??????

August 9, 2019

```
In [1]: from datetime import datetime

In [2]: now=datetime.now()

In [3]: now

Out[3]: datetime.datetime(2019, 7, 23, 16, 16, 6, 257383)

In [4]: now.year

Out[4]: 2019

In [5]: delta=datetime(2011,1,7)-datetime(2008,6,24,8,15)

In [6]: delta

Out[6]: datetime.timedelta(days=926, seconds=56700)

In [7]: delta.days

Out[7]: 926

In [8]: delta.seconds

Out[8]: 56700

In [9]: from datetime import timedelta

In [10]: start=datetime(2011,1,7)

In [11]: start+timedelta(12)

Out[11]: datetime.datetime(2011, 1, 19, 0, 0)

In [12]: start-2*timedelta(12)

Out[12]: datetime.datetime(2010, 12, 14, 0, 0)

In [13]: # timedelta datetime

In [14]: stamp=datetime(2011,1,3)
In [15]: str(stamp)

Out[15]: '2011-01-03 00:00:00'

In [16]: stamp.strftime('%Y-%m-%d')

Out[16]: '2011-01-03'

In [17]: value='2011-01-03'

In [18]: datetime.strptime(value, '%Y-%m-%d')

Out[18]: datetime.datetime(2011, 1, 3, 0, 0)

In [19]: from dateutil.parser import parse

In [20]: parse('2011-01-03')

Out[20]: datetime.datetime(2011, 1, 3, 0, 0)

In [25]: # dateutil !!!

In [26]: import pandas as pd

In [29]: from datetime import datetime

In [33]: dates=[datetime(2011,1,2),datetime(2011,1,5),datetime(2011,1,7),datetime(2011,1,8),da

In [36]: from pandas import Series, DataFrame
         import numpy as np

In [37]: ts=Series(np.random.randn(6),index=dates)

In [38]: ts

Out[38]: 2011-01-02   -0.402102
         2011-01-05    0.526677
         2011-01-07   -0.968652
         2011-01-08   -0.825882
         2011-01-10    1.713213
         2011-01-12   -0.341772
         dtype: float64

In [39]: type(ts)

Out[39]: pandas.core.series.Series

In [40]: ts.index

Out[40]: DatetimeIndex(['2011-01-02', '2011-01-05', '2011-01-07', '2011-01-08',
                        '2011-01-10', '2011-01-12'],
                       dtype='datetime64[ns]', freq=None)
```

```
In [41]: ts+ts[::2]

Out[41]: 2011-01-02   -0.804204
         2011-01-05         NaN
         2011-01-07   -1.937304
         2011-01-08         NaN
         2011-01-10    3.426426
         2011-01-12         NaN
         dtype: float64

In [42]: # datetime64

In [43]: ts.index.dtype

Out[43]: dtype('<M8[ns]')

In [45]: stamp=ts.index[0]

In [46]: stamp

Out[46]: Timestamp('2011-01-02 00:00:00')

In [47]: stamp=ts.index[2]

In [48]: ts[stamp]

Out[48]: -0.968652248787753

In [49]: ts[datetime(2011,1,7):]

Out[49]: 2011-01-07   -0.968652
         2011-01-08   -0.825882
         2011-01-10    1.713213
         2011-01-12   -0.341772
         dtype: float64

In [50]: ts

Out[50]: 2011-01-02   -0.402102
         2011-01-05    0.526677
         2011-01-07   -0.968652
         2011-01-08   -0.825882
         2011-01-10    1.713213
         2011-01-12   -0.341772
         dtype: float64

In [51]: ts.truncate(after='1/9/2011')

Out[51]: 2011-01-02   -0.402102
         2011-01-05    0.526677
         2011-01-07   -0.968652
         2011-01-08   -0.825882
         dtype: float64
```

```
In [79]: date = '2004-02-29'

In [80]: date = pd.to_datetime(date)
         date

Out[80]: Timestamp('2004-02-29 00:00:00')

In [81]: week = date.week
         weekday = date.weekday()
         year = date.year
         quarter = date.quarter
         month = date.month

In [82]: weekday

Out[82]: 6

In [83]: dates=pd.date_range('1/1/2000',periods=100, freq='W-WED')

In [84]: long_df=DataFrame(np.random.randn(100,4),index=dates, columns=['Colorado','Texas','Ne

In [86]: long_df.ix['5-2001']

/Users/yuyangli/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:1: DeprecationWarni
.ix is deprecated. Please use
.loc for label based indexing or
.iloc for positional indexing

See the documentation here:
http://pandas.pydata.org/pandas-docs/stable/indexing.html#ix-indexer-is-deprecated
  """Entry point for launching an IPython kernel.


Out[86]:             Colorado     Texas  New York      Ohio
         2001-05-02  1.227581  0.129310  1.935146 -1.637018
         2001-05-09  0.186076 -1.382144  0.139707  1.777298
         2001-05-16 -0.937695  0.358065 -0.342104 -0.608721
         2001-05-23 -0.316499 -0.730916 -0.433857 -0.895322
         2001-05-30  1.009299  1.029303 -1.593660 -2.077878

In [88]: dates=pd.DatetimeIndex(['1/1/2000','1/2/2000','1/2/2000','1/2/2000','1/3/2000'])

In [89]: dup_ts=Series(np.arange(5),index=dates)

In [90]: grouped=dup_ts.groupby(level=0)

In [91]: grouped.mean()

Out[91]: 2000-01-01    0
         2000-01-02    2
         2000-01-03    4
         dtype: int64
```

```
In [92]: grouped.count()
```

```
Out[92]: 2000-01-01    1
         2000-01-02    3
         2000-01-03    1
         dtype: int64
```

```
In [94]: ts
```

```
Out[94]: 2011-01-02   -0.402102
         2011-01-05    0.526677
         2011-01-07   -0.968652
         2011-01-08   -0.825882
         2011-01-10    1.713213
         2011-01-12   -0.341772
         dtype: float64
```

```
In [95]: ts.resample('D')
```

```
Out[95]: DatetimeIndexResampler [freq=<Day>, axis=0, closed=left, label=left, convention=start
```

```
In [96]: index=pd.date_range('4/1/2012','6/1/2012')
```

```
In [98]: index
```

```
Out[98]: DatetimeIndex(['2012-04-01', '2012-04-02', '2012-04-03', '2012-04-04',
                        '2012-04-05', '2012-04-06', '2012-04-07', '2012-04-08',
                        '2012-04-09', '2012-04-10', '2012-04-11', '2012-04-12',
                        '2012-04-13', '2012-04-14', '2012-04-15', '2012-04-16',
                        '2012-04-17', '2012-04-18', '2012-04-19', '2012-04-20',
                        '2012-04-21', '2012-04-22', '2012-04-23', '2012-04-24',
                        '2012-04-25', '2012-04-26', '2012-04-27', '2012-04-28',
                        '2012-04-29', '2012-04-30', '2012-05-01', '2012-05-02',
                        '2012-05-03', '2012-05-04', '2012-05-05', '2012-05-06',
                        '2012-05-07', '2012-05-08', '2012-05-09', '2012-05-10',
                        '2012-05-11', '2012-05-12', '2012-05-13', '2012-05-14',
                        '2012-05-15', '2012-05-16', '2012-05-17', '2012-05-18',
                        '2012-05-19', '2012-05-20', '2012-05-21', '2012-05-22',
                        '2012-05-23', '2012-05-24', '2012-05-25', '2012-05-26',
                        '2012-05-27', '2012-05-28', '2012-05-29', '2012-05-30',
                        '2012-05-31', '2012-06-01'],
                       dtype='datetime64[ns]', freq='D')
```

```
In [99]: # BM business end of month
```

```
In [100]: pd.date_range('5/2/2012 12:56:31', periods=5, normalize=True)
```

```
Out[100]: DatetimeIndex(['2012-05-02', '2012-05-03', '2012-05-04', '2012-05-05',
                         '2012-05-06'],
                        dtype='datetime64[ns]', freq='D')
```

```
In [102]: from pandas.tseries.offsets import Hour, Minute

In [103]: hour=Hour()

In [104]: hour

Out[104]: <Hour>

In [105]: Hour(1)+Minute(20)

Out[105]: <80 * Minutes>

In [106]: rng=pd.date_range('1/1/2012','9/1/2012',freq='WOM-3FRI')

In [107]: list(rng)

Out[107]: [Timestamp('2012-01-20 00:00:00', freq='WOM-3FRI'),
           Timestamp('2012-02-17 00:00:00', freq='WOM-3FRI'),
           Timestamp('2012-03-16 00:00:00', freq='WOM-3FRI'),
           Timestamp('2012-04-20 00:00:00', freq='WOM-3FRI'),
           Timestamp('2012-05-18 00:00:00', freq='WOM-3FRI'),
           Timestamp('2012-06-15 00:00:00', freq='WOM-3FRI'),
           Timestamp('2012-07-20 00:00:00', freq='WOM-3FRI'),
           Timestamp('2012-08-17 00:00:00', freq='WOM-3FRI')]

In [108]: # WOM-3FRI

In [109]: ts=Series(np.random.randn(4),index=pd.date_range('1/1/2000',periods=4,freq='M'))

In [110]: ts

Out[110]: 2000-01-31   -0.727768
          2000-02-29    0.564953
          2000-03-31   -0.927963
          2000-04-30   -1.195021
          Freq: M, dtype: float64

In [111]: ts.shift(2)

Out[111]: 2000-01-31         NaN
          2000-02-29         NaN
          2000-03-31   -0.727768
          2000-04-30    0.564953
          Freq: M, dtype: float64

In [113]: ts.shift(-2)

Out[113]: 2000-01-31   -0.927963
          2000-02-29   -1.195021
          2000-03-31         NaN
          2000-04-30         NaN
          Freq: M, dtype: float64
```

```
In [114]: ts/ts.shift(1)-1

Out[114]: 2000-01-31         NaN
          2000-02-29   -1.776282
          2000-03-31   -2.642547
          2000-04-30    0.287790
          Freq: M, dtype: float64

In [115]: from pandas.tseries.offsets import Day, MonthEnd

In [116]: now = datetime(2011, 11, 17)

In [117]: offset=MonthEnd()

In [118]: offset.rollforward(now)

Out[118]: Timestamp('2011-11-30 00:00:00')

In [119]: MonthEnd

Out[119]: pandas.tseries.offsets.MonthEnd

In [128]: ts=Series(np.random.randn(20),index=pd.date_range('1/15/2000',periods=20,freq='4d'))

In [129]: ts

Out[129]: 2000-01-15    0.514568
          2000-01-19    0.154488
          2000-01-23    2.032775
          2000-01-27    0.154592
          2000-01-31    0.782221
          2000-02-04   -1.230421
          2000-02-08   -0.474745
          2000-02-12   -0.177912
          2000-02-16    1.201206
          2000-02-20   -0.311956
          2000-02-24   -2.029122
          2000-02-28    0.308957
          2000-03-03    0.111617
          2000-03-07   -0.300777
          2000-03-11   -1.384269
          2000-03-15    0.716037
          2000-03-19    0.263511
          2000-03-23    0.707939
          2000-03-27    1.115489
          2000-03-31    1.212484
          Freq: 4D, dtype: float64

In [130]: offset.rollforward
```

```
Out[130]: <bound method DateOffset.rollforward of <MonthEnd>>

In [136]: ts.groupby(offset.rollforward).mean()

Out[136]: 2000-01-31    0.727729
          2000-02-29   -0.387713
          2000-03-31    0.305254
          dtype: float64

In [137]: # offset.rollforward

In [138]: ts.resample('M',how='mean')

/Users/yuyangli/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:1: FutureWarning:
the new syntax is .resample(...).mean()
  """Entry point for launching an IPython kernel.


Out[138]: 2000-01-31    0.727729
          2000-02-29   -0.387713
          2000-03-31    0.305254
          Freq: M, dtype: float64

In [139]: import pytz

In [140]: pytz.common_timezones[-5:]

Out[140]: ['US/Eastern', 'US/Hawaii', 'US/Mountain', 'US/Pacific', 'UTC']

In [142]: tz=pytz.timezone('US/Eastern')

In [143]: tz

Out[143]: <DstTzInfo 'US/Eastern' LMT-1 day, 19:04:00 STD>

In [144]: rng=pd.date_range('3/9/2012 9:30',periods=6,freq='D')

In [145]: ts=Series(np.random.randn(len(rng)),index=rng)

In [146]: print(ts.index.tz)

None


In [147]: pd.date_range('3/9/2012 9:30',periods=10,freq='D',tz='UTC')

Out[147]: DatetimeIndex(['2012-03-09 09:30:00+00:00', '2012-03-10 09:30:00+00:00',
                         '2012-03-11 09:30:00+00:00', '2012-03-12 09:30:00+00:00',
                         '2012-03-13 09:30:00+00:00', '2012-03-14 09:30:00+00:00',
                         '2012-03-15 09:30:00+00:00', '2012-03-16 09:30:00+00:00',
                         '2012-03-17 09:30:00+00:00', '2012-03-18 09:30:00+00:00'],
                        dtype='datetime64[ns, UTC]', freq='D')
```

```
In [148]: ts_utc=ts.tz_localize('UTC')
```

```
In [149]: ts_utc
```

```
Out[149]: 2012-03-09 09:30:00+00:00    0.628595
          2012-03-10 09:30:00+00:00   -0.620246
          2012-03-11 09:30:00+00:00   -0.701098
          2012-03-12 09:30:00+00:00   -0.868790
          2012-03-13 09:30:00+00:00   -0.507773
          2012-03-14 09:30:00+00:00    1.815915
          Freq: D, dtype: float64
```

```
In [150]: ts_utc.index
```

```
Out[150]: DatetimeIndex(['2012-03-09 09:30:00+00:00', '2012-03-10 09:30:00+00:00',
                         '2012-03-11 09:30:00+00:00', '2012-03-12 09:30:00+00:00',
                         '2012-03-13 09:30:00+00:00', '2012-03-14 09:30:00+00:00'],
                        dtype='datetime64[ns, UTC]', freq='D')
```

```
In [151]: ts_utc.tz_convert('US/Eastern')
```

```
Out[151]: 2012-03-09 04:30:00-05:00    0.628595
          2012-03-10 04:30:00-05:00   -0.620246
          2012-03-11 05:30:00-04:00   -0.701098
          2012-03-12 05:30:00-04:00   -0.868790
          2012-03-13 05:30:00-04:00   -0.507773
          2012-03-14 05:30:00-04:00    1.815915
          Freq: D, dtype: float64
```

```
In [152]: ts_eastern=ts.tz_localize('US/Eastern')
```

```
In [153]: ts_eastern.tz_convert('UTC')
```

```
Out[153]: 2012-03-09 14:30:00+00:00    0.628595
          2012-03-10 14:30:00+00:00   -0.620246
          2012-03-11 13:30:00+00:00   -0.701098
          2012-03-12 13:30:00+00:00   -0.868790
          2012-03-13 13:30:00+00:00   -0.507773
          2012-03-14 13:30:00+00:00    1.815915
          Freq: D, dtype: float64
```

```
In [154]: ts_eastern.tz_convert('Europe/Berlin')
```

```
Out[154]: 2012-03-09 15:30:00+01:00    0.628595
          2012-03-10 15:30:00+01:00   -0.620246
          2012-03-11 14:30:00+01:00   -0.701098
          2012-03-12 14:30:00+01:00   -0.868790
          2012-03-13 14:30:00+01:00   -0.507773
          2012-03-14 14:30:00+01:00    1.815915
          Freq: D, dtype: float64
```

```
In [155]: ts.index.tz_localize('Asia/Shanghai')

Out[155]: DatetimeIndex(['2012-03-09 09:30:00+08:00', '2012-03-10 09:30:00+08:00',
                         '2012-03-11 09:30:00+08:00', '2012-03-12 09:30:00+08:00',
                         '2012-03-13 09:30:00+08:00', '2012-03-14 09:30:00+08:00'],
                        dtype='datetime64[ns, Asia/Shanghai]', freq='D')

In [157]: stamp=pd.Timestamp('2011-03-12 04:00')

In [158]: stamp_utc=stamp.tz_localize('utc')

In [159]: stamp_utc.tz_convert('US/Eastern')

Out[159]: Timestamp('2011-03-11 23:00:00-0500', tz='US/Eastern')

In [160]: stamp_moscow=pd.Timestamp('2011-03-12 04:00', tz='Europe/Moscow')

In [161]: stamp_moscow

Out[161]: Timestamp('2011-03-12 04:00:00+0300', tz='Europe/Moscow')

In [162]: rng=pd.date_range('3/7/2012 9:30', periods=10, freq='B')

In [163]: ts=Series(np.random.randn(len(rng)),index=rng)

In [164]: ts

Out[164]: 2012-03-07 09:30:00    0.241602
          2012-03-08 09:30:00    0.272515
          2012-03-09 09:30:00    0.396179
          2012-03-12 09:30:00    0.465669
          2012-03-13 09:30:00    0.077151
          2012-03-14 09:30:00   -0.679810
          2012-03-15 09:30:00   -1.554990
          2012-03-16 09:30:00   -0.392588
          2012-03-19 09:30:00   -0.508937
          2012-03-20 09:30:00    0.308505
          Freq: B, dtype: float64

In [165]: ts1=ts[:7].tz_localize('Europe/London')

In [166]: ts2=ts1[2:].tz_convert('Europe/Moscow')

In [167]: result=ts1+ts2

In [169]: result.index

Out[169]: DatetimeIndex(['2012-03-07 09:30:00+00:00', '2012-03-08 09:30:00+00:00',
                         '2012-03-09 09:30:00+00:00', '2012-03-12 09:30:00+00:00',
                         '2012-03-13 09:30:00+00:00', '2012-03-14 09:30:00+00:00',
                         '2012-03-15 09:30:00+00:00'],
                        dtype='datetime64[ns, UTC]', freq='B')
```

```
In [170]: p=pd.Period(2007, freq='A-DEC')

In [171]: p

Out[171]: Period('2007', 'A-DEC')

In [172]: p+5

Out[172]: Period('2012', 'A-DEC')

In [173]: p-2

Out[173]: Period('2005', 'A-DEC')

In [174]: values=['2001Q3','2002Q2','2003Q1']

In [175]: index=pd.PeriodIndex(values, freq='Q-DEC')

In [176]: index

Out[176]: PeriodIndex(['2001Q3', '2002Q2', '2003Q1'], dtype='period[Q-DEC]', freq='Q-DEC')

In [177]: p=pd.Period('2007',freq='A-DEC')

In [178]: p.asfreq('M',how='start')

Out[178]: Period('2007-01', 'M')

In [179]: p=pd.Period('2012Q4',freq='Q-JAN')

In [180]: p

Out[180]: Period('2012Q4', 'Q-JAN')

In [181]: p4pm=(p.asfreq('B','e')-1).asfreq('T','s')+16*60

In [182]: p4pm

Out[182]: Period('2012-01-30 16:00', 'T')

In [183]: p4pm.to_timestamp()

Out[183]: Timestamp('2012-01-30 16:00:00')

In [194]: rng=pd.date_range('1/1/2000',periods=12,freq='T')

In [195]: ts=Series(np.arange(12),index=rng)

In [196]: ts.resample('5min',how='ohlc')

/Users/yuyangli/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:1: FutureWarning:
the new syntax is .resample(...).ohlc()
  """Entry point for launching an IPython kernel.
```

```
Out[196]:                      open  high  low  close
          2000-01-01 00:00:00     0     4    0      4
          2000-01-01 00:05:00     5     9    5      9
          2000-01-01 00:10:00    10    11   10     11
```

```
In [207]: rng1=pd.date_range('1/1/2000',periods=100,freq='D')
```

```
In [209]: ts=Series(np.arange(100),index=rng1)
```

```
In [210]: ts.groupby(lambda x: x.month).mean()
```

```
Out[210]: 1    15
          2    45
          3    75
          4    95
          dtype: int64
```

```
In [211]: ts.groupby(lambda x: x.weekday).mean()
```

```
Out[211]: 0    47.5
          1    48.5
          2    49.5
          3    50.5
          4    51.5
          5    49.0
          6    50.0
          dtype: float64
```

```
In [212]: frame=DataFrame(np.random.randn(2,4),index=pd.date_range('1/1/2000',periods=2, freq=
```

```
In [213]: frame[:5]
```

```
Out[213]:             Colorado     Texas  New York      Ohio
          2000-01-05  0.679431  1.892319  0.807650  1.614580
          2000-01-12 -2.002261  0.092897  0.604535 -0.520433
```

```
In [216]: df_daily=frame.resample('D')
```

```
In [218]: df_daily
```

```
Out[218]: DatetimeIndexResampler [freq=<Day>, axis=0, closed=left, label=left, convention=star
```

```
In [219]: frame.resample('D',fill_method='ffill')
```

```
/Users/yuyangli/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:1: FutureWarning:
the new syntax is .resample(...).ffill()
  """Entry point for launching an IPython kernel.
```

```
Out[219]:             Colorado     Texas  New York      Ohio
          2000-01-05  0.679431  1.892319  0.807650  1.614580
          2000-01-06  0.679431  1.892319  0.807650  1.614580
          2000-01-07  0.679431  1.892319  0.807650  1.614580
          2000-01-08  0.679431  1.892319  0.807650  1.614580
          2000-01-09  0.679431  1.892319  0.807650  1.614580
          2000-01-10  0.679431  1.892319  0.807650  1.614580
          2000-01-11  0.679431  1.892319  0.807650  1.614580
          2000-01-12 -2.002261  0.092897  0.604535 -0.520433
```

```
In [220]: frame.resample('D', fill_method='ffill',limit=2)
```

```
/Users/yuyangli/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:1: FutureWarning:
the new syntax is .resample(...).ffill(limit=2)
  """Entry point for launching an IPython kernel.
```

```
Out[220]:             Colorado     Texas  New York      Ohio
          2000-01-05  0.679431  1.892319  0.807650  1.614580
          2000-01-06  0.679431  1.892319  0.807650  1.614580
          2000-01-07  0.679431  1.892319  0.807650  1.614580
          2000-01-08       NaN       NaN       NaN       NaN
          2000-01-09       NaN       NaN       NaN       NaN
          2000-01-10       NaN       NaN       NaN       NaN
          2000-01-11       NaN       NaN       NaN       NaN
          2000-01-12 -2.002261  0.092897  0.604535 -0.520433
```

```
In [221]: #
          # spx_px = close_px_all['SPX']
          # spx_rets=spx_px/spx_px.shift(1)-1
          # returns=close_px.pct_change()
          # corr=pd.rolling_corr(returns.AAPL, spx_rets, 125, min_periods=100)
          # corr.plot()
```

```
In [222]: # corr=pd.rolling_corr(returns, spx_rets, 125, min_periods=100)
          # corr.plot()
```

```
In [223]: # from scipy.stats import percentileofscore
          # score_at_2percent = lambda x: percentileofscore(x, 0.02)
          # result = pd.rolling_apply(returns.AAPL, 250, score_at_2percent)
          # result.plot()
```

```
In [ ]:
```