Advanced Data Processing and Visualization of Python

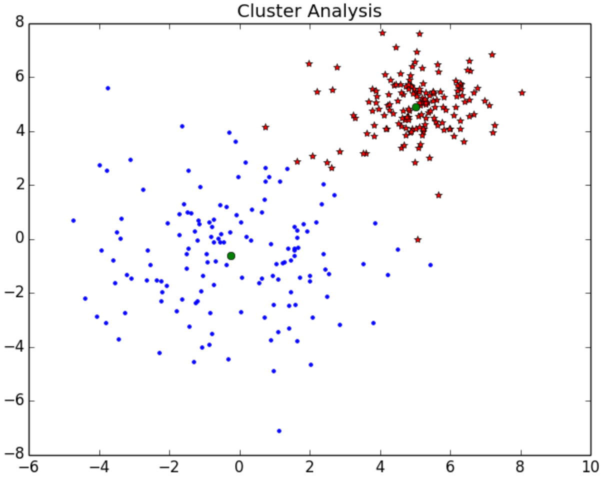
Python高级数据处理与可视化

Department of Computer Science and Technology
Department of University Basic Computer Teaching

1

聚类分析

聚类



- 聚类分析(cluster analysis)

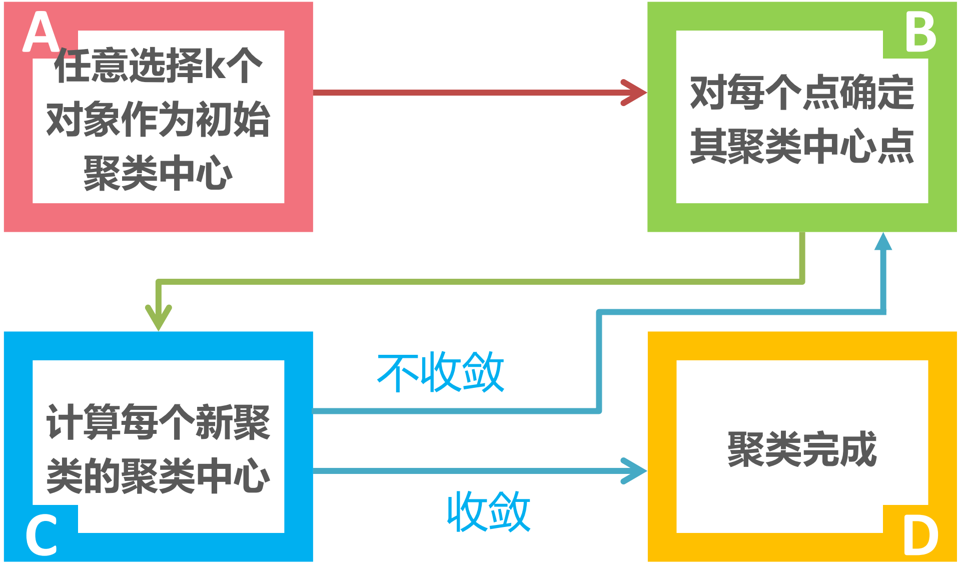
以相似性为基础把相似的对象通过静态分类的方法分成不同的组别或者更多的子集

– 特性

  - 基于相似性
  - 有多个聚类中心

K-MEANS

K-均值算法表示以空间中k个点为中心进行聚类，对最靠近他们的对象归类。



一个日常小例子

	高数	英语	Python	音乐
小明	88	64	96	85
大明	92	99	95	94
小朋	91	87	99	95
大朋	78	99	97	81
小萌	88	78	98	84
大萌	100	95	100	92

```
# Filename: kmeansStu.py
from pylab import *
from scipy.cluster.vq import *
list1 = [88,74,96,85]
list2 = [92,99,95,94]
list3 = [91,87,99,95]
list4 = [78,99,97,81]
list5 = [88,78,98,84]
list6 = [100,95,100,92]
data = vstack((list1,list2,list3,list4,list5,list6))
centroids,_ = kmeans(data,2)
result,_= vq(data,centroids)
print result
```

scipy.cluster.vq更新后其参数只支持float和double，所以此处的数字都要+.0改成浮点数形式

```
Output:
[0 1 1 1 0 1]
```

另一个例子

按相邻两天的收盘价涨跌规律对2014年第3季度（7月-9月）构成道琼斯工业指数的30只股票聚类

File

```
# Filename: kmeansDJI.py
listDji = ['AXP','BA','CAT','CSCO',…, 'VZ','WMT','XOM']
quotes = [ [0 for col in range(90)] for row in range(30)]
listTemp = [ [0 for col in range(90)] for row in range(30)]
for i in range(30):
    quotes[i] = quotes_historical_yahoo_ochl(listDji[i], start, end)
listTemp[i][j] = 1  or -1  # 1 if the latter is larger than former, otherwise the result is  -1
data = vstack(listTemp)
centroids,_ = kmeans(data,4)
result,_= vq(data,centroids)
```

同前一页，此处
需要改成浮点数

另一个例子

```
Output:
[0 3 3 2 0 3 0 1 1 3 2 2 0 1 2 0 1 2 2 1 1 3 2 1 3 0 1 2 0 0]
```

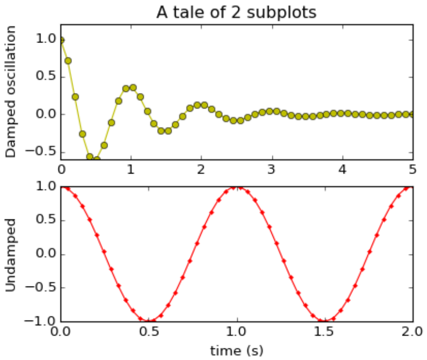
第0类	AXP,CVX,DIS,JNJ,MCD,UTX,WMT,XOM
第1类	GE,GS,JPM,MMM,NKE,PFE,TRV,V
第2类	CSCO,IBM,INTC,KO,MRK,MSFT,T,VZ
第3类	BA,CAT,DD,HD,PG,UNH

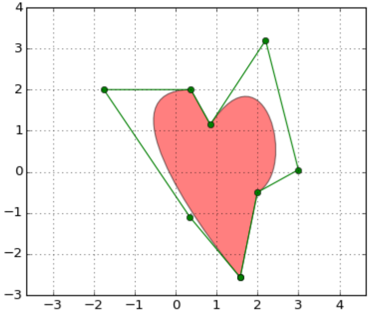
用Python玩转数据

2

MATPLOTLIB 绘图基础

Matplotlib绘图





- ## Matplotlib绘图

  最著名Python绘图库，主要用于二维绘图

  – 画图质量高

  – 方便快捷的绘图模块

    - 绘图API——pyplot模块
    - 集成库——pylab模块（包含NumPy和pyplot中的常用函数）

数据源

可口可乐公司近一年来股票收盘价的月平均价

Source

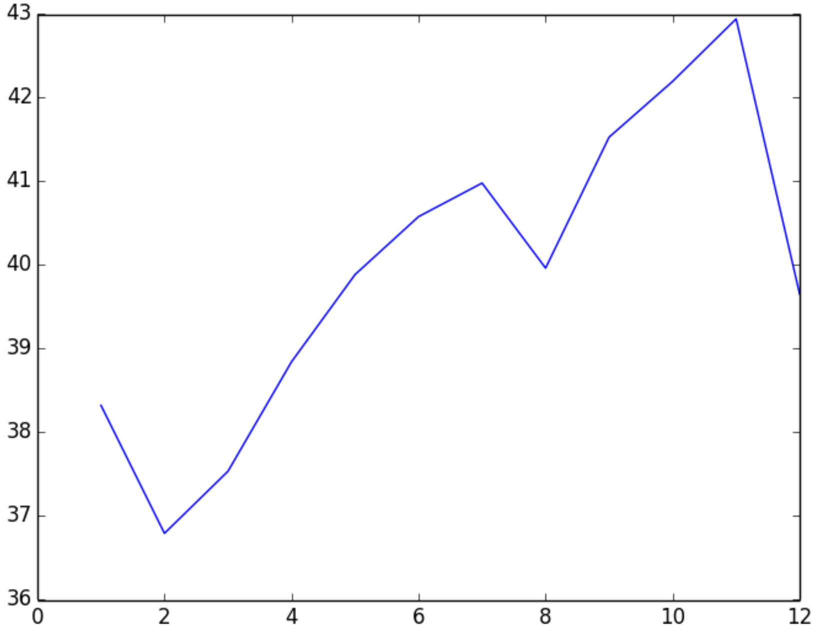
```
>>> closeMeansKO = tempkodf.groupby('month').mean().close
>>> closeMeansKO
month
1       38.589524
2       37.047368
3       37.795238
…
10      42.488261
11      41.429500
12      39.201905
Name: close, dtype: float64
```

折线图

? 将可口可乐公司近一年来股票收盘价的月平均价绘制成折线图

File

```
# Filename: closeMeansKO.py
import matplotlib.pyplot as plt
…
listKO = []
for i in range(1,13):
    listKO.append(closeMeansKO[i])
listKOIndex = closeMeansKO.index
plt.plot(listKOIndex,listKO)
plt.show()
```

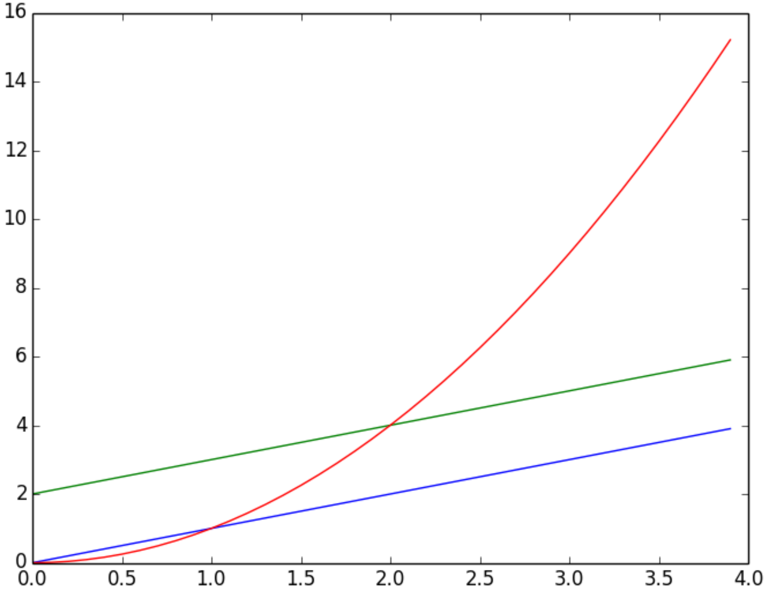


折线图

NumPy数组也可以作为Matplotlib的参数

Source

```
>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>> t=np.arange(0.,4.,0.1)
>>> plt.plot(t,t,t,t+2,t,t**2)
>>> plt.show()
```

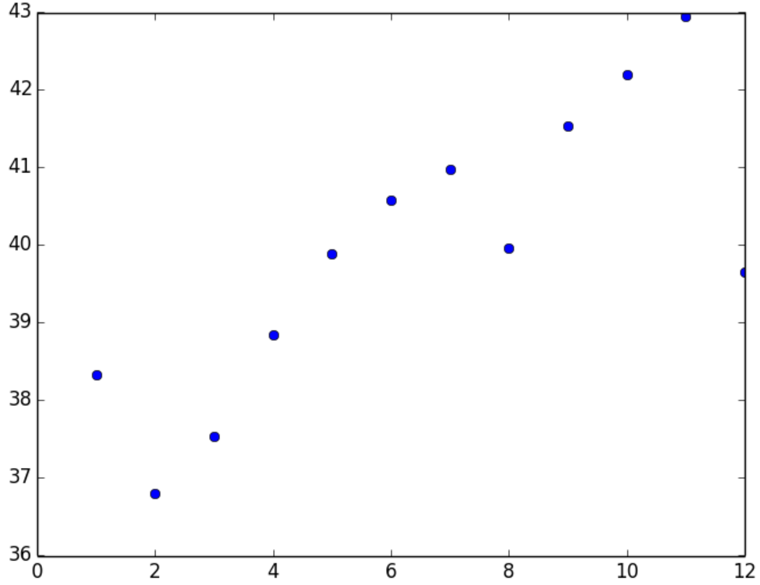


散点图

? 将可口可乐公司近一年来股票收盘价的月平均价绘制成散点图

```
plt.plot(listKOIndex,listKO)
```

```
plt.plot(listKOIndex,listKO, 'o')
```

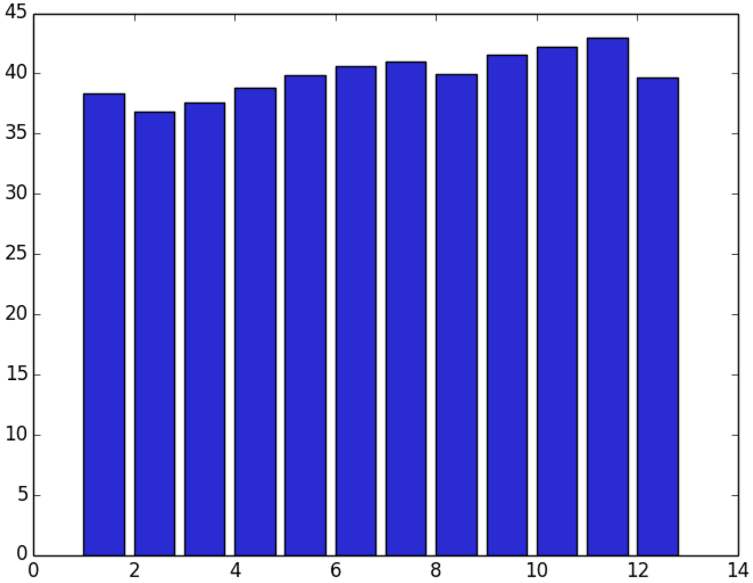


柱状图

将可口可乐公司近一年来股票收盘价的月平均价绘制成柱状图

```
plt.plot(listKOIndex,listKO)
```

```
plt.bar(listKOIndex,listKO)
```

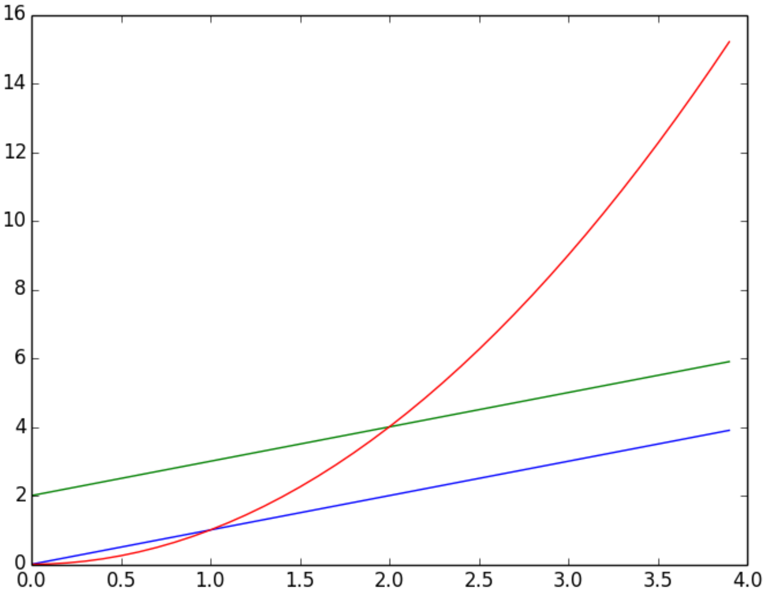


pylab绘图

numpy数组也可以作为matplotlib的参数

Source

```
>>> import numpy as np
>>> import pylab as pl
>>> t=np.arange(0.,4.,0.1)
>>> pl.plot(t,t,t,t+2,t,t**2)
>>> pl.show()
```



3

用Python玩转数据

Matplotlib图像属性控制

Matplotlib属性

……

文字和字体属性

坐标轴和网格属性

子图(axes)

子区(subplots)

色彩和样式

线宽

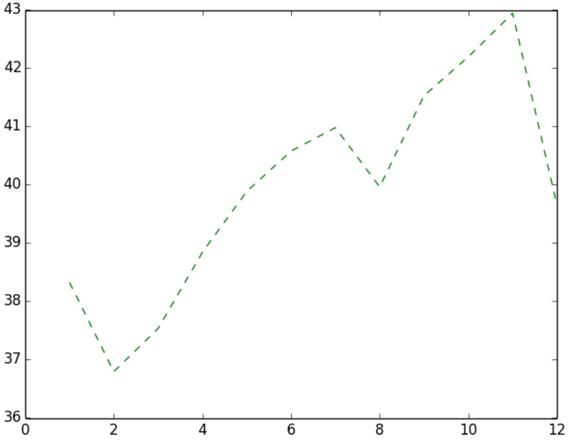
每英寸点数

图像大小

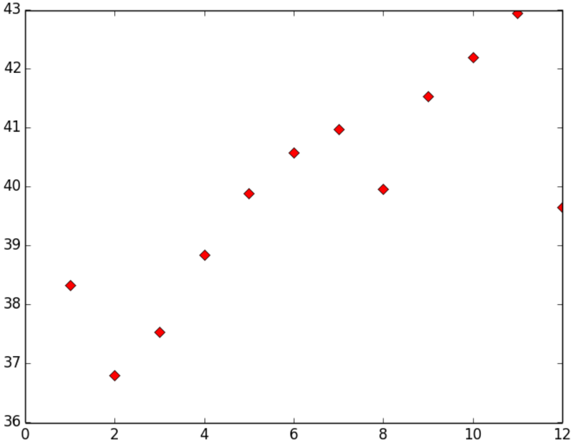
Matplotlib可以控制的默认属性

色彩和样式

绘图颜色和线条类型和样式可以更改吗？



plt.plot(listKOIndex,listKO, 'g--')



plt.plot(listKOIndex,listKO, ' rD')

色彩和样式

符号	颜色
b	blue
g	green
r	red
c	cyan
m	magenta
Y	yellow
k	black
w	white

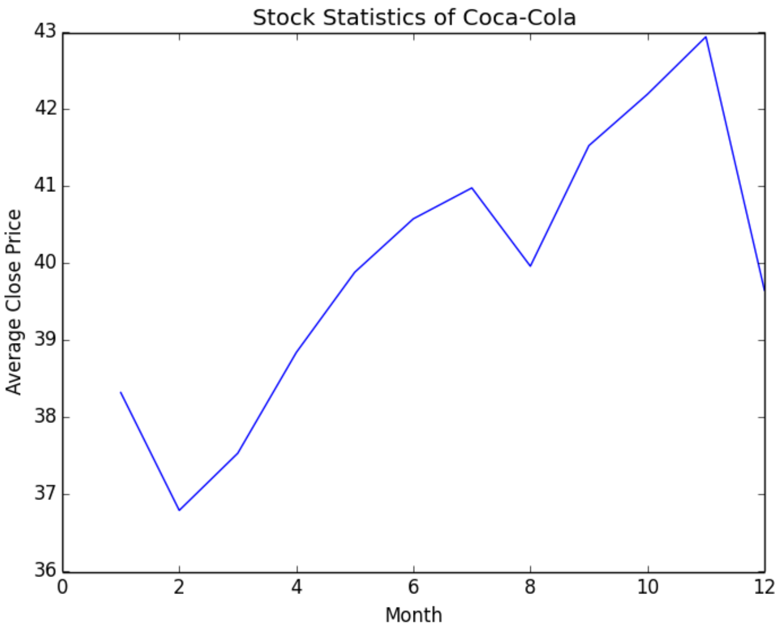
线型	描述
'-'	solid
'--'	dashed
'-.'	dash_dot
':'	dotted
'None'	draw nothing
' '	draw nothing
''	draw nothing

标记	描述
"o"	circle
"v"	triangle_down
"s"	square
"p"	pentagon
"*"	star
"h"	hexagon1
"+"	plus
"D"	diamond
…	…

文字

加标题：图、横轴和纵轴

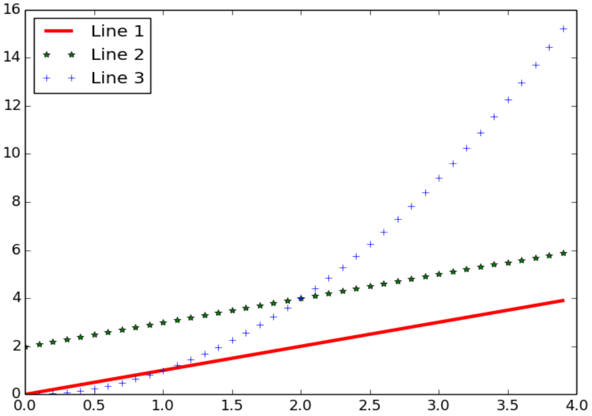
```
# Filename: closeMeansKO.py
import matplotlib.pyplot as plt
…
listKO = []
for i in range(1,13):
    listKO.append(closeMeansKO[i])
listKOIndex = closeMeansKO.index
plt.plot(listKOIndex,listKO)
plt.title('Stock Statistics of Coca-Cola')
plt.xlabel('Month')
plt.ylabel('Average Close Price')
plt.show()
```



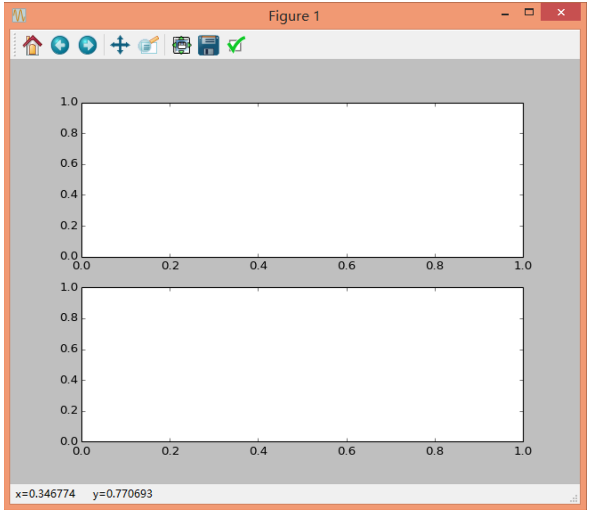
其他属性

File

```
# Filename: multilines.py
import pylab as pl
import numpy as np
pl.figure(figsize=(8,6),dpi=100)
t=np.arange(0.,4.,0.1)
pl.plot(t,t,color='red',linestyle='-',linewidth=3,label='Line 1')
pl.plot(t,t+2,color='green',linestyle='',marker='*',linewidth=3,label='Line 2')
pl.plot(t,t**2,color='blue',linestyle='',marker='+',linewidth=3,label='Line 3')
pl.legend(loc='upper left')
```

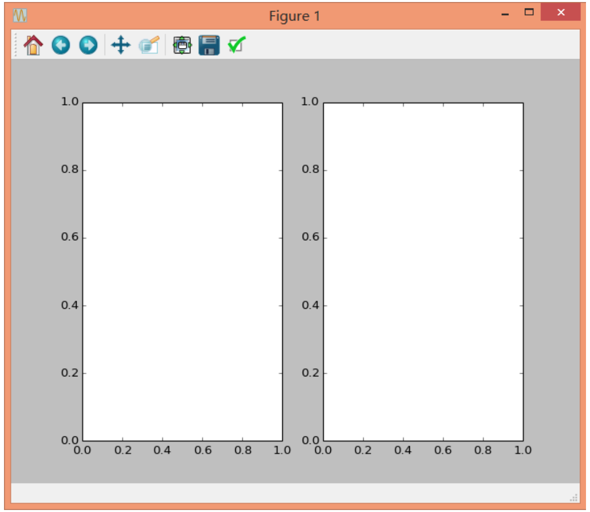


子图-subplots

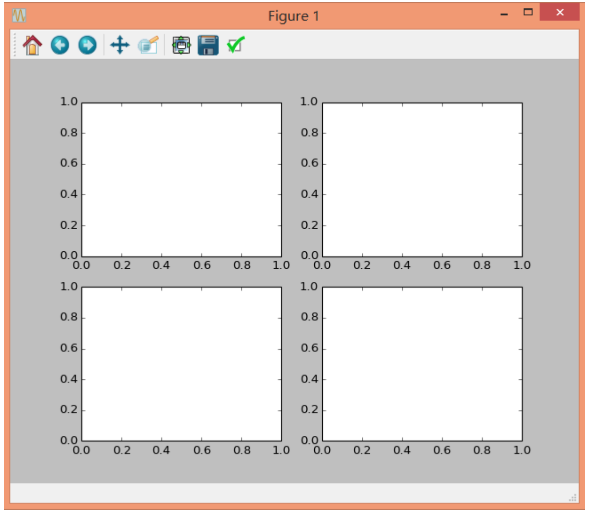


x=0.346774 y=0.770693

plt.subplot(211)
plt.subplot(212)



plt.subplot(121)
plt.subplot(122)



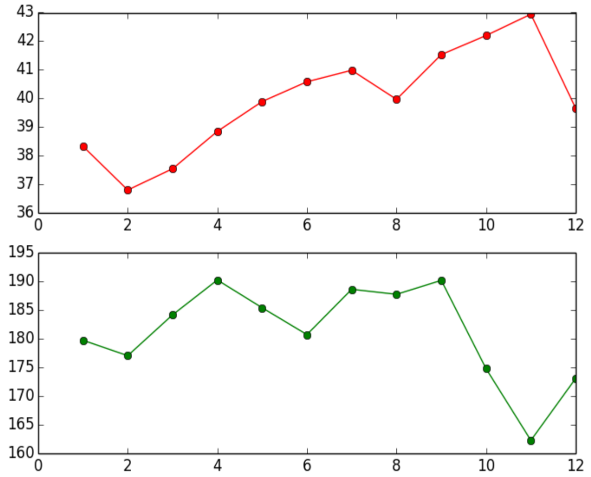
plt.subplot(221)
plt.subplot(222)
plt.subplot(223)
plt.subplot(224)

子图-subplots

将可口可乐公司和IBM公司近一年来股票收盘价的月平均价绘制在一张图中

Source

```
… #The data of Coca-Cola and IBM is ready
>>> subplot(211)
<matplotlib.axes.AxesSubplot object at 0x08B90CD0>
>>> plt.plot(listKOIndex,listKO,color='r',marker='o')
[<matplotlib.lines.Line2D object at 0x04BA5310>]
>>> subplot(212)
<matplotlib.axes.AxesSubplot object at 0x08B90110>
>>> plt.plot(listIBMIndex,listIBM,color='green',marker='o')
[<matplotlib.lines.Line2D object at 0x08917830>]
```

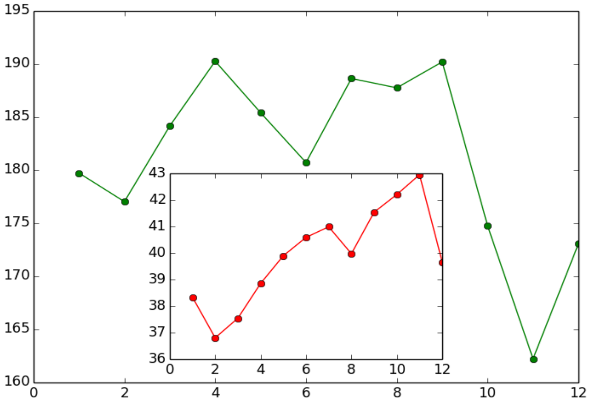


子图-axes

将可口可乐公司和IBM公司近一年来股票收盘价的月平均价绘制在一张图中

Source

```
… #The data of Coca-Cola and IBM is ready
>>> plt.axes([.1,.1,0.8,0.8])
<matplotlib.axes.Axes object at 0x08926210>
>>> plt.plot(listIBMIndex,listIBM,color='green',marker='o')
[<matplotlib.lines.Line2D object at 0x0887EF70>]
>>> plt.axes([.3,.15,0.4,0.4])
<matplotlib.axes.Axes object at 0x08E0C690>
>>> plt.plot(listKOIndex,listKO,color='r',marker='o')
[<matplotlib.lines.Line2D object at 0x08AF3910>]
```



axes([left,bottom,width,height])
参数范围为(0,1)

4

用Python玩转数据

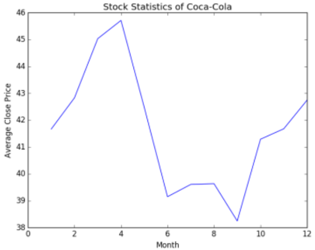
PANDAS作图

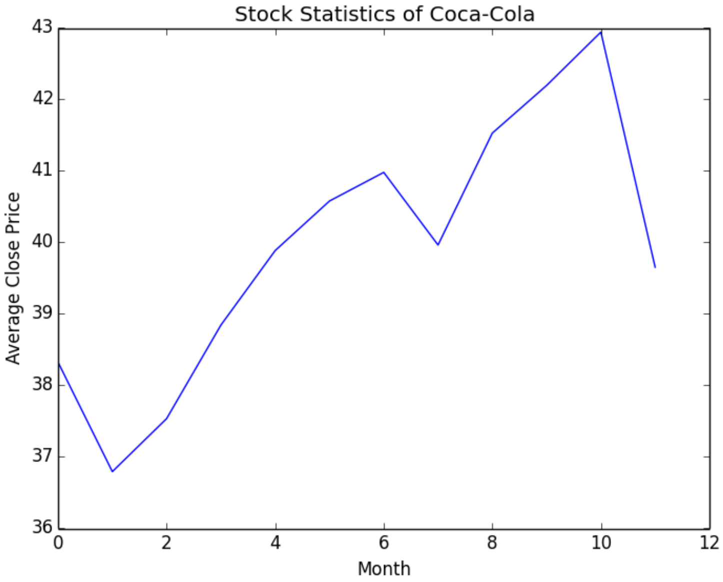
Python实例

Source

```
>>> plt.plot(closeMeansKO)
>>> plt.title('Stock Statistics of Coca-Cola')
>>> plt.xlabel('Month')
>>> plt.ylabel('Average Close Price')
>>> plt.show()
```

新版pyplot已经修正了此问题，现如右边小图

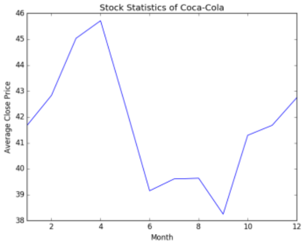




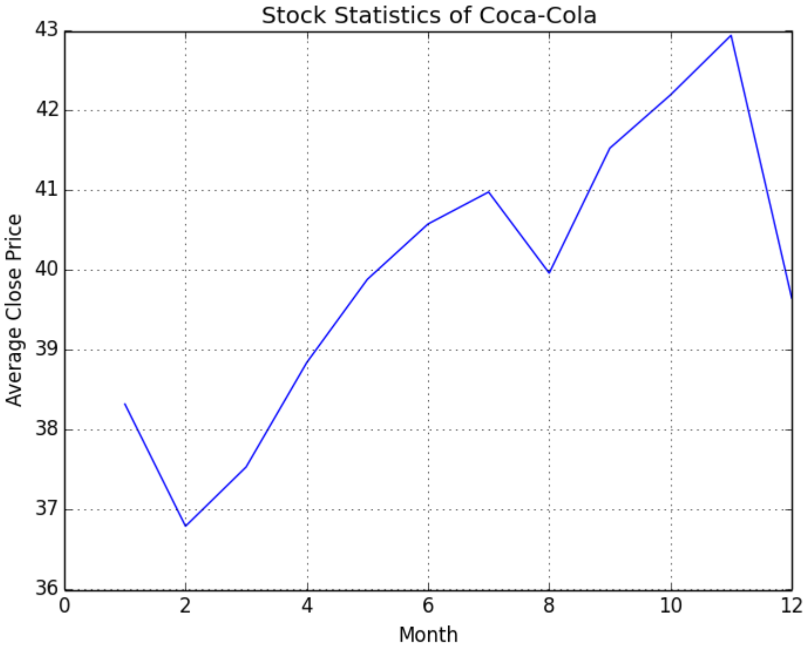
pandas绘图

Source

新版
pandas效
果如右图



```
>>> import pandas as pd
>>> closeMeansKO.plot()
>>> plt.title('Stock Statistics of Coca-Cola')
>>> plt.xlabel('Month')
>>> plt.ylabel('Average Close Price')
>>> plt.show()
```

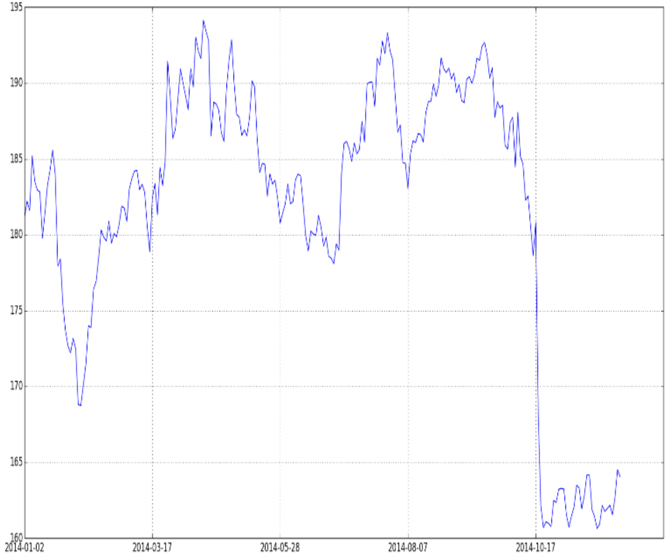


pandas绘图

绘制IBM公司2014年一年的股票收盘价折线图

File

```
# Filename: quotesdfbar.py
...
start = datetime(2014,1,1)
end = datetime(2014,12,31)
quotes = quotes_historical_yahoo_ochl('IBM', start, end)
fields = ['date','open','close','high','low','volume']
...
quotesdfIBM = pd.DataFrame(quotesIBM, index = list1, columns = fields)
quotesdfIBM = quotesdfIBM.drop(['date'], axis = 1)
quotesdfIBM.close.plot()
```



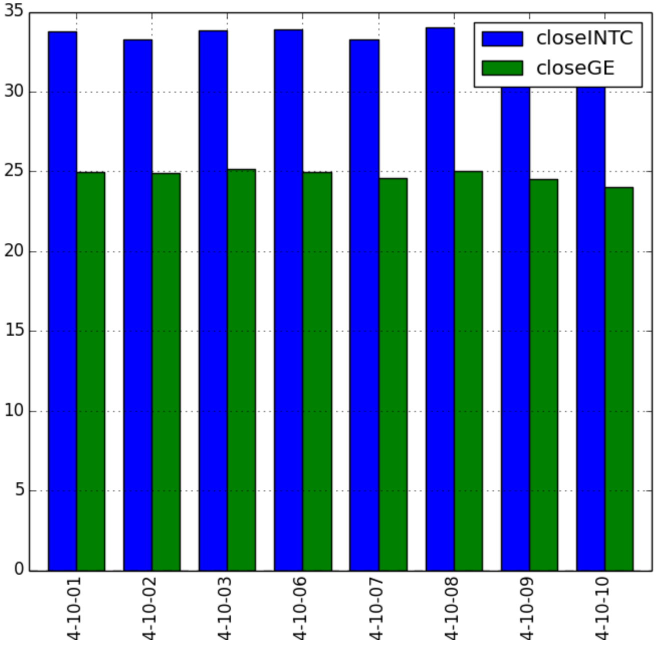
pandas控制图像形式

?

用柱状图比较Intel和GE两家科技公司2014年10月上旬的股票收盘价

Source

```
... #The data of Coca-Cola and IBM is ready
>>> quotesdf = pd.DataFrame()
>>> quotesdf['closeINTC'] = quotesdfINTC.close
>>> quotesdf['closeGE'] = quotesdfGE.close
>>> quotesdf.plot(kind='bar')
```

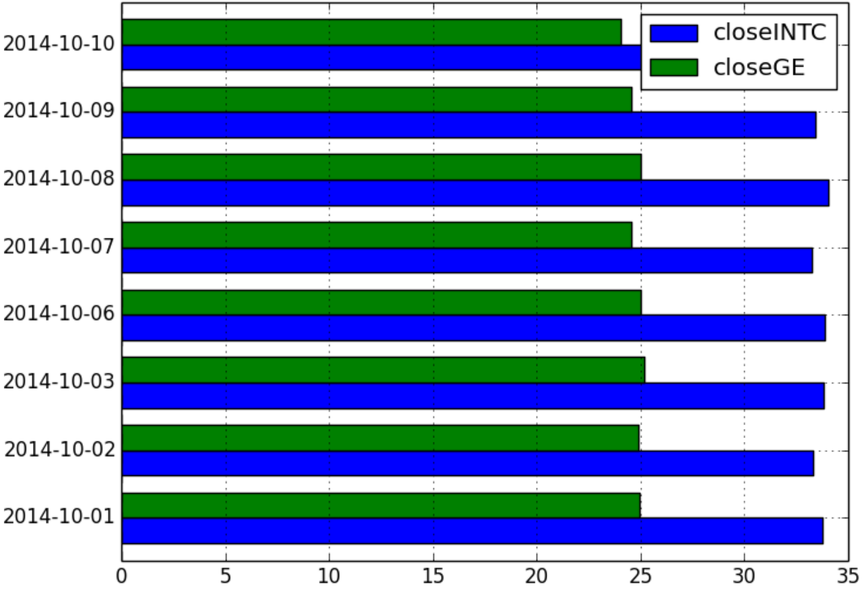


pandas控制图像形式

用柱状图比较Intel和GE两家科技公司2014年10月上旬的股票收盘价

quotesdf.plot(kind='bar')

quotesdf.plot(kind='barh')

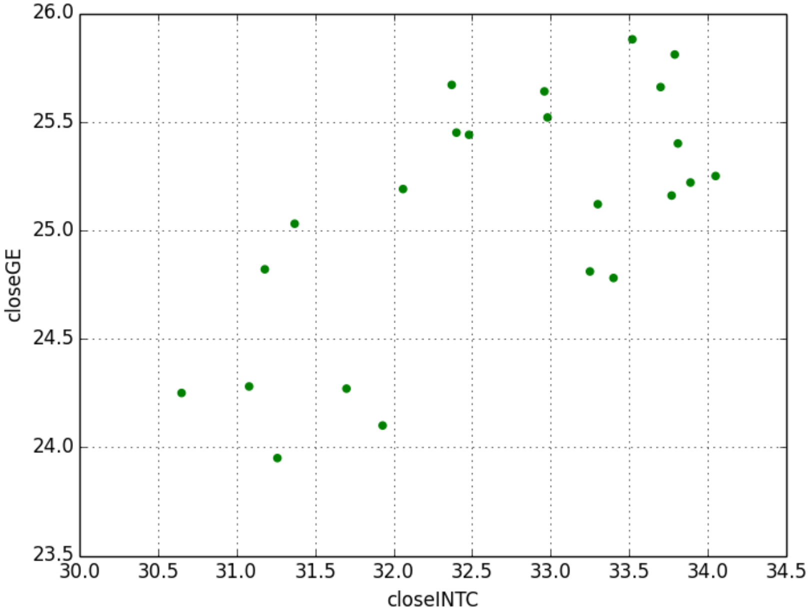


pandas控制图像形式

用柱状图比较Intel和GE两家科技公司2014年10月的股票收盘价

quotesdf.plot(kind='bar')

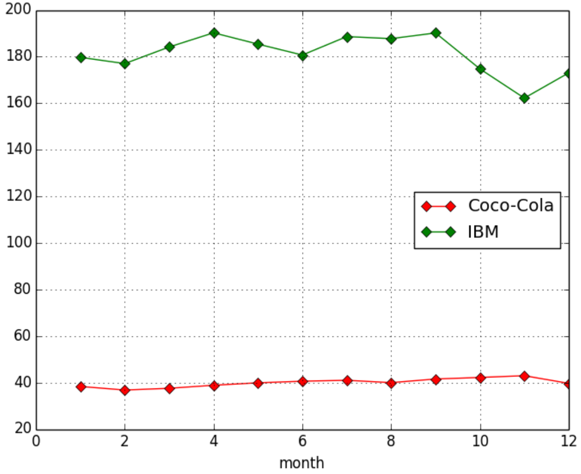
quotesdf.plot(kind='scatter',x='closeINTC',color='g',y='closeGE')



pandas控制图像属性

Source

```
… #The data of Coca-Cola and IBM is ready
>>> closeMeansKO.plot(color='r',marker='D',label='Coco-Cola')
<matplotlib.axes.AxesSubplot object at 0x08D5C650>
>>> closeMeansIBM.plot(color='g',marker='D',label='IBM')
<matplotlib.axes.AxesSubplot object at 0x08D5C650>
>>> plt.legend(loc='best')
<matplotlib.legend.Legend object at 0x08CBB2F0>
```

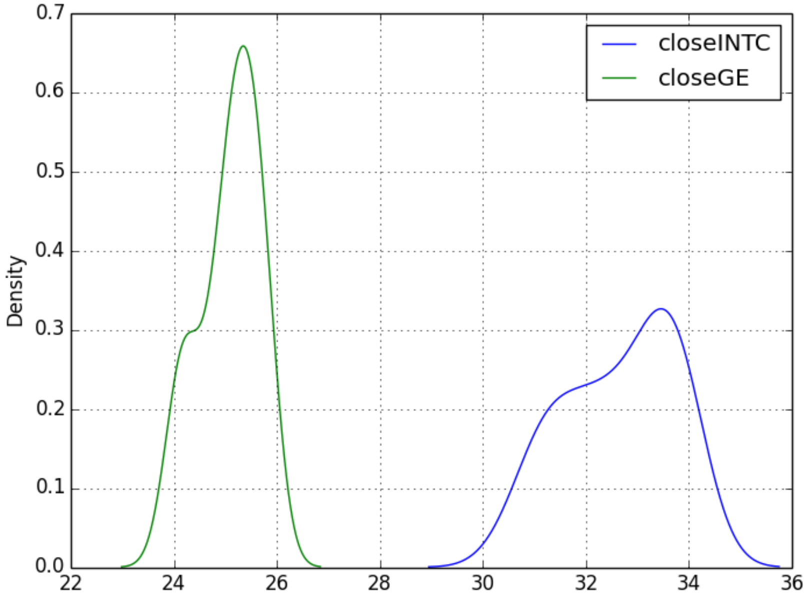


pandas控制图像属性

绘图显示Intel和GE两家科技公司2014年10月的股票收盘价的概率分布

quotesdf.plot(kind='bar')

quotesdf.plot(kind='kde')



5

用Python玩转数据

数据存取

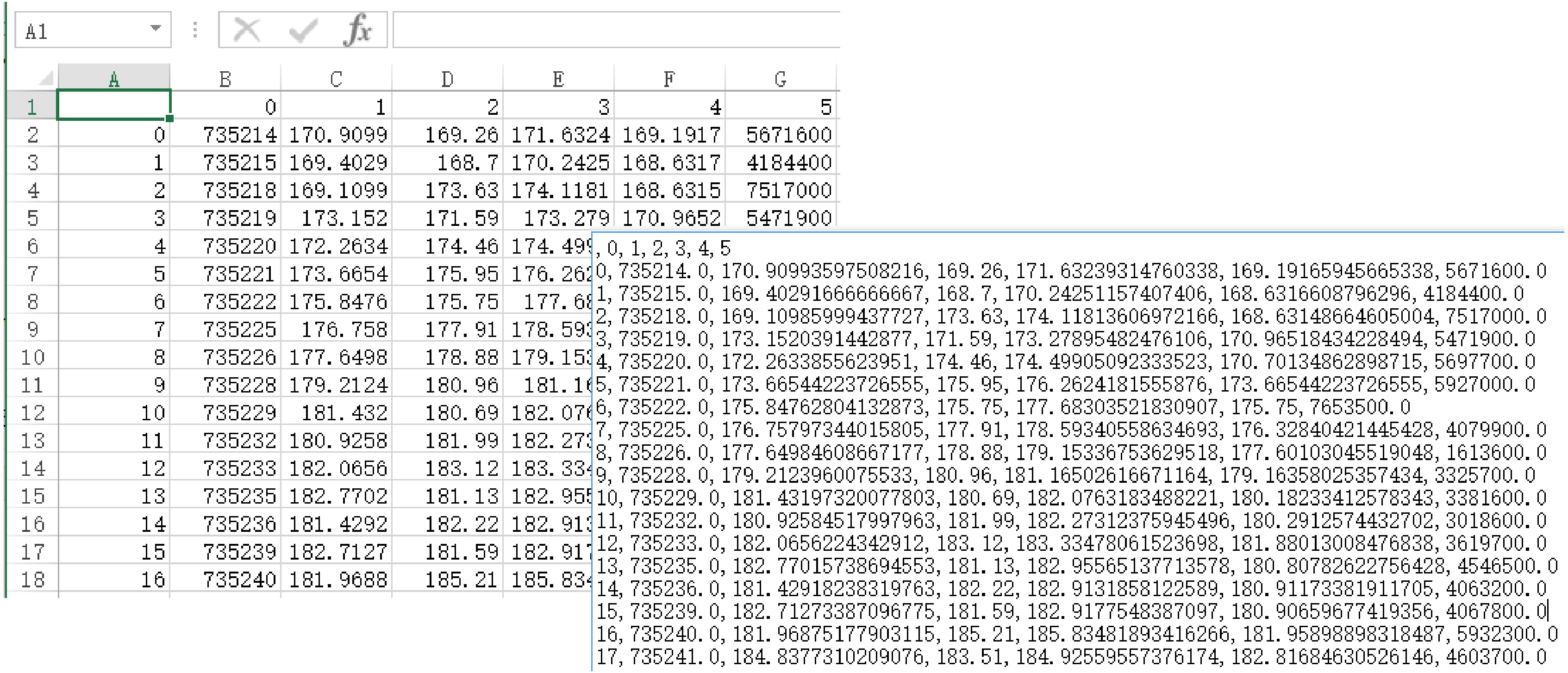
csv格式数据存取

？将IBM公司近一年来的股票基本信息存入文件stockIBM.csv中

File

```
# Filename: to_csv.py
from matplotlib.finance import quotes_historical_yahoo_ochl
from datetime import date
import pandas as pd
today = date.today()
start = (today.year-1, today.month, today.day)
quotes = quotes_historical_yahoo_ochl('IBM', start, today)
df = pd.DataFrame(quotes)
df.to_csv('stockIBM.csv')
```

csv格式数据存取



	A	B	C	D	E	F	G
1		0	1	2	3	4	5
2	0	735214	170.9099	169.26	171.6324	169.1917	5671600
3	1	735215	169.4029	168.7	170.2425	168.6317	4184400
4	2	735218	169.1099	173.63	174.1181	168.6315	7517000
5	3	735219	173.152	171.59	173.279	170.9652	5471900
6	4	735220	172.2634	174.46	174.49		
7	5	735221	173.6654	175.95	176.26		
8	6	735222	175.8476	175.75	177.6		
9	7	735225	176.758	177.91	178.59		
10	8	735226	177.6498	178.88	179.15		
11	9	735228	179.2124	180.96	181.1		
12	10	735229	181.432	180.69	182.07		
13	11	735232	180.9258	181.99	182.27		
14	12	735233	182.0656	183.12	183.33		
15	13	735235	182.7702	181.13	182.95		
16	14	735236	181.4292	182.22	182.91		
17	15	735239	182.7127	181.59	182.91		
18	16	735240	181.9688	185.21	185.83		

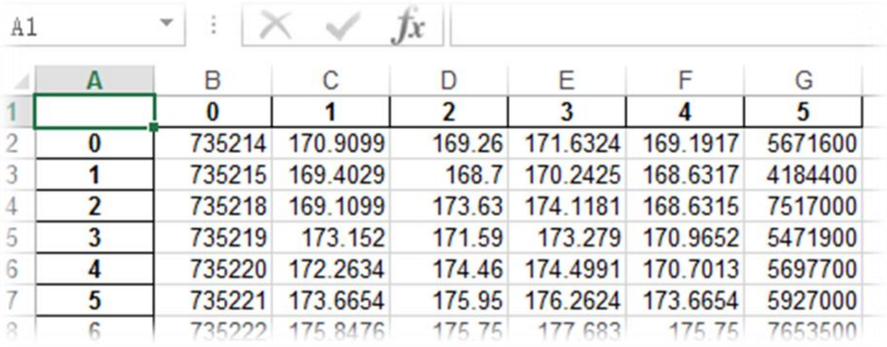
```
,0,1,2,3,4,5
0,735214.0,170.90993597508216,169.26,171.63239314760338,169.19165945665338,5671600.0
1,735215.0,169.40291666666667,168.7,170.24251157407406,168.6316608796296,4184400.0
2,735218.0,169.10985999437727,173.63,174.11813606972166,168.63148664605004,7517000.0
3,735219.0,173.1520391442877,171.59,173.27895482476106,170.96518434228494,5471900.0
4,735220.0,172.2633855623951,174.46,174.49905092333523,170.70134862898715,5697700.0
5,735221.0,173.66544223726555,175.95,176.2624181555876,173.66544223726555,5927000.0
6,735222.0,175.84762804132873,175.75,177.68303521830907,175.75,7653500.0
7,735225.0,176.75797344015805,177.91,178.59340558634693,176.32840421445428,4079900.0
8,735226.0,177.64984608667177,178.88,179.15336753629518,177.60103045519048,1613600.0
9,735228.0,179.2123960075533,180.96,181.1650261667164,179.1635802535743 4,3325700.0
10,735229.0,181.43197320077803,180.69,182.0763183488221,180.18233412578343,3381600.0
11,735232.0,180.92584517997963,181.99,182.27312375945496,180.2912574432702,3018600.0
12,735233.0,182.0656224342912,183.12,183.33478061523698,181.88013008476838,3619700.0
13,735235.0,182.77015738694553,181.13,182.95565137713578,180.8078262275642 8,4546500.0
14,735236.0,181.42918238319763,182.22,182.9131858122589,180.91173381911705,4063200.0
15,735239.0,182.71273387096775,181.59,182.9177548387097,180.90659677419356,4067800.0
16,735240.0,181.96875177903115,185.21,185.83481893416266,181.95898898318487,5932300.0
17,735241.0,184.8377310209076,183.51,184.92559557376174,182.81684630526146,4603700.0
```

csv格式数据存取

Source

```
>>> result = pd.read_csv('stockIBM.csv')
>>> result
   Unnamed: 0      0           1       2           3           4        5
0           0 735214  170.909936  169.26  171.632393  169.191659  5671600
1           1 735215  169.402917  168.70  170.242512  168.631661  4184400
2           2 735218  169.109860  173.63  174.118136  168.631487  7517000
3           3 735219  173.152039  171.59  173.278955  170.965184  5471900
…
>>> print result['2']
0    169.26
1    168.70
2    173.63
3    171.59
…
```

xls格式数据存取



A1

	A	B	C	D	E	F	G
1		0	1	2	3	4	5
2	0	735214	170.9099	169.26	171.6324	169.1917	5671600
3	1	735215	169.4029	168.7	170.2425	168.6317	4184400
4	2	735218	169.1099	173.63	174.1181	168.6315	7517000
5	3	735219	173.152	171.59	173.279	170.9652	5471900
6	4	735220	172.2634	174.46	174.4991	170.7013	5697700
7	5	735221	173.6654	175.95	176.2624	173.6654	5927000
8	6	735222	175.8476	175.75	177.683	175.75	7653500

File

```
# Filename: to_excel.py
from datetime import date
import pandas as pd
from matplotlib.finance import quotes_historical_yahoo_ochl
today = date.today()
start = (today.year-1, today.month, today.day)
quotes = quotes_historical_yahoo_ochl('IBM', start, today)
df = pd.DataFrame(quotes)
df.to_excel('stockIBM.xls', sheet_name='IBM')
```

新版pandas已经支持读写xlsx格式文件

用Python玩转数据

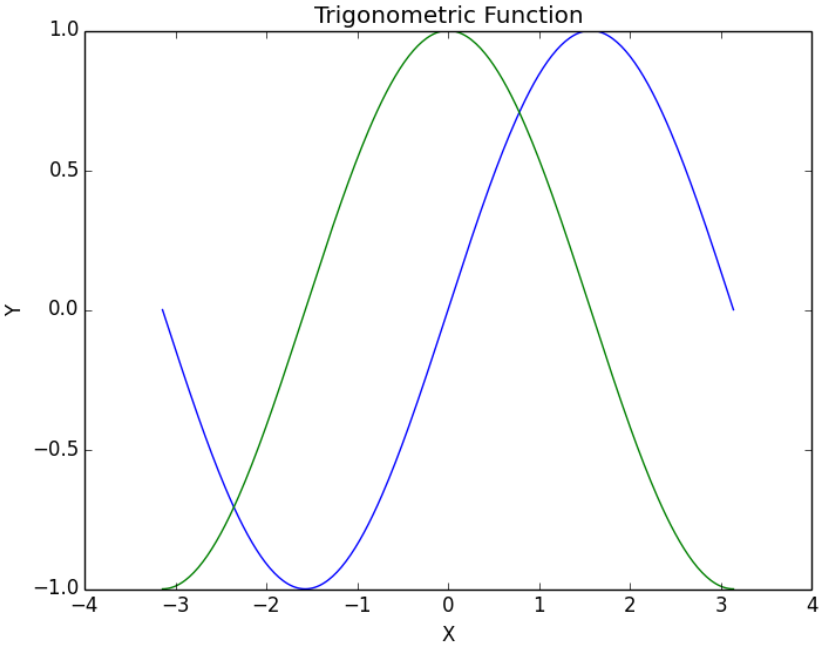
6

PYTHON的理工类应用

简单的三角函数计算

File

```
# Filename: mathA.py
import numpy as np
import pylab as pl
x = np.linspace(-np.pi, np.pi, 256)
s = np.sin(x)
c = np.cos(x)
pl.title('Trigonometric Function')
pl.xlabel('X')
pl.ylabel('Y')
pl.plot(x,s)
pl.plot(x,c)
```

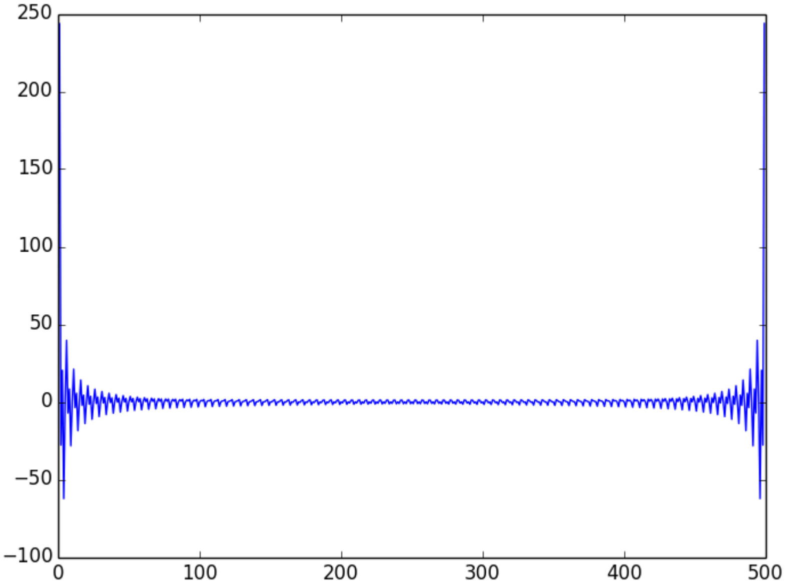


一组数据的傅里叶变换

数组：[1,1,...,1,-1,-1,...,1,1,1...,1]

File

```
# Filename: mathB.py
import scipy as sp
import pylab as pl
listA = sp.ones(500)
listA[100:300] = -1
f = sp.fft(listA)
pl.plot(f)
pl.show()
```



Biopython

功能

- 将生物信息学文件分析成Python可利用的数据结构
- 处理常用的在线生物信息学数据库代码
- 提供常用生物信息程序的界面

计划

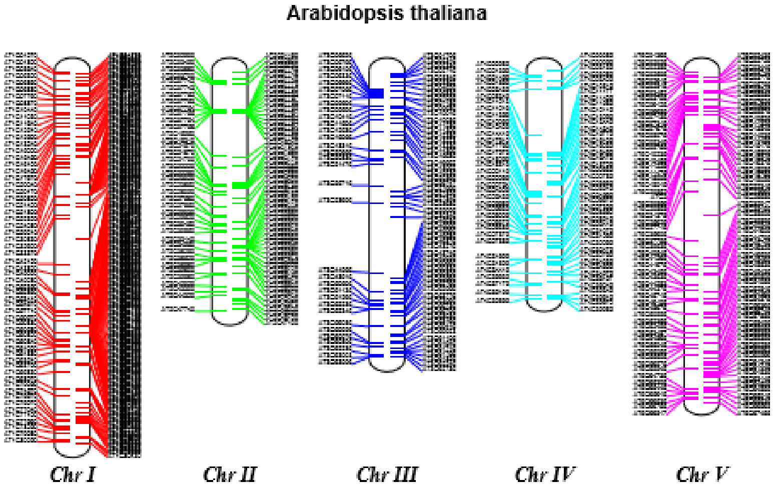
- 一个使用Python开发计算分子生物学工具的国际社团

Biopython

序列、字母表和染色体图

Source

```
>>> from Bio.Seq import Seq
>>> my_seq = Seq("AGTACACTGGT")
>>> my_seq.alphabet
Alphabet()
>>> print my_seq
AGTACACTGGT
```

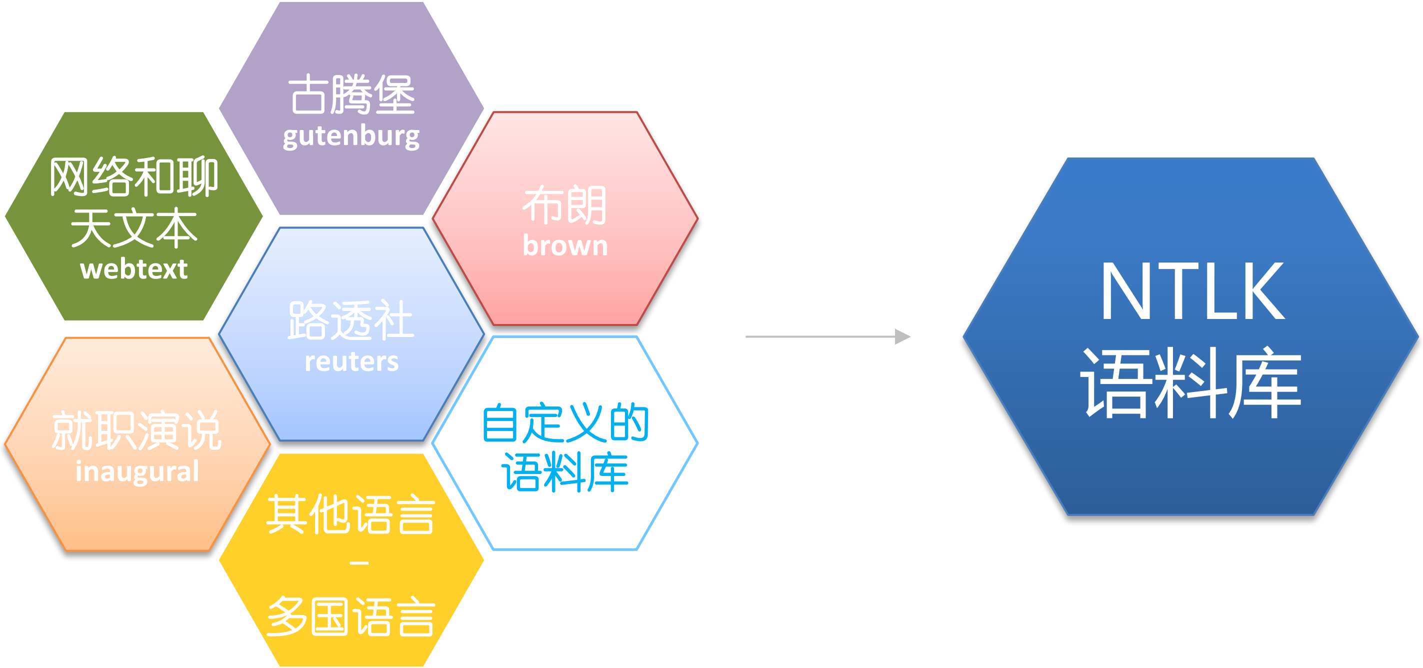


7

用Python玩转数据

PYTHON的人文社科类应用

NTLK语料库


古腾堡
gutenburg
网络和聊天文本
webtext
布朗
brown
路透社
reuters
就职演说
inaugural
自定义的语料库
其他语言
–
多国语言
NTLK
语料库

古滕堡项目

- 计算NTLK中目前收录的古滕堡项目的书

Source

```
>>> from nltk.corpus import gutenberg
>>> gutenberg.fileids()
[u'austen-emma.txt', u'austen-persuasion.txt', u'austen-sense.txt',
u'bible-kjv.txt', u'blake-poems.txt', u'bryant-stories.txt', u'burgess-
busterbrown.txt', u'carroll-alice.txt', u'chesterton-ball.txt',
u'chesterton-brown.txt', u'chesterton-thursday.txt', u'edgeworth-
parents.txt', u'melville-moby_dick.txt', u'milton-paradise.txt',
u'shakespeare-caesar.txt', u'shakespeare-hamlet.txt', u'shakespeare-
macbeth.txt', u'whitman-leaves.txt']
```

古滕堡项目

- 一些简单的计算

Source

```
>>> from nltk.corpus import gutenberg
>>> allwords = gutenberg.words('shakespeare-hamlet.txt')
>>> len(allwords)
37360
>>> len(set(allwords))
5447
>>> all_words.count('Hamlet')
99
>>> A = set(allwords)
>>> longwords = [w for w in A if len(w) > 12]
>>> print sorted(longwords)
```

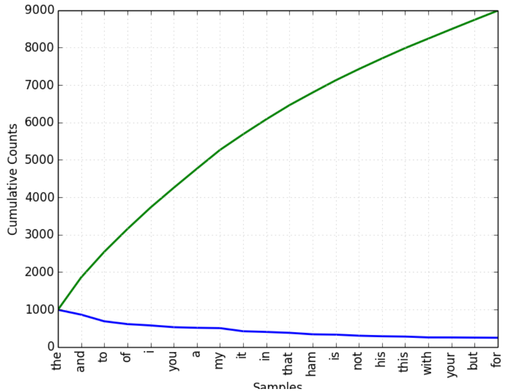
Output:

```
[u'Circumstances',
u'Guildensterne',
u'Incontinencie',
u'Recognizances',
u'Vnderstanding',
u'determination',
u'encompassement',
u'entertainment',
u'imperfections',
u'indifferently',
u'instrumentall',
u'reconcilement',
u'stubbornnesse',
u'transformation',
u'vnderstanding']
```

古滕堡项目

File

```
# Filename: freqG20.py
from nltk.corpus import gutenberg
from nltk.probability import *
fd2 = FreqDist([sx.lower() for sx in allwords if sx.isalpha()])
print fd2.B()
print fd2.N()
fd2.tabulate(20)
fd2.plot(20)
fd2.plot(20, cumulative = True)
```



Output:

```
4699
30266
 the  and   to   of    i  you    a   my   it   in that  ham
is  not  his this with your  but  for
 993  863  685  610  574  527  511  502  419  400
377  337  328  300  285  276  254  253  249  245
```

就职演说语料库

Source

```
>>> from nltk.corpus import inaugural
>>> fd3 = FreqDist([s for s in inaugural.words()])
>>> print fd3.freq('freedom')
0.00119394791917
```

File

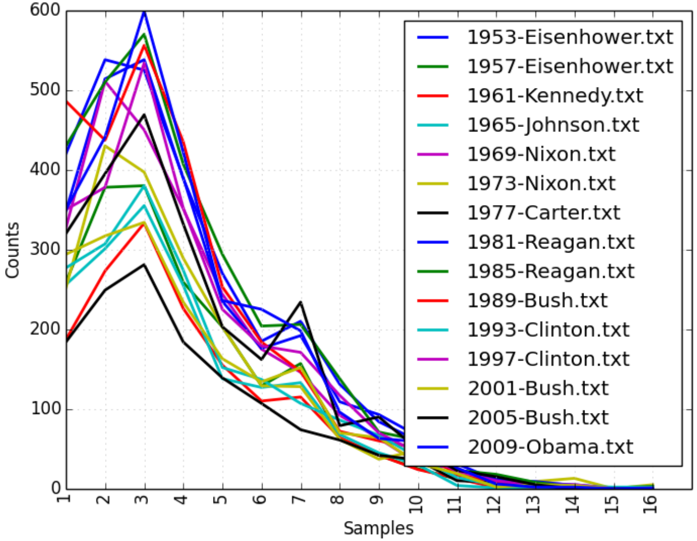
```
# Filename: inaugural.py
from nltk.corpus import inaugural
from nltk.probability import ConditionalFreqDist
cfd = ConditionalFreqDist(
            (fileid, len(w))
            for fileid in inaugural.fileids()
            for w in inaugural.words(fileid)
            if fileid > '1950')
print cfd.items()[:40]
cfd.plot()
```

新版NLTK需要加上这行，并且要保证语料库的下载

就职演说语料库

Output:

```
[(u'1965-Johnson.txt', FreqDist({3: 355, 2:
301, 1: 256, 4: 255, 5: 138, 7: 133, 6: 127, 8:
68, 9: 45, 10: 30, ...})), (u'1997-Clinton.txt',
FreqDist({3: 534, 2: 378, 4: 352, 1: 350, 5:
225, 6: 179, 7: 171, 8: 117, 9: 70, 10:
45, ...})), (u'2009-Obama.txt', FreqDist({3:
599, 2: 441, 4: 422, 1: 350, 5: 236, 6: 225, 7:
198, 8: 96, 9: 63, 10: 59, ...})), ...
```


Counts
Samples
1953-Eisenhower.txt
1957-Eisenhower.txt
1961-Kennedy.txt
1965-Johnson.txt
1969-Nixon.txt
1973-Nixon.txt
1977-Carter.txt
1981-Reagan.txt
1985-Reagan.txt
1989-Bush.txt
1993-Clinton.txt
1997-Clinton.txt
2001-Bush.txt
2005-Bush.txt
2009-Obama.txt