

```
In [1]: 1 from IPython.core.interactiveshell import InteractiveShell
        2 InteractiveShell.ast_node_interactivity = "all"
```

时间序列:索引为时间

字符串序列:值为字符串

```
In [2]: 1 #导入相关库
        2 import pandas as pd
        3 import numpy as np
        4 import time
```

```
In [5]: 1 time.time()    #当前时间，返回从1970年到现在经过的秒数
```

```
Out[5]: 1645062416.9998336
```

```
In [6]: 1 pd.Timestamp(time.time(),unit='s')    #转换为标准时间，时间戳的形式，时刻
```

```
Out[6]: Timestamp('2022-02-17 01:51:55.991060734')
```

在计算机中，时间实际上是用数字表示的。

我们把1970年1月1日 00:00:00 UTC+00:00时区的时刻称为epoch time(纪元)，记为0（1970年以前的时间timestamp为负数），当前时间就是相对于epoch time的秒数，称为timestamp。

timestamp的值与时区毫无关系，因为timestamp一旦确定，其UTC时间就确定了，转换到任意时区的时间也是完全确定的，这就是为什么计算机存储的当前时间是以timestamp表示的，因为全球各地的计算机在任意时刻的timestamp都是完全相同的（假定时间已校准）。

1 时间的生成

1.1 pd.Timestamp

```
In [7]: 1 time.time()
```

```
Out[7]: 1645062863.8511145
```

```
In [8]: 1 pd.Timestamp(time.time(),unit='s')
```

```
Out[8]: Timestamp('2022-02-17 01:54:30.092604876')
```

```
In [9]: 1 pd.Timestamp(year=2021,month=3,day=15,hour=10,minute=30,second=25) #关键字参数
```

```
Out[9]: Timestamp('2021-03-15 10:30:25')
```

```
In [10]: 1 pd.Timestamp(2020,10,22,13,52,12) #位置参数
```

```
Out[10]: Timestamp('2020-10-22 13:52:12')
```

```
In [11]: 1 pd.Timestamp('2020-11-25')
```

```
Out[11]: Timestamp('2020-11-25 00:00:00')
```

```
In [12]: 1 pd.Timestamp('2020/11/25')
```

```
Out[12]: Timestamp('2020-11-25 00:00:00')
```

1.2 pd.to_datetime

```
In [15]: 1 pd.to_datetime('2020/10/15 13:28:56')#单个转换为时间戳
```

```
Out[15]: Timestamp('2020-10-15 13:28:56')
```

```
In [16]: 1 pd.to_datetime(['2020/10/15 13:28:56','2021/10/15 13:28:56']) #多个时间转换为时间戳
```

```
Out[16]: DatetimeIndex(['2020-10-15 13:28:56', '2021-10-15 13:28:56'], dtype='datetime64[ns]', freq=None)
```

```
In [17]: 1 pd.to_datetime('2020-8-12')
```

```
Out[17]: Timestamp('2020-08-12 00:00:00')
```

```
In [18]: 1 pd.to_datetime('210628', format='%y%m%d') #指定格式
```

```
Out[18]: Timestamp('2021-06-28 00:00:00')
```

```
In [19]: 1 pd.to_datetime('21年6月28日', format='%y年%m月%d日') #要非常严格的对齐
```

```
Out[19]: Timestamp('2021-06-28 00:00:00')
```

类型	描述
%Y	四位的年份
%y	两位的年份
%m	两位的月份
%d	两位的日期号
%H	小时，24小时制00-23
%I	小时，12小时制01-12
%M	两位的分钟00-59
%S	秒00-61，其中60,61是闰秒
%w	星期日期0-6，其中0表示星期日
%U	一年中的星期数00-53。以星期天为每周第一天，一年中第一个星期天前日期作为第“0”周
%W	一年中的星期数00-53。以星期一为每周第一天，一年中第一个星期一前的日期作为第“0”周
%z	格式为+HHMM或者-HHMM的UTC时区偏移；如果没有时区则为空。
%F	%Y-%m-%d的简写
%D	%m/%d/%y的简写

```
In [20]: 1 pd.to_datetime('2020--8--12', format='%Y--%m--%d')
```

```
Out[20]: Timestamp('2020-08-12 00:00:00')
```

```
In [21]: 1 pd.to_datetime('21&06&28', format='%y&%m&%d')
```

```
Out[21]: Timestamp('2021-06-28 00:00:00')
```

```
In [22]: 1 pd.to_datetime(['2020--8--12', '2021--8--12'], format='%Y--%m--%d')
```

```
Out[22]: DatetimeIndex(['2020-08-12', '2021-08-12'], dtype='datetime64[ns]', freq=None)
```

```
In [ ]: 1
```

1.3 pd.date_range

```
In [23]: 1 pd.date_range('2019-1-20', freq='T', periods=5) #固定跨度时间的生成
```

```
Out[23]: DatetimeIndex(['2019-01-20 00:00:00', '2019-01-20 00:01:00',  
                        '2019-01-20 00:02:00', '2019-01-20 00:03:00',  
                        '2019-01-20 00:04:00'],  
                        dtype='datetime64[ns]', freq='T')
```

- freq:

- Y: 年
- M: 月
- D: 天
- H: 小时
- T: 分钟
- S: 秒
- 其他频率时间单位: https://pandas.pydata.org/pandas-docs/stable/user_guide/timeseries.html#timeseries-offset-aliases
(https://pandas.pydata.org/pandas-docs/stable/user_guide/timeseries.html#timeseries-offset-aliases)

```
In [24]: 1 pd.date_range('2019-1-20', '2019-1-25', freq='12H') #两边闭合
```

```
Out[24]: DatetimeIndex(['2019-01-20 00:00:00', '2019-01-20 12:00:00',  
                        '2019-01-21 00:00:00', '2019-01-21 12:00:00',  
                        '2019-01-22 00:00:00', '2019-01-22 12:00:00',  
                        '2019-01-23 00:00:00', '2019-01-23 12:00:00',  
                        '2019-01-24 00:00:00', '2019-01-24 12:00:00',  
                        '2019-01-25 00:00:00'],  
                        dtype='datetime64[ns]', freq='12H')
```

2 时间格式转换

不规则的字符串日期 $\longrightarrow$ `pd.to_datetime()` $\longrightarrow$ 日期类型的数据

已经有了datetime类型的日期字段 $\longrightarrow$ `.dt.strftime()` $\longrightarrow$ 字符串日期

要转化成字符串日期（可能不是常见的日期规格）

```
In [25]: 1 data=pd.read_excel("上证指数收盘价历史数据(1990-2017.9).xlsx",header=0)
          2 data.head()
```

Out[25]:

	日期	收盘价	最高价	最低价	开盘价	涨跌额	涨跌幅	成交量	成交额
0	2017-09-14	3371.4256	3391.6435	3361.3335	3383.4700	-12.7214	-0.3759	221306487	256744421274
1	2017-09-13	3384.1470	3387.1397	3366.5412	3374.7185	4.659	0.1379	194550715	230733408867
2	2017-09-12	3379.4880	3391.0694	3370.8519	3381.4870	3.0692	0.0909	272910319	325024951982
3	2017-09-11	3376.4188	3384.8100	3360.0462	3365.3506	11.1762	0.3321	219011019	256888810547
4	2017-09-08	3365.2426	3380.8898	3353.6876	3364.4275	-0.2548	-0.0076	198405184	235160227356

```
In [26]: 1 data['日期']
```

Out[26]:

0	2017-09-14
1	2017-09-13
2	2017-09-12
3	2017-09-11
4	2017-09-08
...	
6537	1990-12-25
6538	1990-12-24
6539	1990-12-21
6540	1990-12-20
6541	1990-12-19

Name: 日期, Length: 6542, dtype: datetime64[ns]

```
In [27]: 1 data.日期.dt.strftime('%y/%m/%d')
```

```
Out[27]: 0      17/09/14
          1      17/09/13
          2      17/09/12
          3      17/09/11
          4      17/09/08
          ...
        6537    90/12/25
        6538    90/12/24
        6539    90/12/21
        6540    90/12/20
        6541    90/12/19
        Name: 日期, Length: 6542, dtype: object
```

```
In [28]: 1 data.日期.dt.strftime('%y-%m')
```

```
Out[28]: 0      17-09
          1      17-09
          2      17-09
          3      17-09
          4      17-09
          ...
        6537    90-12
        6538    90-12
        6539    90-12
        6540    90-12
        6541    90-12
        Name: 日期, Length: 6542, dtype: object
```

```
In [29]: 1 data.日期.dt.strftime('%Y')
```

```
Out[29]: 0      2017
          1      2017
          2      2017
          3      2017
          4      2017
          ...
          6537   1990
          6538   1990
          6539   1990
          6540   1990
          6541   1990
          Name: 日期, Length: 6542, dtype: object
```

```
In [30]: 1 data.日期.dt.strftime('%m')
```

```
Out[30]: 0      09
          1      09
          2      09
          3      09
          4      09
          ..
          6537   12
          6538   12
          6539   12
          6540   12
          6541   12
          Name: 日期, Length: 6542, dtype: object
```

```
In [31]: 1 data.日期.dt.strftime('%d')
```

```
Out[31]: 0      14
          1      13
          2      12
          3      11
          4       8
          ..
        6537    25
        6538    24
        6539    21
        6540    20
        6541    19
        Name: 日期, Length: 6542, dtype: object
```

3 提取时间元素

从datetime中提取特定的时间单位元素：

```
In [34]: 1 data['日期']
```

```
Out[34]: 0      2017-09-14
          1      2017-09-13
          2      2017-09-12
          3      2017-09-11
          4      2017-09-08
          ...
        6537    1990-12-25
        6538    1990-12-24
        6539    1990-12-21
        6540    1990-12-20
        6541    1990-12-19
        Name: 日期, Length: 6542, dtype: datetime64[ns]
```

In [35]:

1 data

Out[35]:

	日期	收盘价	最高价	最低价	开盘价	涨跌额	涨跌幅	成交量	成交额
0	2017-09-14	3371.4256	3391.6435	3361.3335	3383.4700	-12.7214	-0.3759	221306487	256744421274
1	2017-09-13	3384.1470	3387.1397	3366.5412	3374.7185	4.659	0.1379	194550715	230733408867
2	2017-09-12	3379.4880	3391.0694	3370.8519	3381.4870	3.0692	0.0909	272910319	325024951982
3	2017-09-11	3376.4188	3384.8100	3360.0462	3365.3506	11.1762	0.3321	219011019	256888810547
4	2017-09-08	3365.2426	3380.8898	3353.6876	3364.4275	-0.2548	-0.0076	198405184	235160227356
...	...	...	...	...	...	...	...	...	...
6537	1990-12-25	120.2500	120.2500	114.5500	120.0900	5.7	4.976	15	6000
6538	1990-12-24	114.5500	114.5500	109.1300	113.5700	5.42	4.9666	32	31000
6539	1990-12-21	109.1300	109.1300	103.7300	109.0700	4.74	4.5407	28	16000
6540	1990-12-20	104.3900	104.3900	99.9800	104.3000	4.41	4.4109	197	84000
6541	1990-12-19	99.9800	99.9800	95.7900	96.0500	None	None	1260	494000

6542 rows × 9 columns

```
In [36]: 1 data.日期.dt.year #把年份提取出来是int类型
```

```
Out[36]: 0      2017
          1      2017
          2      2017
          3      2017
          4      2017
          ...
        6537    1990
        6538    1990
        6539    1990
        6540    1990
        6541    1990
Name: 日期, Length: 6542, dtype: int64
```

```
In [37]: 1 data.日期.dt.month #把月份取出来是int类型
```

```
Out[37]: 0      9
          1      9
          2      9
          3      9
          4      9
          ..
        6537    12
        6538    12
        6539    12
        6540    12
        6541    12
Name: 日期, Length: 6542, dtype: int64
```

```
In [38]: 1 data.日期.dt.day #把日取出来是int类型
```

```
Out[38]: 0      14
          1      13
          2      12
          3      11
          4       8
          ..
        6537    25
        6538    24
        6539    21
        6540    20
        6541    19
        Name: 日期, Length: 6542, dtype: int64
```

```
In [39]: 1 data.日期.dt.date #把日期调出来, 是一个object类型
```

```
Out[39]: 0      2017-09-14
          1      2017-09-13
          2      2017-09-12
          3      2017-09-11
          4      2017-09-08
          ...
        6537    1990-12-25
        6538    1990-12-24
        6539    1990-12-21
        6540    1990-12-20
        6541    1990-12-19
        Name: 日期, Length: 6542, dtype: object
```

```
In [40]: 1 data.日期.dt.time #把时间调出来, 是一个object类型
```

```
Out[40]: 0      00:00:00
1      00:00:00
2      00:00:00
3      00:00:00
4      00:00:00
...
6537   00:00:00
6538   00:00:00
6539   00:00:00
6540   00:00:00
6541   00:00:00
Name: 日期, Length: 6542, dtype: object
```

```
In [44]: 1 data.日期.dt.dayofweek+1
```

```
Out[44]: 0      4
1      3
2      2
3      1
4      5
..
6537   2
6538   1
6539   5
6540   4
6541   3
Name: 日期, Length: 6542, dtype: int64
```

4 日期索引序列

DatetimeIndex 的主要作用是之一一是用作 Pandas 对象的索引, 使用它作为索引除了拥有普通索引对象的所有基本功能外, 还拥有简化频率处理的高级时间序列方法

In [45]:

```
1 data.head(3)
```

Out[45]:

	日期	收盘价	最高价	最低价	开盘价	涨跌额	涨跌幅	成交量	成交额
0	2017-09-14	3371.4256	3391.6435	3361.3335	3383.4700	-12.7214	-0.3759	221306487	256744421274
1	2017-09-13	3384.1470	3387.1397	3366.5412	3374.7185	4.659	0.1379	194550715	230733408867
2	2017-09-12	3379.4880	3391.0694	3370.8519	3381.4870	3.0692	0.0909	272910319	325024951982

In [46]:

```
1 data.set_index('日期', inplace=True)
2 data.head(3)
```

Out[46]:

	收盘价	最高价	最低价	开盘价	涨跌额	涨跌幅	成交量	成交额
日期								
2017-09-14	3371.4256	3391.6435	3361.3335	3383.4700	-12.7214	-0.3759	221306487	256744421274
2017-09-13	3384.1470	3387.1397	3366.5412	3374.7185	4.659	0.1379	194550715	230733408867
2017-09-12	3379.4880	3391.0694	3370.8519	3381.4870	3.0692	0.0909	272910319	325024951982

```
In [47]: 1 data['收盘价']
```

```
Out[47]: 日期
2017-09-14    3371.4256
2017-09-13    3384.1470
2017-09-12    3379.4880
2017-09-11    3376.4188
2017-09-08    3365.2426
...
1990-12-25     120.2500
1990-12-24     114.5500
1990-12-21     109.1300
1990-12-20     104.3900
1990-12-19      99.9800
Name: 收盘价, Length: 6542, dtype: float64
```

```
In [48]: 1 data.index
```

```
Out[48]: DatetimeIndex(['2017-09-14', '2017-09-13', '2017-09-12', '2017-09-11',
                        '2017-09-08', '2017-09-07', '2017-09-06', '2017-09-05',
                        '2017-09-04', '2017-09-01',
                        ...
                        '1991-01-02', '1990-12-31', '1990-12-28', '1990-12-27',
                        '1990-12-26', '1990-12-25', '1990-12-24', '1990-12-21',
                        '1990-12-20', '1990-12-19'],
                        dtype='datetime64[ns]', name='日期', length=6542, freq=None)
```

4.1 提取时间元素

```
In [49]: 1 data.index.day
```

```
Out[49]: Int64Index([14, 13, 12, 11,  8,  7,  6,  5,  4,  1,
                    ...
                    2, 31, 28, 27, 26, 25, 24, 21, 20, 19],
                    dtype='int64', name='日期', length=6542)
```

```
In [50]: 1 data.index.quarter
```

```
Out[50]: Int64Index([3, 3, 3, 3, 3, 3, 3, 3, 3,
...
1, 4, 4, 4, 4, 4, 4, 4, 4],
dtype='int64', name='日期', length=6542)
```

```
In [51]: 1 data.index.is_month_start #判断是不是每月的第一天
```

```
Out[51]: array([False, False, False, ..., False, False, False])
```

4.2 索引和切片

4.2.1 常规操作

```
In [52]: 1 data.head()
```

```
Out[52]:
```

	收盘价	最高价	最低价	开盘价	涨跌额	涨跌幅	成交量	成交额
日期								
2017-09-14	3371.4256	3391.6435	3361.3335	3383.4700	-12.7214	-0.3759	221306487	256744421274
2017-09-13	3384.1470	3387.1397	3366.5412	3374.7185	4.659	0.1379	194550715	230733408867
2017-09-12	3379.4880	3391.0694	3370.8519	3381.4870	3.0692	0.0909	272910319	325024951982
2017-09-11	3376.4188	3384.8100	3360.0462	3365.3506	11.1762	0.3321	219011019	256888810547
2017-09-08	3365.2426	3380.8898	3353.6876	3364.4275	-0.2548	-0.0076	198405184	235160227356

In [53]: 1 data.loc['2017-09-14'] #选单个

Out[53]:

	收盘价	最高价	最低价	开盘价	涨跌额	涨跌幅	成交量	成交额
日期								
2017-09-14	3371.4256	3391.6435	3361.3335	3383.47	-12.7214	-0.3759	221306487	256744421274

In [54]: 1 data.loc['2017-09-14':'2017-09-12'] # 切片，选多个，两边闭合

Out[54]:

	收盘价	最高价	最低价	开盘价	涨跌额	涨跌幅	成交量	成交额
日期								

In [55]: 1 data.iloc[:,2,:] #隐式索引

Out[55]:

	收盘价	最高价	最低价	开盘价	涨跌额	涨跌幅	成交量	成交额
日期								
2017-09-14	3371.4256	3391.6435	3361.3335	3383.4700	-12.7214	-0.3759	221306487	256744421274
2017-09-13	3384.1470	3387.1397	3366.5412	3374.7185	4.659	0.1379	194550715	230733408867

In []: 1

In [56]: 1 #常规操作选出2017-09
2 data['年月']=data.index.strftime('%Y/%m')

```
In [57]: 1 data[data['年月']=='2017/09']
```

Out[57]:

	收盘价	最高价	最低价	开盘价	涨跌额	涨跌幅	成交量	成交额	年月
日期									
2017-09-14	3371.4256	3391.6435	3361.3335	3383.4700	-12.7214	-0.3759	221306487	256744421274	2017/09
2017-09-13	3384.1470	3387.1397	3366.5412	3374.7185	4.659	0.1379	194550715	230733408867	2017/09
2017-09-12	3379.4880	3391.0694	3370.8519	3381.4870	3.0692	0.0909	272910319	325024951982	2017/09
2017-09-11	3376.4188	3384.8100	3360.0462	3365.3506	11.1762	0.3321	219011019	256888810547	2017/09
2017-09-08	3365.2426	3380.8898	3353.6876	3364.4275	-0.2548	-0.0076	198405184	235160227356	2017/09
2017-09-07	3365.4974	3387.7956	3363.1765	3383.6281	-19.8914	-0.5876	221118685	264037023512	2017/09
2017-09-06	3385.3888	3391.0105	3364.7645	3372.4277	1.0718	0.0317	229090785	262034653491	2017/09
2017-09-05	3384.3170	3390.8233	3371.5706	3377.1968	4.734	0.1401	216552946	245596608276	2017/09
2017-09-04	3379.5830	3381.4027	3359.1309	3369.7185	12.4636	0.3702	267427849	296348432251	2017/09
2017-09-01	3367.1194	3381.9252	3358.4724	3365.9913	6.3091	0.1877	282497584	313722092738	2017/09

4.2.2 高级操作

```
In [58]: 1 data.loc['2017-09']
```

```
Out[58]:
```

	收盘价	最高价	最低价	开盘价	涨跌额	涨跌幅	成交量	成交额	年月
日期									
2017-09-14	3371.4256	3391.6435	3361.3335	3383.4700	-12.7214	-0.3759	221306487	256744421274	2017/09
2017-09-13	3384.1470	3387.1397	3366.5412	3374.7185	4.659	0.1379	194550715	230733408867	2017/09
2017-09-12	3379.4880	3391.0694	3370.8519	3381.4870	3.0692	0.0909	272910319	325024951982	2017/09
2017-09-11	3376.4188	3384.8100	3360.0462	3365.3506	11.1762	0.3321	219011019	256888810547	2017/09
2017-09-08	3365.2426	3380.8898	3353.6876	3364.4275	-0.2548	-0.0076	198405184	235160227356	2017/09
2017-09-07	3365.4974	3387.7956	3363.1765	3383.6281	-19.8914	-0.5876	221118685	264037023512	2017/09
2017-09-06	3385.3888	3391.0105	3364.7645	3372.4277	1.0718	0.0317	229090785	262034653491	2017/09
2017-09-05	3384.3170	3390.8233	3371.5706	3377.1968	4.734	0.1401	216552946	245596608276	2017/09
2017-09-04	3379.5830	3381.4027	3359.1309	3369.7185	12.4636	0.3702	267427849	296348432251	2017/09
2017-09-01	3367.1194	3381.9252	3358.4724	3365.9913	6.3091	0.1877	282497584	313722092738	2017/09

In [59]: 1 data.loc['2017'] #特有的方法

Out[59]:

	收盘价	最高价	最低价	开盘价	涨跌额	涨跌幅	成交量	成交额	年月
日期									
2017-09-14	3371.4256	3391.6435	3361.3335	3383.4700	-12.7214	-0.3759	221306487	256744421274	2017/09
2017-09-13	3384.1470	3387.1397	3366.5412	3374.7185	4.659	0.1379	194550715	230733408867	2017/09
2017-09-12	3379.4880	3391.0694	3370.8519	3381.4870	3.0692	0.0909	272910319	325024951982	2017/09
2017-09-11	3376.4188	3384.8100	3360.0462	3365.3506	11.1762	0.3321	219011019	256888810547	2017/09
2017-09-08	3365.2426	3380.8898	3353.6876	3364.4275	-0.2548	-0.0076	198405184	235160227356	2017/09
...	...	...	...	...	...	...	...	...	...
2017-01-09	3171.2362	3173.1360	3147.7351	3148.5317	16.9152	0.5363	171714075	192110579318	2017/01
2017-01-06	3154.3210	3172.0347	3153.0253	3163.7761	-11.0899	-0.3503	183708966	207296038444	2017/01
2017-01-05	3165.4109	3168.5021	3154.2810	3157.9063	6.6169	0.2095	174727645	199692026733	2017/01
2017-01-04	3158.7940	3160.1029	3130.1145	3133.7873	22.8732	0.7294	167860850	195914292648	2017/01
2017-01-03	3135.9208	3136.4558	3105.3085	3105.3085	32.2835	1.0402	141567187	159887139472	2017/01

173 rows × 9 columns

5 时间偏移：DateOffset对象

DateOffset 从名称中就可以看出来是要做日期偏移的

```
In [60]: 1 d=pd.Timestamp('2021-03-25')
          2 d
```

```
Out[60]: Timestamp('2021-03-25 00:00:00')
```

```
In [61]: 1 d=pd.DateOffset(years=1)
```

```
Out[61]: Timestamp('2020-03-25 00:00:00')
```

```
In [62]: 1 d+pd.DateOffset(years=1)
```

```
Out[62]: Timestamp('2022-03-25 00:00:00')
```

```
In [63]: 1 d+pd.DateOffset(years=1, months=5)
```

```
Out[63]: Timestamp('2022-08-25 00:00:00')
```

```
In [64]: 1 d+pd.DateOffset(years=1, days=5)
```

```
Out[64]: Timestamp('2022-03-30 00:00:00')
```

```
In [65]: 1 pd.Timestamp(time.time(), unit='s') +pd.DateOffset(hours=8) #时区的转换
```

```
Out[65]: Timestamp('2022-02-17 10:47:49.246124029')
```

```
In [ ]: 1
```

6 resample：重采样

resample 表示根据日期维度进行数据聚合，可以按照分钟、小时、工作日、周、月、年等来作为日期维度

比如我们想要查看每一年的成交量

```
In [66]: 1 data=pd.read_excel("上证指数收盘价历史数据(1990-2017.9).xlsx",header=0)
          2 data
```

Out[66]:

	日期	收盘价	最高价	最低价	开盘价	涨跌额	涨跌幅	成交量	成交额
0	2017-09-14	3371.4256	3391.6435	3361.3335	3383.4700	-12.7214	-0.3759	221306487	256744421274
1	2017-09-13	3384.1470	3387.1397	3366.5412	3374.7185	4.659	0.1379	194550715	230733408867
2	2017-09-12	3379.4880	3391.0694	3370.8519	3381.4870	3.0692	0.0909	272910319	325024951982
3	2017-09-11	3376.4188	3384.8100	3360.0462	3365.3506	11.1762	0.3321	219011019	256888810547
4	2017-09-08	3365.2426	3380.8898	3353.6876	3364.4275	-0.2548	-0.0076	198405184	235160227356
...	...	...	...	...	...	...	...	...	...
6537	1990-12-25	120.2500	120.2500	114.5500	120.0900	5.7	4.976	15	6000
6538	1990-12-24	114.5500	114.5500	109.1300	113.5700	5.42	4.9666	32	31000
6539	1990-12-21	109.1300	109.1300	103.7300	109.0700	4.74	4.5407	28	16000
6540	1990-12-20	104.3900	104.3900	99.9800	104.3000	4.41	4.4109	197	84000
6541	1990-12-19	99.9800	99.9800	95.7900	96.0500	None	None	1260	494000

6542 rows × 9 columns

6.1 常规方法

```
In [67]: 1 data['年份']=data['日期'].dt.year  
2 data.head()
```

Out[67]:

	日期	收盘价	最高价	最低价	开盘价	涨跌额	涨跌幅	成交量	成交额	年份
0	2017-09-14	3371.4256	3391.6435	3361.3335	3383.4700	-12.7214	-0.3759	221306487	256744421274	2017
1	2017-09-13	3384.1470	3387.1397	3366.5412	3374.7185	4.659	0.1379	194550715	230733408867	2017
2	2017-09-12	3379.4880	3391.0694	3370.8519	3381.4870	3.0692	0.0909	272910319	325024951982	2017
3	2017-09-11	3376.4188	3384.8100	3360.0462	3365.3506	11.1762	0.3321	219011019	256888810547	2017
4	2017-09-08	3365.2426	3380.8898	3353.6876	3364.4275	-0.2548	-0.0076	198405184	235160227356	2017

```
In [68]: 1 data.groupby('年份')['成交量'].sum()
```

```
Out[68]: 年份
1990      1884
1991    1258450
1992    18243640
1993    153907556
1994    686346172
1995    517133001
1996    1178384863
1997    1272013024
1998    1457521860
1999    2388494426
2000    3434397008
2001    2970147444
2002    2349106210
2003    3130995928
2004    3853626296
2005    3984953878
2006    10282524705
2007    24323472663
2008    16283039153
2009    33637048392
2010    25909513462
2011    21109960286
2012    18855913817
2013    26459332613
2014    42386825292
2015    101470359632
2016    44635983674
2017    32067658038
Name: 成交量, dtype: int64
```

6.2 高级方法

In [69]:

1

data.set_index('日期', inplace=True)

2

data.head(3)

Out[69]:

	收盘价	最高价	最低价	开盘价	涨跌额	涨跌幅	成交量	成交额	年份
日期									
2017-09-14	3371.4256	3391.6435	3361.3335	3383.4700	-12.7214	-0.3759	221306487	256744421274	2017
2017-09-13	3384.1470	3387.1397	3366.5412	3374.7185	4.659	0.1379	194550715	230733408867	2017
2017-09-12	3379.4880	3391.0694	3370.8519	3381.4870	3.0692	0.0909	272910319	325024951982	2017

```
In [70]: 1 data.resample('Y')['成交量'].sum()    #重采样，表示对年份做groupby  
        2 #重采样首先要把日期设置为索引, 然后再进行年份的选择, 对成交量求和
```

```
Out[70]: 日期  
1990-12-31      1884  
1991-12-31    1258450  
1992-12-31    18243640  
1993-12-31    153907556  
1994-12-31    686346172  
1995-12-31    517133001  
1996-12-31    1178384863  
1997-12-31    1272013024  
1998-12-31    1457521860  
1999-12-31    2388494426  
2000-12-31    3434397008  
2001-12-31    2970147444  
2002-12-31    2349106210  
2003-12-31    3130995928  
2004-12-31    3853626296  
2005-12-31    3984953878  
2006-12-31    10282524705  
2007-12-31    24323472663  
2008-12-31    16283039153  
2009-12-31    33637048392  
2010-12-31    25909513462  
2011-12-31    21109960286  
2012-12-31    18855913817  
2013-12-31    26459332613  
2014-12-31    42386825292  
2015-12-31    101470359632  
2016-12-31    44635983674  
2017-12-31    32067658038  
Freq: A-DEC, Name: 成交量, dtype: int64
```

```
In [71]: 1 data.sort_index()
```

Out[71]:

	收盘价	最高价	最低价	开盘价	涨跌额	涨跌幅	成交量	成交额	年份
日期									
1990-12-19	99.9800	99.9800	95.7900	96.0500	None	None	1260	494000	1990
1990-12-20	104.3900	104.3900	99.9800	104.3000	4.41	4.4109	197	84000	1990
1990-12-21	109.1300	109.1300	103.7300	109.0700	4.74	4.5407	28	16000	1990
1990-12-24	114.5500	114.5500	109.1300	113.5700	5.42	4.9666	32	31000	1990
1990-12-25	120.2500	120.2500	114.5500	120.0900	5.7	4.976	15	6000	1990
...	...	...	...	...	...	...	...	...	...
2017-09-08	3365.2426	3380.8898	3353.6876	3364.4275	-0.2548	-0.0076	198405184	235160227356	2017
2017-09-11	3376.4188	3384.8100	3360.0462	3365.3506	11.1762	0.3321	219011019	256888810547	2017
2017-09-12	3379.4880	3391.0694	3370.8519	3381.4870	3.0692	0.0909	272910319	325024951982	2017
2017-09-13	3384.1470	3387.1397	3366.5412	3374.7185	4.659	0.1379	194550715	230733408867	2017
2017-09-14	3371.4256	3391.6435	3361.3335	3383.4700	-12.7214	-0.3759	221306487	256744421274	2017

6542 rows × 9 columns

```
In [72]: 1 data.sort_index().resample('M')['成交量'].sum()
```

```
Out[72]: 日期
1990-12-31      1884
1991-01-31     67197
1991-02-28     50982
1991-03-31     24528
1991-04-30     70671
...
2017-05-31    3228021196
2017-06-30    3204652287
2017-07-31    4187623651
2017-08-31    5059863051
2017-09-30    2322871573
Freq: M, Name: 成交量, Length: 322, dtype: int64
```

```
In [73]: 1 data['月份']=data.sort_index().index.month
2 data.groupby('月份')['成交量'].sum()
```

```
Out[73]: 月份
1      35818802525
2      32830630556
3      42478593395
4      40991989388
5      37245725582
6      33335440160
7      34998491068
8      37154797084
9      32952413126
10     30854917998
11     30919199219
12     35237163266
Name: 成交量, dtype: int64
```

```
In [ ]: 1
```

