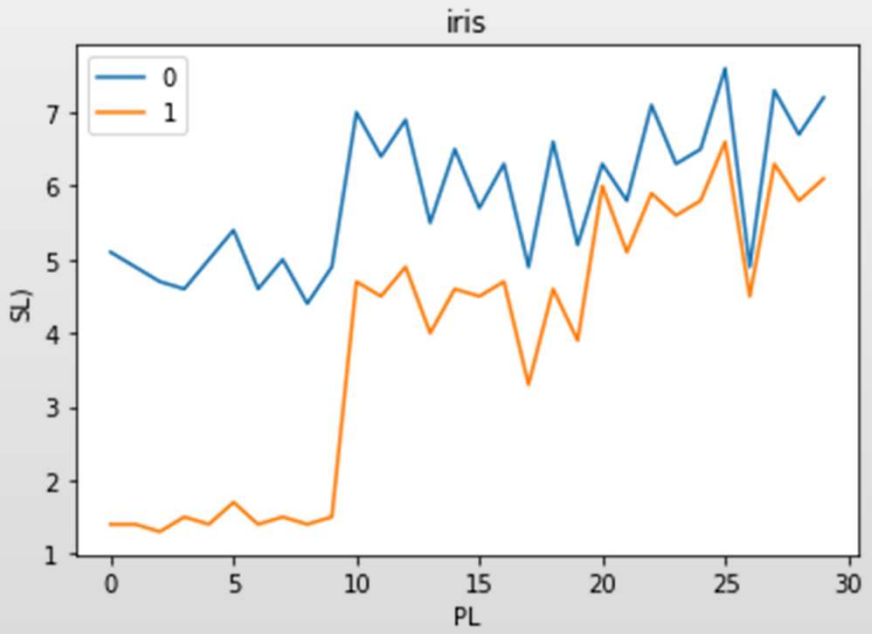
a = np.arange(10)

plt.plot(a,a*1.5,a,a*2.5,a,a*3.5,a,a*4.5)

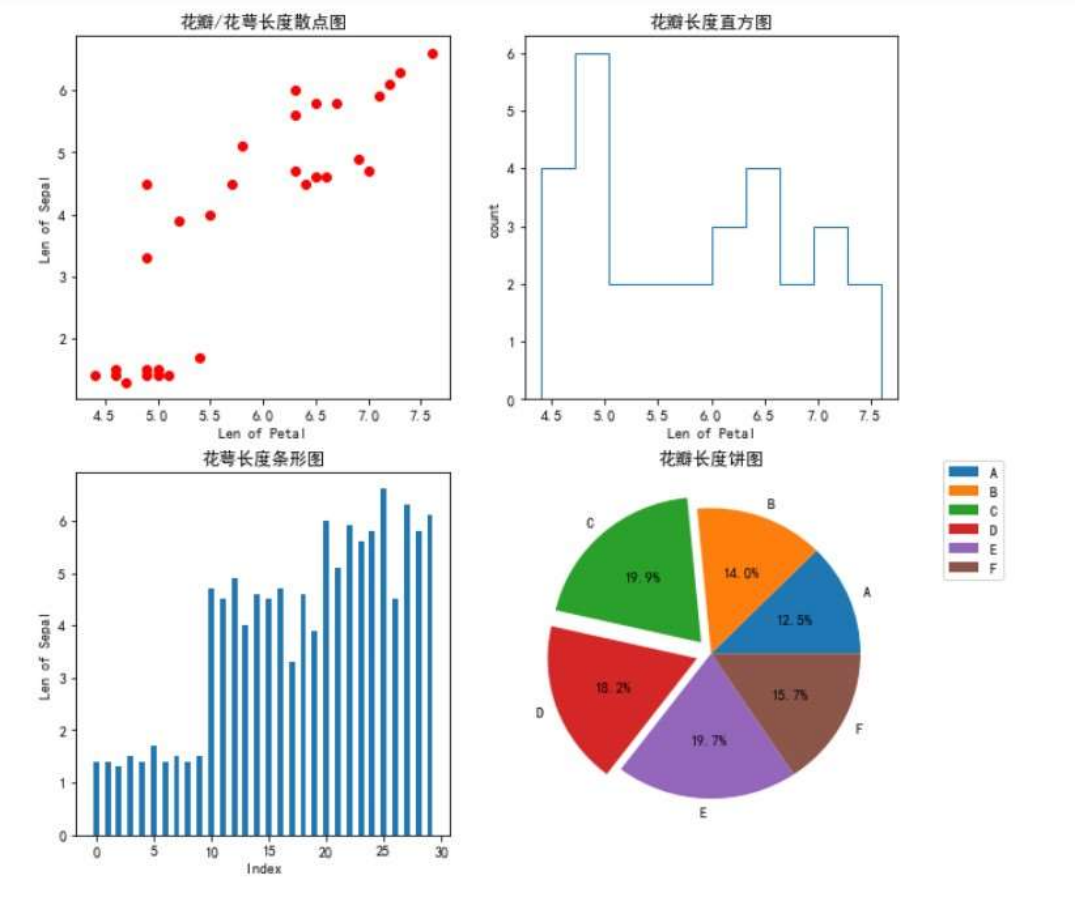
- pandas中内嵌了基于matplotlib的绘图函数。
- 例如，Series和DataFrame都包含生成图表的plot方法，默认情况下，它们生成的是线型图。

实验14 使用dataframe的绘图函数

```
import pandas as pd
data = pd.read_csv('iris.txt',',',header=None)
df = pd.DataFrame(data)
ax=df.plot(title='iris')
ax.set_xlabel('PL')   #设置x轴标签
ax.set_ylabel('SL)')  #设置y轴标签
```



小练习1：观察下面的图表，思考绘制这样一幅图的方法。



参考：code 小练习1.txt中的代码

HOMEWORK1

研究人员在2014年，分别在中国与日本的四个城市对将近2万名7-18岁两国儿童青少年，进行了相关测试。平均年龄/身高数据如下：

编写程序，实现如下功能：

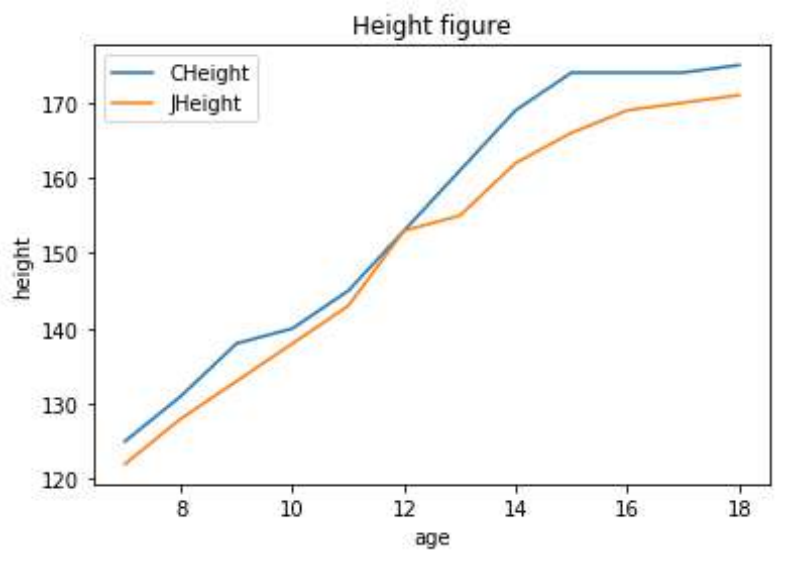
1、从中日7-18岁男生平均身高“avgHgt.csv”文件中读取身高数据。

2、把数据绘制成曲线图。

3、把数据写入文本文件avgHgt.txt。

（文本文件中的数据要注意换行）

年龄	中国男孩身高（c	
7	125	
8	131	
9	138	
10	140	
11	145	
12	153	
13	161	
14	169	
15	174	
16	174	169
17	174	170
18	175	171



OpenCV模块

OpenCV视觉模块

- OpenCV是由Gary Bradsky在1999年英特尔启动的项目，2000年发布了第一个版本。可以在不同的平台上使用，包括Windows、Linux、OS X、Android和iOS。基于CUDA和OpenCL的GPU操作接口也于2010年9月开始实现。
- OpenCV Python是一个用于解决计算机视觉问题的Python库，是用基于C++实现的OpenCV构成的Python包。
- OpenCV Python与NUMPY兼容，数据都被转换成NUMPY数据结构，这使得OpenCV
  更容易与其他库（如SCIPY和Matplotlib）集成。
- OpenCV的版本不断发展，OpenCV 3.x后，用的较多的是cv2包。

一、OpenCV窗口操作

1.imshow()

imshow函数在指定的窗口中显示图像，窗口自动调整为图像大小。格式为imshow(string winName,Array InputData)，第一个参数winName是窗口名 称，InputData为输入的图像。如果创建多个窗口，需要具有不同的名称。

2. destroyAllWindows()与DestroyWindow(string winName)

两个函数都可以卸载当前窗口。区别在于，函数destroyAllWindows()卸载全部窗口，而函数DestroyWindow()卸载参数winName指定的窗口。

3. waitKey(int delay=0)

等待用户按键的函数waitKey，其中的delay是延迟的时间，单位为毫秒。在等待时间内如果检测到键盘动作，则返回按键的ASCII码。如果没有按下 任何键，则返回-1。默认为0，一直等待键盘输入。

二.OpenCV处理图像

基本图像处理的函数包括：imread、 imwrite、 split、 merge等。

函数imread()从指定的文件加载图像并返回一个矩阵。支持bitmap位图、JPEG文件、png图形等多种图像格式。

imread函数格式：

Imread(filename,int flags) 参数：

filename:文件名

flag:图像色彩模式，可取ImreadModes枚举列表中的值。默认为IMREAD_COLOR。

实验14：OpenCV读取并处理图像文件

```
import cv2
img = cv2.imread('img.jpg',0)  #转变为灰度图
cv2.imshow('image',img)
k = cv2.waitKey(0)
if k == 27:         #按Esc键直接退出
    cv2.destroyAllWindows()
```

接下来增加下面几句

```
elif k == ord('s'): # 按's'键先保存灰度图，再退出
    cv2.imwrite('result.png',img)
    cv2.destroyAllWindows()
```

三. OpenCV捕获摄像头图像

（1）打开摄像头捕获图像

可以使用cv2.VideoCapture()来截取摄像头的当前帧。

- VideoCapture(cam)：打开摄像头并捕获视频。参数cam为0时，表示从摄像头直接获取；也可以读取视频文件，这时参数为视频文件的路径。
- read()：读取视频的帧。返回值有两个——ret,frame。ret 是布尔值，读取正确返回True，如果读取为空（例如是视频 文件结尾）则返回False。frame是一帧图像，是三维数组（长、宽、颜色通道）。

实验15：OpenCV捕获摄像头图像

```
#打开摄像头捕获图像
import cv2
cap  = cv2.VideoCapture(0)
while(True):
    ret, frame = cap.read()
    cv2.imshow(u"Capture", frame)
    key = cv2.waitKey(1)
    if key & 0xff == ord('q') or key == 27:
        print(frame.shape,ret)
        break
cap.release()
cv2.destroyAllWindows()
exit()  #在摄像头关闭异常时强制退出
```

Wordcloud词云

一. wordcloud展示高频文字数据

- 词云图worldcloud，也叫文字云，是对文本中出现频率较高的“关键词”数据给予视觉差异化的展现。词云图突出展示高频高质的数据信息，凸显出数据中所包含的文本的主旨。可以用于快速显示数据中的文本频率。

实验16：使用词云展示高频词汇

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt

f = open(r'texten.txt','r').read()

#生成词云
wordcloud=WordCloud(background_color="white",width=1000,height=860,margin=2 ).generate(f)

#显示词云图片
plt.imshow(wordcloud)
plt.axis("off")
plt.show()

#保存图片
wordcloud.to_file('test.png')
```

修改上面的程序，设置词云的背景图片

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt from
imageio import imread

text = open('song.txt','r').read()
#读入背景图片
bg_pic = imread('notation.png')
#生成词云 stopwd=['is','a','the','to','of','in','on','at','and']
wdcd=WordCloud(mask=bg_pic,background_color='white',scale=1.5,stopwords=sto
wdcd=wdcd.generate(text)
plt.imshow(wdcd)
plt.axis('off') plt.show()
wdcd.to_file('pic.jpg')
```

课堂练习2

选择网络上的一篇名人的英文演讲稿，制作词云

要求：

1、设置常用的停用词

2、背景图片自定

3、分组作业：查找资料，思考并讨论制作中文词云的过程。