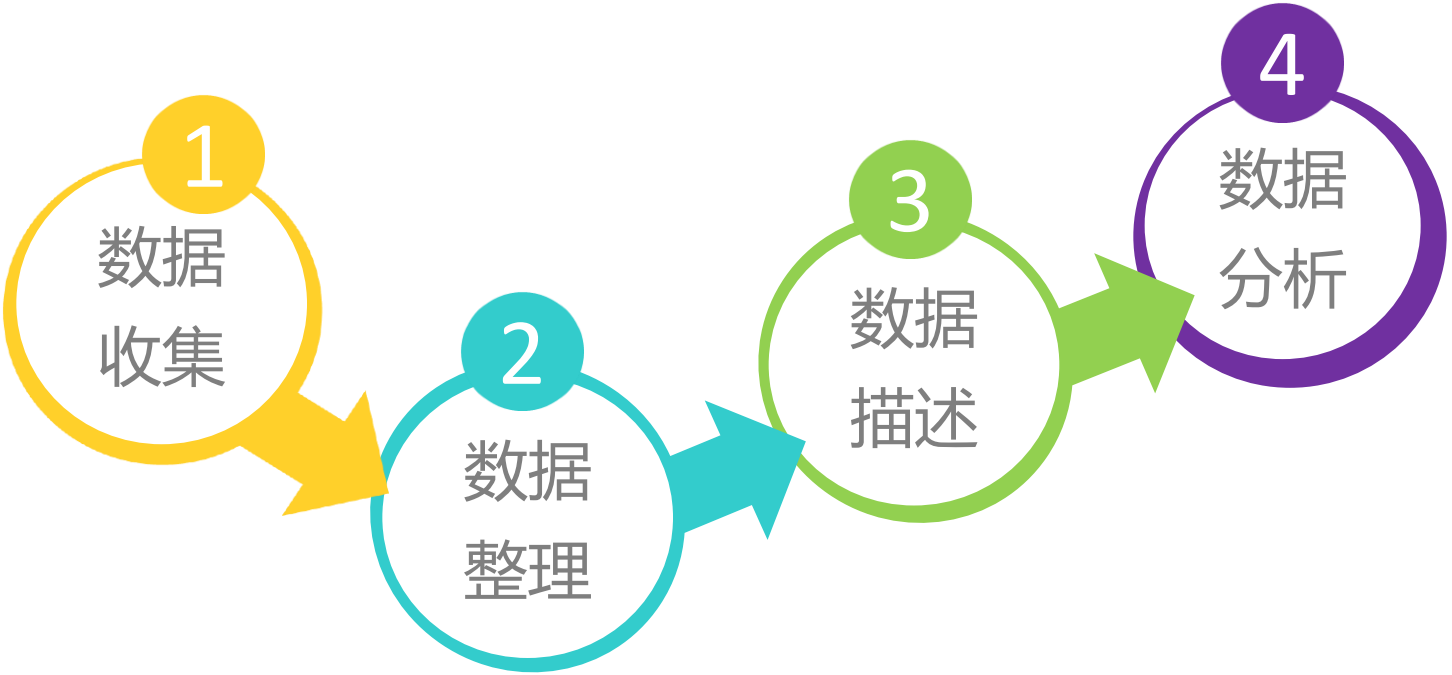
Basic data processing of Python

Python基本数据统计

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

简单数据处理过程

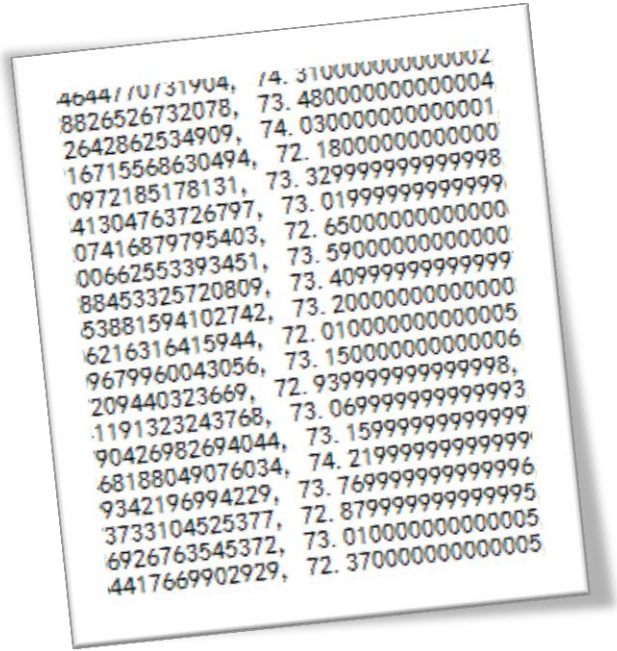


1

用Python玩转数据

便捷数据获取

用Python获取数据



本地数据如何获取?

文件的打开，读写和关闭

- 文件打开
- 读文件

  写文件
- 文件关闭

用Python获取数据

ymbol	Company Name	Pric
OK	Nokia Corporation	8.4
APL	Apple Inc.	104.7
AC	Bank of America Cor...	16.6
	AT&T, Inc.	33.5
BR	Petr	12.5
QQ	PowerShares QQQ	97.9
B	Facebook, Inc.	80.0
IO	iBio, Inc.	1.4
ELP	Yelp, Inc.	58.5
NFN	Infinera Corporation	13.0
ISFT	Microsoft Corporation	44.7
O	The Coca-Cola Comp...	41.2
FE	Pfizer Inc	28.8

网络数据如何获取?

抓取网页，解析网页内容

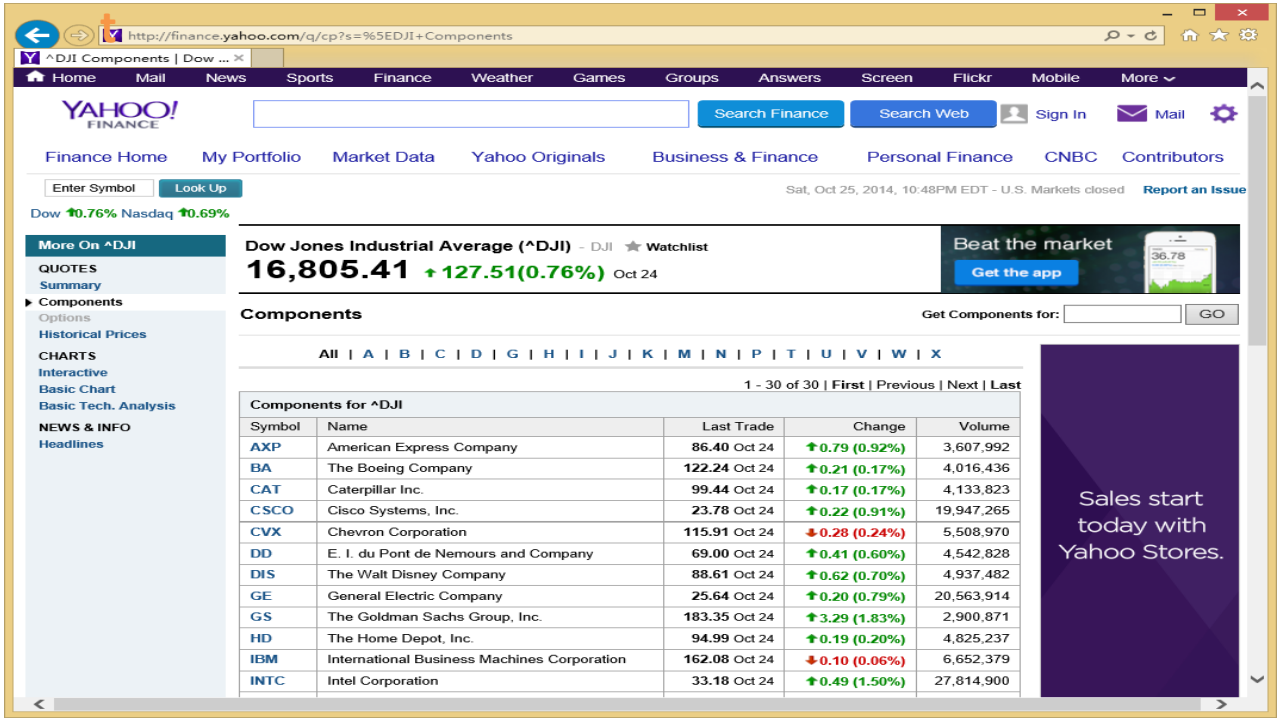
- urllib
- urllib2
- httplib
- httplib2

Python 3中被urllib.request代替

Python 3中被http.client代替

yahoo财经数据

http://finance.yahoo.com/q/cp?s=%5EDJI+Componen



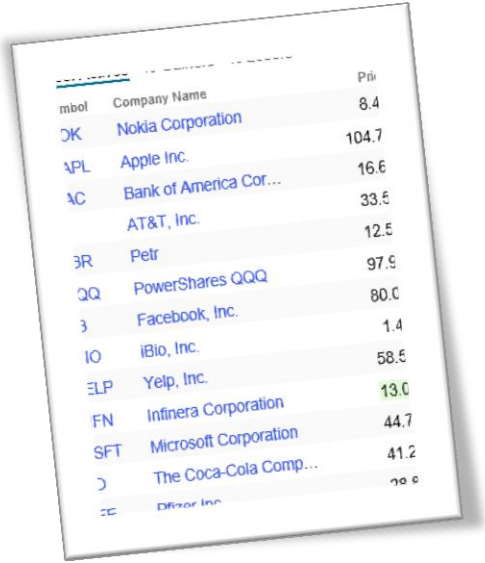
利用urllib库获取yahoo财经数据

File

```
# Filename: dji.py
import urllib
import re
dBytes = urllib.request.urlopen('http://finance.yahoo.com/q/cp?s=%5EDJI+Components').read()
dStr = dBytes.decode() #在python3中urllib.read()返回bytes对象而非str，语句功能是将dBytes转换成Str
m = re.findall('<tr><td class="yfnc_tabledata1"><b><a href=".*?">(.*?)</a></b></td><td
                class="yfnc_tabledata1">(.*?)</td>.*?<b>(.*?)</b>.*?</tr>', dStr)
if m:
    print m
    print '\n'
    print len(m)
else:
    print 'not match'
```

数据形式

- 包含多个字符串（dji）
  - 'AXP', 'American Express Company', '86.40'
  - 'BA', 'The Boeing Company', '122.24'
  - 'CAT', 'Caterpillar Inc.', '99.44'
  - 'CSCO', 'Cisco Systems, Inc.', '23.78'
  - 'CVX', 'Chevron Corporation', '115.91'
  - …



mbol	Company Name	Pri
OK	Nokia Corporation	8.4
APL	Apple Inc.	104.7
AC	Bank of America Cor…	16.6
	AT&T, Inc.	33.5
BR	Petr	12.5
QQ	PowerShares QQQ	97.9
B	Facebook, Inc.	80.0
IO	iBio, Inc.	1.4
ELP	Yelp, Inc.	58.5
FN	Infinera Corporation	13.0
SFT	Microsoft Corporation	44.7
D	The Coca-Cola Comp…	41.2
	Pfizer Inc	

便捷网络数据

是否能够简单方便并且快速的方式获得雅虎财经上各上市公司股票的历史数据？

File

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

函数目前更新为 quotes_historical_yahoo_ochl

便捷网络数据

quotes的内容

日期 开盘价 收盘价 最高价 最低价 成交量

	0	1	2	3	4	5
0	735187	82.137528	81.91	82.365056	81.484622	4439700
1	735190	81.954487	81.45	82.083082	81.271946	3104800
2	735191	81.490863	81.57	81.985467	81.382051	2775000
3	735192	82.062317	81.36	82.517340	81.073138	3510500
4	735193	81.744151	83.05	83.099464	81.684795	4355200
5	735194	83.019249	82.95	83.029141	82.475152	2690300
6	735197	83.412905	83.68	83.848171	83.175486	3657700
7	735198	83.833676	84.17	84.664594	83.784217	4896800
8	735199	84.343547	84.67	84.739248	84.066557	2520000
9	735201	84.959142	84.88	85.265818	84.593110	2144600
10	735204	85.092126	84.37	85.596624	84.241402	3620800
11	735205	83.976989	83.70	84.412256	83.294410	3546100
12	735206	83.303123	83.59	84.322031	82.857969	3579700

便捷网络数据

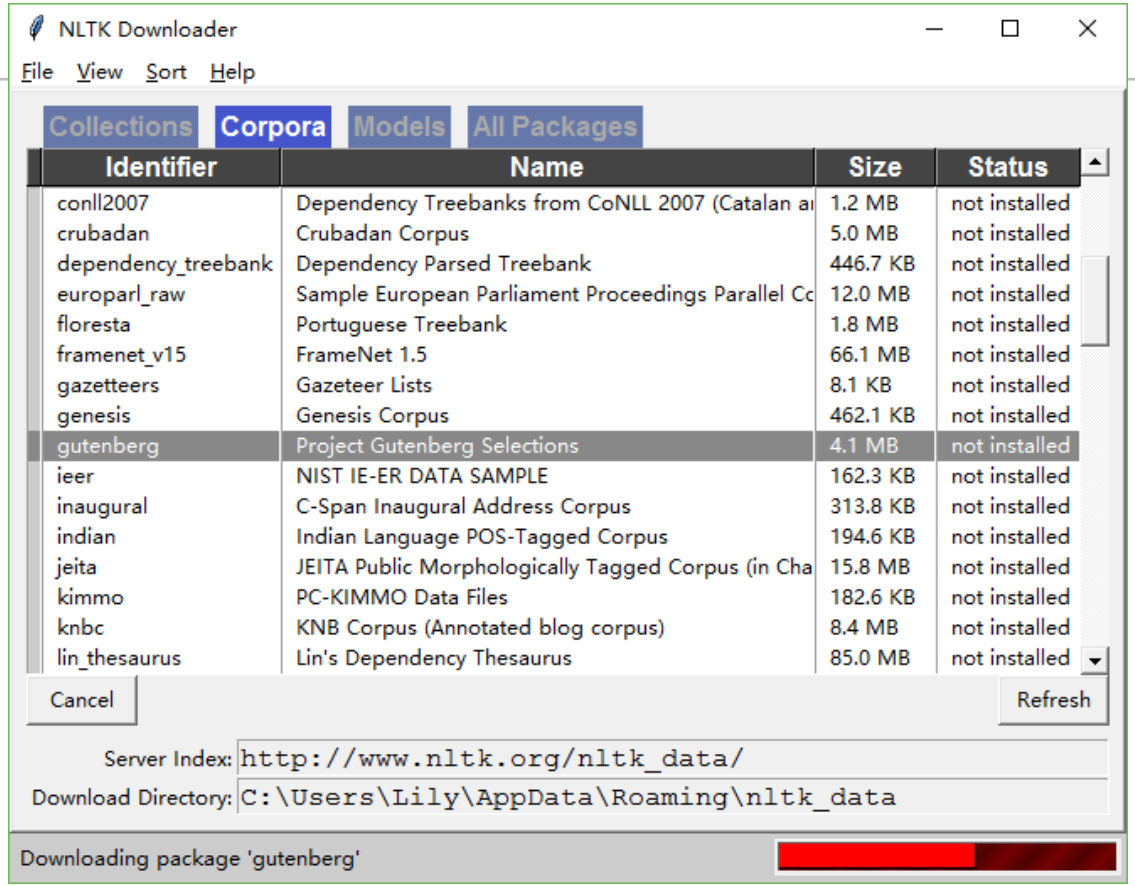
需要先执行nltk.download()下载某一个或多个包，若下载失败，可以在官网（http://www.nltk.org/nltk_data/）单独下载后放到本地python目录的nltk_data\corpora下

brown

```
>>> from nltk.corpus import gutenberg
>>> import nltk
>>> print 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']
>>> texts = gutenberg.words('shakespeare-hamlet.txt')
[u'[', u'The', u'Tragedie', u'of', u'Hamlet', u'by', ...]
```

自然语言工具包NLTK

- 古腾堡语料库
- 布朗语料库
- 路透社语料库
- 网络和聊天文本
- …


NLTK Downloader
File View Sort Help
Collections Corpora Models All Packages
Identifier Name Size Status
conll2007 Dependency Treebanks from CoNLL 2007 (Catalan a 1.2 MB not installed
crubadan Crubadan Corpus 5.0 MB not installed
dependency_treebank Dependency Parsed Treebank 446.7 KB not installed
europarl_raw Sample European Parliament Proceedings Parallel Co 12.0 MB not installed
floresta Portuguese Treebank 1.8 MB not installed
framenet_v15 FrameNet 1.5 66.1 MB not installed
gazetteers Gazeteer Lists 8.1 KB not installed
genesis Genesis Corpus 462.1 KB not installed
gutenberg Project Gutenberg Selections 4.1 MB not installed
ieer NIST IE-ER DATA SAMPLE 162.3 KB not installed
inaugural C-Span Inaugural Address Corpus 313.8 KB not installed
indian Indian Language POS-Tagged Corpus 194.6 KB not installed
jeita JEITA Public Morphologically Tagged Corpus (in Cha 15.8 MB not installed
kimmo PC-KIMMO Data Files 182.6 KB not installed
knbc KNB Corpus (Annotated blog corpus) 8.4 MB not installed
lin_thesaurus Lin's Dependency Thesaurus 85.0 MB not installed
Cancel
Refresh
Server Index: http://www.nltk.org/nltk_data/
Download Directory: C:\Users\Lily\AppData\Roaming\nltk_data
Downloading package 'gutenberg'

2

用Python玩转数据

数据准备

数据形式

30支成分股（dji）股票数据的逻辑结构

公司代码	公司名	最近一次成交价

美国运通公司（quotes）股票详细数据的逻辑结构

日期	开盘价	收盘价	最高价	最低价	成交量

数据整理

quotes数据加属性名

File

```
# Filename: quotesproc.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('AXP', start, today)
fields = ['date','open','close','high','low','volume']
quotesdf = pd.DataFrame(quotes, columns = fields)
print quotesdf
```

	0	1	2	3	4	5
0	735190	81.954487	81.45	82.083082	81.271946	3104800
1	735191	81.490863	81.57	81.985467	81.382051	2775000
2	735192	82.062317	81.36	82.517340	81.073138	3510500
3	735193	81.744151	83.05	83.099464	81.684795	4355200
4	735194	83.019249	82.95	83.029141	82.475152	2690300
5	735197	83.412905	83.68	83.848171	83.175486	3657700
6	735198	83.833676	84.17	84.664594	83.784217	4896800
7	735199	84.343547	84.67	84.739248	84.066557	2520000
8	735201	84.959142	84.88	85.265818	84.593110	2144600
9	735204	85.092126	84.37	85.596624	84.241402	3620800
10	735205	83.976989	83.70	84.412256	83.294410	3546100
11	735206	83.303123	83.59	84.322031	82.857969	3579700
12	735207	83.362906	83.63	84.075156	83.244198	3677800
13	735208	84.663680	85.00	85.158268	84.426277	2666600
14	735211	85.245494	84.83	85.473026	84.434292	2436500
15	735212	84.459029	84.37	84.815146	84.162265	2354200
16	735213	84.365197	83.00	84.751013	82.910965	3782300
17	735214	82.874814	82.40	83.369412	82.310972	3574300
18	735215	82.423872	82.78	83.017419	82.334840	3594900
19	735218	83.056029	83.61	84.055155	83.056029	3599000
20	735219	83.793509	83.20	83.803400	83.081298	2893700
21	735220	83.388278	85.06	85.188594	82.646389	5332300
22	735221	85.015058	85.48	85.598708	84.985381	3956300
23	735222	85.779231	86.63	86.837746	85.512130	6294500
24	735225	87.053335	87.36	87.627095	86.993980	3242600
25	735226	87.383788	87.73	87.779459	87.235412	1042700

数据整理

dji数据：加属性名

code	name	lasttrade
AXP		
BA		
CAT		
…		
XOM		

quotes数据：加属性名

date	open	close	high	low	volume
735190.0					
735191.0					
735192.0					
…					
735551.0					

数据整理

用1,2,...作为索引

```
quotesdf = pd.DataFrame(quotes, columns = fields)
```

```
quotesdf = pd.DataFrame(quotes, index = range(1,len(quotes)+1),columns = fields)
```

	date	open	close	high	low	volume
0	735190	81.954487	81.45	82.083082	81.271946	3104800
1	735191	81.490863	81.57	81.985467	81.382051	2775000
2	735192	82.062317	81.36	82.517340	81.073138	3510500
3	735193	81.744151	83.05	83.099464	81.684795	4355200
4	735194	83.019249	82.95	83.029141	82.475152	2690300

	date	open	close	high	low	volume
1	735190	81.954487	81.45	82.083082	81.271946	3104800
2	735191	81.490863	81.57	81.985467	81.382051	2775000
3	735192	82.062317	81.36	82.517340	81.073138	3510500
4	735193	81.744151	83.05	83.099464	81.684795	4355200
5	735194	83.019249	82.95	83.029141	82.475152	2690300

数据整理

如果可以直接用date作为索引，quotes的时间能否转换成常规形式（如下图中的效果）？

	open	close	high	low	volume
2013-11-18	81.954487	81.45	82.083082	81.271946	3104800
2013-11-19	81.490863	81.57	81.985467	81.382051	2775000
2013-11-20	82.062317	81.36	82.517340	81.073138	3510500
2013-11-21	81.744151	83.05	83.099464	81.684795	4355200
2013-11-22	83.019249	82.95	83.029141	82.475152	2690300
2013-11-25	83.412905	83.68	83.848171	83.175486	3657700
2013-11-26	83.833676	84.17	84.664594	83.784217	4896800
2013-11-27	84.343547	84.67	84.739248	84.066557	2520000
2013-11-29	84.959142	84.88	85.265818	84.593110	2144600
2013-12-02	85.092126	84.37	85.596624	84.241402	3620800
2013-12-03	83.976989	83.70	84.412256	83.294410	3546100
2013-12-04	83.303123	83.59	84.322031	82.857969	3579700
2013-12-05	83.362906	83.63	84.075156	83.244198	3677800
2013-12-06	84.663680	85.00	85.158268	84.426277	2666600
2013-12-09	85.245494	84.83	85.473026	84.434292	2436500
2013-12-10	84.459029	84.37	84.815146	84.162265	2354200
2013-12-11	84.365197	83.00	84.751013	82.910965	3782300
2013-12-12	82.874814	82.40	83.369412	82.310972	3574300

Source

```
>>> from datetime import date
>>> firstday = date.fromordinal(735190)
>>> lastday = date.fromordinal(735551)
>>> firstday
datetime.date(2013, 11, 18)
>>> lastday
datetime.date(2014, 11, 14)
```

时间序列

File

```
# Filename: quotesproc.py
from matplotlib.finance import quotes_historical_yahoo_ochl
from datetime import date
from datetime import datetime
import pandas as pd
today = date.today()
start = (today.year-1, today.month, today.day)
quotes = quotes_historical_yahoo_ochl('AXP', start, today)
fields = ['date','open','close','high','low','volume']
list1 = []
for i in range(0,len(quotes)):
    x = date.fromordinal(int(quotes[i][0]))
    y = datetime.strftime(x,'%Y-%m-%d')
    list1.append(y)
quotesdf = pd.DataFrame(quotes, index = list1, columns = fields)
quotesdf = quotesdf.drop(['date'], axis = 1)
print quotesdf
```

转换成常规时间

转换成固定格式

删除原date列

创建时间序列

Source

```
>>> import pandas as pd
>>> dates = pd.date_range('20141001', periods=7)
>>> dates
<class 'pandas.tseries.index.DatetimeIndex'>
[2014-10-01, ..., 2014-10-07]
Length: 7, Freq: D, Timezone: None
>>> import numpy as np
>>> dates = pd.DataFrame(np.random.randn(7,3),index=dates,columns = list('ABC'))
>>> dates
                   A         B         C
2014-10-01  1.302600 -1.214708  1.411628
2014-10-02 -0.512343  2.277474  0.403811
2014-10-03 -0.788498 -0.217161  0.173284
2014-10-04  1.042167 -0.453329 -2.107163
2014-10-05 -1.628075  1.663377  0.943582
2014-10-06 -0.091034  0.335884  2.455431
2014-10-07 -0.679055 -0.865973  0.246970
[7 rows x 3 columns]
```

3

用Python玩转数据

数据显示

数据显示

	code	name	lasttrade
0	AXP	American Express Company	90.67
1	BA	The Boeing Company	128.86
2	CAT	Caterpillar Inc.	101.34
3	CSCO	Cisco Systems, Inc.	26.32
4	CVX	Chevron Corporation	116.32
5	DD	E. I. du Pont de Nemours and Company	70.80
6	DIS	The Walt Disney Company	90.80
7	GE	General Electric Company	26.46
8	GS	The Goldman Sachs Group, Inc.	189.98
9	HD	The Home Depot, Inc.	98.24
10	IBM	International Business Machines Corporation	164.16
11	INTC	Intel Corporation	33.95
12	JNJ	Johnson & Johnson	108.16
13	JPM	JPMorgan Chase & Co.	60.28
14	KO	The Coca-Cola Company	42.73
15	MCD	McDonald's Corp.	96.21
16	MMM	3M Company	158.85
17	MRK	Merck & Co. Inc.	59.07
18	MSFT	Microsoft Corporation	49.58
19	NKE	Nike, Inc.	95.50
20	PFE	Pfizer Inc.	30.34
21	PG	The Procter & Gamble Company	88.11
22	T	AT&T, Inc.	35.90
23	TRV	The Travelers Companies, Inc.	102.43
24	UNH	UnitedHealth Group Incorporated	95.11
25	UTX	United Technologies Corporation	107.45
26	V	Visa Inc.	248.84
27	VZ	Verizon Communications Inc.	51.50
28	WMT	Wal-Mart Stores Inc.	82.96
29	XOM	Exxon Mobil Corporation	95.09

djidf

	open	close	high	low	volume
2013-11-18	81.954487	81.45	82.083082	81.271946	3104800
2013-11-19	81.490863	81.57	81.985467	81.382051	2775000
2013-11-20	82.062317	81.36	82.517340	81.073138	3510500
2013-11-21	81.744151	83.05	83.099464	81.684795	4355200
2013-11-22	83.019249	82.95	83.029141	82.475152	2690300
2013-11-25	83.412905	83.68	83.848171	83.175486	3657700
2013-11-26	83.833676	84.17	84.664594	83.784217	4896800
2013-11-27	84.343547	84.67	84.739248	84.066557	2520000
2013-11-29	84.959142	84.88	85.265818	84.593110	2144600
2013-12-02	85.092126	84.37	85.596624	84.241402	3620800
2013-12-03	83.976989	83.70	84.412256	83.294410	3546100
2013-12-04	83.303123	83.59	84.322031	82.857969	3579700
2013-12-05	83.362906	83.63	84.075156	83.244198	3677800
2013-12-06	84.663680	85.00	85.158268	84.426277	2666600
2013-12-09	85.245494	84.83	85.473026	84.434292	2436500
2013-12-10	84.459029	84.37	84.815146	84.162265	2354200
2013-12-11	84.365197	83.00	84.751013	82.910965	3782300
2013-12-12	82.874814	82.40	83.369412	82.310972	3574300

quotesdf

数据显示

显示方式：

- 显示索引
- 显示列名
- 显示数据的值
- 显示数据描述

Source

```
>>> djidf.index
Int64Index([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24, 25, 26, 27, 28, 29], dtype='int64')
>>> djidf.columns
Index([u'code', u'name', u'lasttrade'], dtype='object')
>>> dijdf.values
array([['AXP', 'American Express Company', '90.67'],
        ['BA', 'The Boeing Company', '128.86'],
         ...
        ['XOM', 'Exxon Mobil Corporation', '95.09']], dtype=object)
>>> djidf.describe
<bound method DataFrame.describe of   code      name     lasttrade
0     AXP                American Express Company       90.67
1      BA                         The Boeing Company      128.86
...
29   XOM                      Exxon Mobil Corporation       95.09
```

数据显示

索引的格式

Source

```
>>> quotesdf.index
Index([u'2013-11-18', u'2013-11-19', u'2013-11-20', u'2013-11-21',
u'2013-11-22', u'2013-11-25', u'2013-11-26', u'2013-11-27',
...
-04-08', u'2014-04-09', u'2014-04-10', u'2014-04-11', ...],
dtype='object')
```

数据显示

查看道琼斯工业股中前5只和后5只的股票基本信息？

显示方式：

- 显示行
  - 专用方式
  - 切片

Source

df[:5]

```
>>> djidf.head(5)
    code                          name  lasttrade
0    AXP  American Express Company        90.67
1     BA          The Boeing Company       128.86
2    CAT              Caterpillar Inc.     101.34
3   CSCO           Cisco Systems, Inc.      26.32
4    CVX          Chevron Corporation      116.32

[5 rows x 3 columns]
```

df[25:]

```
>>> djidf.tail(5)
     code                          name  lasttrade
25    UTX  United Technologies Corporation     107.45
26      V                       Visa Inc.     248.84
27     VZ     Verizon Communications Inc.      51.50
28    WMT              Wal-Mart Stores Inc.     82.96
29    XOM          Exxon Mobil Corporation      95.09

[5 rows x 3 columns]
```

4

用Python玩转数据

数据选择

数据选择

	code	name	lasttrade
0	AXP	American Express Company	90.67
1	BA	The Boeing Company	128.86
2	CAT	Caterpillar Inc.	101.34
3	CSCO	Cisco Systems, Inc.	26.32
4	CVX	Chevron Corporation	116.32
5	DD	E. I. du Pont de Nemours and Company	70.80
6	DIS	The Walt Disney Company	90.80
7	GE	General Electric Company	26.46
8	GS	The Goldman Sachs Group, Inc.	189.98
9	HD	The Home Depot, Inc.	98.24
10	IBM	International Business Machines Corporation	164.16
11	INTC	Intel Corporation	33.95
12	JNJ	Johnson & Johnson	108.16
13	JPM	JPMorgan Chase & Co.	60.28
14	KO	The Coca-Cola Company	42.73
15	MCD	McDonald's Corp.	96.21
16	MMM	3M Company	158.85
17	MRK	Merck & Co. Inc.	59.07
18	MSFT	Microsoft Corporation	49.58
19	NKE	Nike, Inc.	95.50
20	PFE	Pfizer Inc.	30.34
21	PG	The Procter & Gamble Company	88.11
22	T	AT&T, Inc.	35.90
23	TRV	The Travelers Companies, Inc.	102.43
24	UNH	UnitedHealth Group Incorporated	95.11
25	UTX	United Technologies Corporation	107.45
26	V	Visa Inc.	248.84
27	VZ	Verizon Communications Inc.	51.50
28	WMT	Wal-Mart Stores Inc.	82.96
29	XOM	Exxon Mobil Corporation	95.09

选择方式：

- 选择行
- 选择列
- 选择区域
- 筛选（条件选择）

	open	close	high	low	volume
2013-11-18	81.954487	81.45	82.083082	81.271946	3104800
2013-11-19	81.490863	81.57	81.985467	81.382051	2775000
2013-11-20	82.062317	81.36	82.517340	81.073138	3510500
2013-11-21	81.744151	83.05	83.099464	81.684795	4355200
2013-11-22	83.019249	82.95	83.029141	82.475152	2690300
2013-11-25	83.412905	83.68	83.848171	83.175486	3657700
2013-11-26	83.833676	84.17	84.664594	83.784217	4896800
2013-11-27	84.343547	84.67	84.739248	84.066557	2520000
2013-11-29	84.959142	84.88	85.265818	84.593110	2144600
2013-12-02	85.092126	84.37	85.596624	84.241402	3620800
2013-12-03	83.976989	83.70	84.412256	83.294410	3546100
2013-12-04	83.303123	83.59	84.322031	82.857969	3579700
2013-12-05	83.362906	83.63	84.075156	83.244198	3677800
2013-12-06	84.663680	85.00	85.158268	84.426277	2666600
2013-12-09	85.245494	84.83	85.473026	84.434292	2436500
2013-12-10	84.459029	84.37	84.815146	84.162265	2354200
2013-12-11	84.365197	83.00	84.751013	82.910965	3782300
2013-12-12	82.874814	82.40	83.369412	82.310972	3574300

数据选择

美国运通公司2013年12月2日至2013年12月6日间的股票交易信息？

选择方式：

- 选择行
  - 切片
  - 索引

Source

```
>>> quotesdf[u'2013-12-02':u'2013-12-06']
                 open  close       high       low   volume
2013-12-02  85.092126  84.37  85.596624  84.241402  3620800
2013-12-03  83.976989  83.70  84.412256  83.294410  3546100
2013-12-04  83.303123  83.59  84.322031  82.857969  3579700
2013-12-05  83.362906  83.63  84.075156  83.244198  3677800
2013-12-06  84.663680  85.00  85.158268  84.426277  2666600

[5 rows x 5 columns]
```

数据选择

道琼斯工业股公司代码？

选择方式：

- 选择列
  - 列名

Source

```
>>> djidf['code']
0      AXP
1       BA
2      CAT
...
29     XOM
Name: code, dtype: object
>>> djidf.code
0      AXP
1       BA
2      CAT
...
29     XOM
Name: code, dtype: object
```

不支持

djidf['code', 'lasttrade']

djidf['code':'lasttrade']

数据选择

道琼斯工业股中标号是1至5的股票信息以及所有股票的代码和最近一次交易价？

选择方式：

- 行、列
  - 标签label（loc）

Source

```
>>> djidf.loc[1:5,]
   code                                  name  lasttrade
1    BA                   The Boeing Company     128.86
2   CAT                         Caterpillar Inc.   101.34
3  CSCO                     Cisco Systems, Inc.     26.32
4   CVX                   Chevron Corporation    116.32
5    DD  E. I. du Pont de Nemours and Company     70.80
[5 rows x 3 columns]
>>> djidf.loc[:,['code','lasttrade']]
    code  lasttrade
0    AXP      90.67
1     BA     128.86
2    CAT     101.34
...
29   XOM      95.09
[30 rows x 2 columns]
```

数据选择

道琼斯工业股中标号是1至5的股票代码和最近一次交易价?标号是1的股票的最近一次交易价？

选择方式：

- 行和列的区域
  - 标签label（loc）
- 单个值
  - at

Source

```
>>> djidf.loc[1:5,['code','lasttrade']]
    code   lasttrade
1     BA      128.86
2    CAT      101.34
3   CSCO       26.32
4    CVX      116.32
5     DD       70.80

[5 rows x 2 columns]
>>> djidf.loc[1,'lasttrade']
'128.86‘
>>> djidf.at[1,'lasttrade']
'128.86'
```

数据选择

选择方式：

- 行、列和区域
  - 用iloc（位置）
- 取某个值
  - iat

Source

```
>>> djidf.loc[1:5,['code','lasttrade']]
   code  lasttrade
1    BA     128.86
2   CAT     101.34
3  CSCO      26.32
4   CVX     116.32
5    DD      70.80
```

Source

```
>>> djidf.loc[1,'lasttrade']
'128.86'
>>> djidf.at[1,'lasttrade']
'128.86'
```

如果直接写成0:2不加[]则表示列索引即第0和第1列

Source

```
>>> djidf.iloc[1:6,[0,2]]
   code  lasttrade
1    BA     128.86
2   CAT     101.34
3  CSCO      26.32
4   CVX     116.32
5    DD      70.80
```

Source

```
>>> djidf.iloc[1,2]
'128.86'
>>> djidf.iat[1,2]
'128.86'
```

数据选择

美国运通公司2014年的股票信息？进一步寻找美国运通公司2014年收盘价大于等于95的记录？

选择方式：

- 条件筛选

Source

```
>>> quotesdf[quotesdf.index >= u'2014-01-01']
                open   close       high        low   volume
2014-01-02 89.924438  88.49 90.102506  88.420751  5112000
2014-01-03 88.186377  88.77 89.106325  87.671998  3888500
2014-01-06 88.730000  88.73 89.274052  88.413460  2844700
...
2014-03-28 89.531554  89.72 90.811002  89.263763  3138900
           ...   ...    ...       ...       ...
[221 rows x 5 columns]
>>> quotesdf[(quotesdf.index >= u'2014-01-01') & (quotesdf.close >= 95)]
                open   close       high        low   volume
2014-06-09 94.532820  95.02 95.328216  94.105295  3825200
2014-06-18 94.204662  95.01 95.039827  93.538518  2454800
2014-07-03 95.031492  95.29 95.389426  94.673558  1633800
[3 rows x 5 columns]
```

用Python玩转数据

5 简单统计与处理

简单统计与筛选

求道琼斯工业股中30只股票最近一次成交价的平均值？股票最近一次成交价大于等于120的公司名？

Source

```
>>> djidf.mean(columns = 'lasttrade')
lasttrade    91.533667
dtype: float64
>>> djidf[djidf.lasttrade >= 120].name
1                                  The Boeing Company
8                        The Goldman Sachs Group, Inc.
10    International Business Machines Corporation
16                                        3M Company
26                                           Visa Inc.
Name: name, dtype: object
```

简单统计与筛选

统计美国运通公司近一年股票涨和跌分别的天数？

Source

```
>>> len(quotesdf[quotesdf.close > quotesdf.open])
131
>>> len(quotesdf)-131
120
```

统计美国运通公司近一年相邻两天收盘价的涨跌情况？

Source

```
>>> status = np.sign(np.diff(quotesdf.close))
>>> status
array([ 1., -1.,  1., -1.,  1.,  1.,  1.,  1., -1., -1., -1.,  1.,  1.,
        ...
        -1., -1., -1.])
>>> status[np.where( status == 1.)].size
130
>>> status[np.where( status == -1.)].size
120
```

排序

DataFrame的sort()函数已不推荐使用，相同功能推荐使用sort_index()函数

按最近一次成交价对30只道琼斯工业股股票进行排序。根据排序结果列出前三甲公司名。

Source

```
>>> djidf.sort(columns = 'lasttrade')
    code                                  name  lasttrade
3   CSCO                     Cisco Systems, Inc.      26.32
7     GE              General Electric Company      26.46
20   PFE                            Pfizer Inc.      30.34
11  INTC                      Intel Corporation      33.95
...
8     GS        The Goldman Sachs Group, Inc.     189.98
26     V                                Visa Inc.     248.84
[30 rows x 3 columns]
>>> djidf.sort(columns = 'lasttrade')[27:].name
10   International Business Machines Corporation
8                  The Goldman Sachs Group, Inc.
26                                     Visa Inc.
Name: name, dtype: object
```

可以添加sort()函数的ascending属性控制顺序/逆序排序，默认该属性=True，即顺序排列

计数统计

统计2014年1月份的股票开盘天数？

Source

```
>>> t = quotesdf[(quotesdf.index >= '2014-01-01') & (quotesdf.index < '2014-02-01')]
>>> t
                   open  close       high        low    volume
2014-01-02  89.924438  88.49  90.102506  88.420751  5112000
2014-01-03  88.186377  88.77  89.106325  87.671998  3888500
...
2014-01-30  85.741393  85.91  86.267049  85.364508  4259000
2014-01-31  84.577859  84.32  85.202672  84.210906  4778000
[21 rows x 5 columns]
>>> len(t)
21
```

计数统计

统计近一年每个月的股票开盘天数？

File

```
# Filename: quotesmonth.py
import time
...
listtemp = []
for i in range(0,len(quotesdf)):
    temp = time.strptime(quotesdf.index[i],"%Y-%m-%d")
    listtemp.append(temp.tm_mon)
print listtemp
tempdf = quotesdf.copy()
tempdf['month'] = listtemp
print tempdf['month'].value_counts()
```

Output:

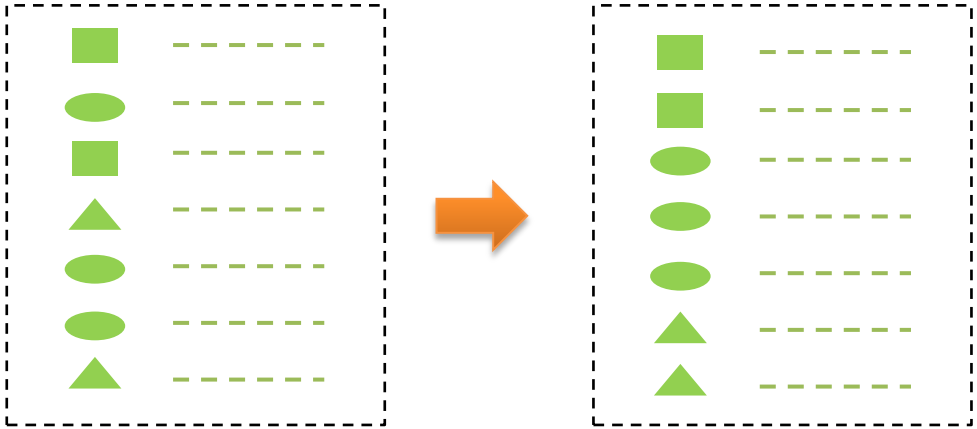
```
10    23
7     22
12    21
9     21
8     21
6     21
5     21
4     21
3     21
1     21
11    19
2     19
dtype: int64
```

6

用Python玩转数据

GROUPING

分组



Grouping的顺序

① Splitting

② Applying

③ Combining

分组

统计近一年每个月的股票开盘天数？

Source

```
>>> tempdf.groupby('month').count()
        open  close  high  low  volume   month
month
1         21     21    21   21      21      21
2         19     19    19   19      19      19
3         21     21    21   21      21      21
...
10        23     23    23   23      23      23
11        20     20    20   20      20      20
12        21     21    21   21      21      21
[12 rows x 6 columns]
>>> tempdf.groupby('month').count().month
```

如果没有特殊处理的话则没有month这一列数据

如果没有month这一列数据，则此处month需要改成其他属性名如open

Output:

```
month
1     21
2     19
3     21
4     21
5     21
6     21
7     22
8     21
9     21
10    23
11    20
12    21
Name: month,
dtype: int64
```

分组

统计近一年每个月的总成交量？

Source

```
>>> tempdf.groupby('month').sum().volume
month
       1      104874000
       2       76173800
       3       71488400
       4       84786400
...
       9       85341400
      10      120822100
      11       67906300
      12       67589400
Name: volume, dtype: float64
```

mean()

min()

max()

...

分组

如果更高效统计近一年每个月的总成交量？

Source

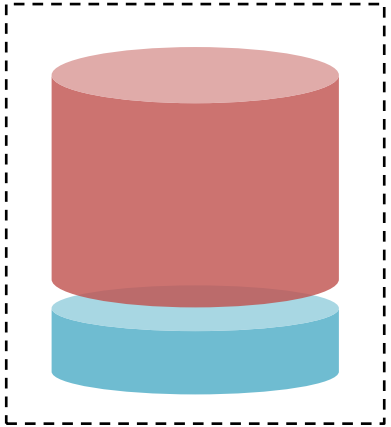
```
>>> g = tempdf.groupby('month')
>>> gvolume = g['volume']
>>> print gvolume.sum()
month
       1      104874000
       2       76173800
       3       71488400
...
      10      120822100
      11       67906300
      12       67589400
Name: volume, dtype: float64
```

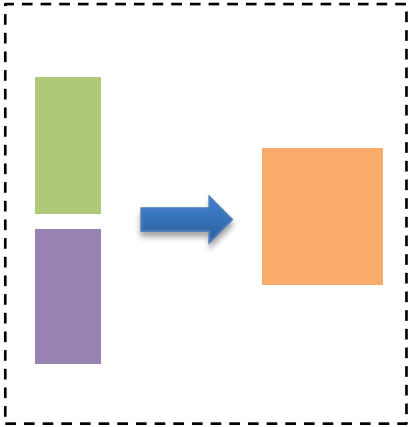
7

用Python玩转数据

MERGE

合并





Merge的形式

- Append
  - 加行到DataFrame
- Concat
  - 连接pandas对象
- Join
  - SQL类型的连接

Append

把美国运通公司2014年1月1日至2014年1月5日间的股票交易信息合并到近一年中前两天的股票信息中？

Source

```
>>> p = quotesdf[:2]
>>> p
                   open  close       high        low   volume
2013-11-18  81.954487  81.45  82.083082  81.271946  3104800
2013-11-19  81.490863  81.57  81.985467  81.382051  2775000
[2 rows x 5 columns]
>>> q = quotesdf[u'2014-01-01':u'2014-01-05']
>>> q
                   open  close       high        low   volume
2014-01-02  89.924438  88.49  90.102506  88.420751  5112000
2014-01-03  88.186377  88.77  89.106325  87.671998  3888500
[2 rows x 5 columns]
>>> p.append(q)
                   open  close       high        low   volume
2013-11-18  81.954487  81.45  82.083082  81.271946  3104800
2013-11-19  81.490863  81.57  81.985467  81.382051  2775000
2014-01-02  89.924438  88.49  90.102506  88.420751  5112000
2014-01-03  88.186377  88.77  89.106325  87.671998  3888500
[4 rows x 5 columns]
```

Concat

将美国运通公司近一年股票数据中的前5个和后5个合并。

Source

```
>>> pieces = [tempdf[:5], tempdf[len(tempdf)-5:]]
>>> pd.concat(pieces)
                 open  close       high        low  volume month
2013-11-18  81.954487  81.45  82.083082  81.271946 3104800    11
2013-11-19  81.490863  81.57  81.985467  81.382051 2775000    11
2013-11-20  82.062317  81.36  82.517340  81.073138 3510500    11
2013-11-21  81.744151  83.05  83.099464  81.684795 4355200    11
2013-11-22  83.019249  82.95  83.029141  82.475152 2690300    11
2014-11-11  92.280000  91.74  92.590000  91.490000 2719800    11
2014-11-12  91.160000  91.55  91.670000  91.100000 3825200    11
2014-11-13  91.700000  91.07  91.700000  90.940000 3637900    11
2014-11-14  91.070000  90.67  91.240000  90.350000 2361600    11
2014-11-17  90.240000  90.13  90.260000  89.650000 2620500    11
[20 rows x 6 columns]
```

Concat

两个不同逻辑结构的对象能否连接？

objs	axis
join	join_axes
keys	levels
names	verify_integrity
ignore_index	

Source

```
>>> piece1 = quotesdf[:3]
>>> piece2 = tempdf[:3]
>>> pd.concat([piece1,piece2],ignore_index = True)
   close       high        low  month       open  volume
0  81.45  82.083082  81.271946    NaN  81.954487 3104800
1  81.57  81.985467  81.382051    NaN  81.490863 2775000
2  81.36  82.517340  81.073138    NaN  82.062317 3510500
3  81.45  82.083082  81.271946     11  81.954487 3104800
4  81.57  81.985467  81.382051     11  81.490863 2775000
5  81.36  82.517340  81.073138     11  82.062317 3510500

[6 rows x 6 columns]
```

Join

code	name	lasttrade
AXP		
KO		

code	month	volume
AXP		
AXP		
KO		
KO		

code	name	lasttrade	month	volume
AXP				
AXP				
KO				
KO				

Join

将美国运通公司和可口可乐公司近一年中每个月的交易总量表（包含公司代码）与30只道琼斯成分股股票信息合并。

code|name|month|volumes

```
>>> djidf
    code                                       name  lasttrade
0    AXP                     American Express Company      90.13
1     BA                           The Boeing Company     128.42
2    CAT                              Caterpillar Inc.     101.70
3   CSCO                          Cisco Systems, Inc.      26.47
4    CVX                          Chevron Corporation     115.75
5     DD          E. I. du Pont de Nemours and Company      70.47
6    DIS                        The Walt Disney Company      90.41
7     GE                      General Electric Company      26.61
8     GS                 The Goldman Sachs Group, Inc.     189.93
9     HD                          The Home Depot, Inc.      98.03
10   IBM  International Business Machines Corporation     164.16
11  INTC                              Intel Corporation      34.24
12   JNJ                         Johnson &amp; Johnson     108.30
13   JPM                       JPMorgan Chase &amp; Co.      60.38
14    KO                         The Coca-Cola Company      42.92
15   MCD                              McDonald's Corp.      95.97
16   MMM                                     3M Company     158.69
17   MRK                          Merck &amp; Co. Inc.      59.46
18  MSFT                         Microsoft Corporation      49.46
19   NKE                                   Nike, Inc.      96.06
20   PFE                                  Pfizer Inc.      30.32
21    PG              The Procter &amp; Gamble Company      87.84
22     T                                AT&amp;T, Inc.      35.85
23   TRV                 The Travelers Companies, Inc.     102.75
24   UNH                UnitedHealth Group Incorporated      96.47
25   UTX                United Technologies Corporation     107.37
26     V                                     Visa Inc.     249.80
27    VZ                    Verizon Communications Inc.      51.40
28   WMT                           Wal-Mart Stores Inc.      83.57
29   XOM                         Exxon Mobil Corporation      95.11

[30 rows x 3 columns]
```

```
>>> AKdf
          volume code  month
month
1      104874000  AXP      1
2       76173800  AXP      2
3       71488400  AXP      3
4       84786400  AXP      4
5       65091900  AXP      5
6       60522000  AXP      6
7       68452900  AXP      7
8       83077800  AXP      8
9       85341400  AXP      9
10     120822100  AXP     10
11      67906300  AXP     11
12      67589400  AXP     12
1      289400900   KO      1
2      385231900   KO      2
3      349073700   KO      3
4      382724700   KO      4
5      213626400   KO      5
6      281965200   KO      6
7      282905200   KO      7
8      247854800   KO      8
9      314443100   KO      9
10     529541900   KO     10
11     265863000   KO     11
12     301144500   KO     12

[24 rows x 3 columns]
```

Join

Source

```
>>> pd.merge(djidf, AKdf, on = 'code')
  code                          name  lasttrade   volume  month
0  AXP  American Express Company      90.13  104874000   1
1  AXP  American Express Company      90.13   76173800   2
2  AXP  American Express Company      90.13   71488400   3
3  AXP  American Express Company      90.13   84786400   4
4  AXP  American Express Company      90.13   65091900   5
...
19  KO      The Coca-Cola Company     42.92  247854800   8
20  KO      The Coca-Cola Company     42.92  314443100   9
21  KO      The Coca-Cola Company     42.92  529541900  10
22  KO      The Coca-Cola Company     42.92  265863000  11
23  KO      The Coca-Cola Company     42.92  301144500  12
[24 rows x 5 columns]
>>> pd.merge(djidf,AKdf,on = 'code').drop(['lasttrade'],axis = 1)
```

```
   code                     name     volume  month
0   AXP  American Express Company  104874000      1
1   AXP  American Express Company   76173800      2
2   AXP  American Express Company   71488400      3
3   AXP  American Express Company   84786400      4
4   AXP  American Express Company   65091900      5
5   AXP  American Express Company   60522000      6
6   AXP  American Express Company   68452900      7
7   AXP  American Express Company   83077800      8
8   AXP  American Express Company   85341400      9
9   AXP  American Express Company  120822100     10
10  AXP  American Express Company   67906300     11
11  AXP  American Express Company   67589400     12
12   KO     The Coca-Cola Company  289400900      1
13   KO     The Coca-Cola Company  385231900      2
14   KO     The Coca-Cola Company  349073700      3
15   KO     The Coca-Cola Company  382724700      4
16   KO     The Coca-Cola Company  213626400      5
17   KO     The Coca-Cola Company  281965200      6
18   KO     The Coca-Cola Company  282905200      7
19   KO     The Coca-Cola Company  247854800      8
20   KO     The Coca-Cola Company  314443100      9
21   KO     The Coca-Cola Company  529541900     10
22   KO     The Coca-Cola Company  265863000     11
23   KO     The Coca-Cola Company  301144500     12

[24 rows x 4 columns]
```

merge函数的参数

left	right	how
on	left_on	right_on
left_index	right_index	sort
suffixes	copy	