18. Time Series và kĩ thuật resample

Phần bài này sẽ hướng dẫn các bạn cách Pandas giải quyết dữ liệu với chuỗi thời gian (time series) như thế nào. Time series là một loạt các dữ liệu, được liệt kê (hoặc được lập chỉ mục) theo thứ tự thời gian. Thông thường, chuỗi thời gian là một dãy các giá trị, có khoảng cách đều nhau theo thời gian. Tất cả mọi thứ bao gồm dữ liệu đo được gắn với thời gian tương ứng có thể được xem như một chuỗi thời gian. Các phép đo có thể được thực hiện bất thường, nhưng trong nhiều trường hợp, các chuỗi thời gian được lấy mẫu với tần suất cố định. Điều này có nghĩa là dữ liệu được đo hoặc được lấy theo một mẫu thông thường, ví dụ như 5 mili giây, mỗi 10 giây hoặc hàng giờ. Hàng loạt các chuỗi thời gian thường được vẽ dưới dạng biểu đồ đường.

Trong chương này chúng tôi sẽ giới thiệu các công cụ từ Pandas để xử lý time series. Bạn sẽ học cách xử lý với các chuỗi thời gian lớn và cách thay đổi chuỗi thời gian.

Trước khi bạn tiếp tục đọc, có thể hữu ích khi xem hướng dẫn khác về các mô đun Python chuẩn liên quan đến xử lý thời gian như datetime, time and calendar:

Indexing pandas time series.

Series: nó được xây dựng dựa trên index của nó là các time stamps.

```
>>> import numpy as np
>>> import pandas as pd
>>> from datetime import datetime, timedelta as delta
>>> ndays = 10
>>> start = datetime(2017, 3, 31)
>>> dates = [start - delta(days=x) for x in range(0, ndays)]
>>> values = [25, 50, 15, 67, 70, 9, 28, 30, 32, 12]
>>> ts = pd.Series(values, index=dates)
>>> ts
2017-03-31    25
2017-03-30    50
2017-03-29    15
2017-03-28    67
2017-03-27    70
2017-03-26     9
2017-03-25    28
2017-03-24    30
2017-03-23    32
2017-03-22    12
dtype: int64
>>> print type(ts)
<class 'pandas.core.series.Series'>
>>>
```

Dataframe: Nhắc lại bài số 8 “CSV reindexing”, chúng ta đã đề cập đến tham số parse_dates=True trong pd.read_csv() có thể đọc strings và chuyển đổi về dạng datetime. Ta cùng nhắc lại ví dụ đó sử dụng “sales_14.csv”.

```
>>> sales = pd.read_csv("sales_14.csv",header=0, index_col='Date', parse_dates=True)
>>> sales.head()
                     Company   Product  Units
Date
2015-02-02 08:30:00           Hooli  Software      3
2015-02-02 21:00:00       Mediacore  Hardware      9
2015-02-03 14:00:00         Initech  Software     13
2015-02-04 15:30:00       Streeplex  Software     13
2015-02-04 22:00:00  Acme Coporation  Hardware     14
>>>
```

Selection: khi indexes là kiểu datetime, chúng ta có thể tiến hành nhiều kiểu selection và slicing phức tạp qua hỗ trợ của .loc[]. Ví dụ như lựa ra toàn bộ hàng thuộc tháng 2 ngày 11 trong sales. Và nó cũng hỗ trợ chỉ chọn năm sales.loc['2015',:] hoặc tháng sales.loc['2015-02',:].

```
>>> sales.loc['2015-02-11',:]
                  Company   Product  Units
Date
2015-02-11 20:00:00  Initech  Software      7
2015-02-11 23:00:00    Hooli  Software      4
```

```
>>> sales.loc['2015-Feb-11',:]
                 Company   Product  Units
Date
2015-02-11 20:00:00  Initech  Software      7
2015-02-11 23:00:00    Hooli  Software      4
>>> sales.loc['February 11, 2015',:]
                 Company   Product  Units
Date
2015-02-11 20:00:00  Initech  Software      7
2015-02-11 23:00:00    Hooli  Software      4
>>> sales.loc['2015-02',:].head(2)
```

Converting to Timestamps

Chuyển đổi một đối tượng giống chẳng hạn một danh sách các string định dạng ngày tháng ví dụ: string, time object, hoặc hỗn hợp, bạn có thể sử dụng hàm

```
>>> pd.to_datetime(pd.Series(['Jul 31, 2009', '2010-01-10', None]))
0   2009-07-31
1   2010-01-10
2          NaT
dtype: datetime64[ns]
>>> pd.to_datetime(['2005/11/23', '2010.12.31'])
DatetimeIndex(['2005-11-23', '2010-12-31'], dtype='datetime64[ns]', freq=None)
>>>
```

Chúng ta có thể tùy biến để có được style mong muốn cho DatetimeIndex. Ví dụ dùng chuẩn châu âu.

```
>>> pd.to_datetime(['04-01-2012 10:00'], dayfirst=True)
DatetimeIndex(['2012-01-04 10:00:00'], dtype='datetime64[ns]', freq=None)
>>>
```

Ta cũng có thể truyền vào một data time frame có định dạng như sau để chuẩn hóa dữ liệu.

```
>>> pd.DataFrame({'year': [2015, 2016],'month': [2, 3],'day': [4, 5],'hour': [2, 3]})
>>> df
   day  hour  month  year
0    4     2      2  2015
1    5     3      3  2016
>>>
```

Cần chuẩn hóa frame ‘df’’ theo kiểu châu âu bằng cách dùng to_datetime() như sau:

```
>>> pd.to_datetime(df,dayfirst=True)
0   2015-02-04 02:00:00
1   2016-03-05 03:00:00
dtype: datetime64[ns]
>>>
```

Khi làm việc với time series, chuẩn hóa time là một bước quan trọng để ta có thể dễ dàng group, hay thực hiện transformation ở các bước tiếp theo.

Để hiểu sâu hơn về time series, các bạn có thể tham khảo tại.

https://pandas.pydata.org/pandas-docs/stable/timeseries.html

Resampling

Sử dụng một phương pháp gọi là resampling, chúng ta có thể thực hiện một số xử lý thống kê trên một khoảng thời gian như .mean(), .sum(), .count() etc. Pandas cung cấp phương thức .resample() có rất nhiều tùy chọn để ta có thể tùy chỉnh. Ta cần chú ý đến 2 contexts sau đây:

+ Downsampling: Giảm tần suất lấy mẫu bằng cách tăng thời gian lấy mẫu từ vài phút đến vài giờ.

+ Upsampling: Tăng tần suất lấy mẫu bằng cách giảm thời gian lấy mẫu từ vài giờ xuống vài phút.

Trong trường hợp Downsampling, mối quan tâm có thể là xác định giá trị quan sát được tính bằng cách sử dụng phép nội suy. Trong trường hợp Upsampling, cần phải chú ý đến việc là giá trị thống kê .sum(), .mean() etc được sử dụng để tính toán các giá trị tổng hợp mới.

Để .resample() hoạt động ta cần chỉ rõ tần suất sample. Và bảng sau cho ta một vài string phổ biến để sử dụng làm đơn vị. Và ta có thể sử dụng như ‘3W’, ‘2A’ etc.

Input	Ý nghĩa
‘min’, ‘T’	Minute
‘H’	Hour
‘D’	Day
‘B’	Business day
‘W’	Week
‘M’	Month
‘Q’	Quarter
‘A’	Year

Upsampling (ffill(), bfill(), head(), first(), interpolate(‘linear’))

```
>>> sales.resample('W')['Units'].ffill()
Date
2015-02-02 08:30:00    3
2015-02-02 21:00:00    9
2015-02-03 14:00:00   13
2015-02-04 15:30:00   13
2015-02-04 22:00:00   14
2015-02-05 02:00:00   19
2015-02-05 22:00:00   10
2015-02-07 23:00:00    1
2015-02-09 09:00:00   19
2015-02-09 13:00:00    7
2015-02-11 20:00:00    7
2015-02-11 23:00:00    4
2015-02-16 12:00:00   10
2015-02-19 11:00:00   16
2015-02-19 16:00:00   10
2015-02-21 05:00:00    3
2015-02-21 20:30:00    3
2015-02-25 00:30:00   10
2015-02-26 09:00:00    4
Name: Units, dtype: int64
>>>
```

Downsampling (sum(), mean(), std() etc)

```
>>> sales.resample('W')['Units'].std()
Date
2015-02-08    5.922114
2015-02-15    6.652067
2015-02-22    5.504544
2015-03-01    4.242641
Freq: W-SUN, Name: Units, dtype: float64
>>>
```

Datetime methods

Để nhận được đối tượng datetime từ time series ta có thể sử dụng .dt như ví dụ sau: chú ý sử dụng lại sales sau khi đã reset_index().

```
>>> sales.reset_index(inplace=True)
>>> sales.Date.dt
```

```
<pandas.core.indexes.accessors.DatetimeProperties object at 0x0000015D1040BC18>
>>>
```

Chúng ta có thể chỉ nhận giờ/phút/năm etc từ cột ‘Date’.

```
>>> sales.Date.dt.hour
0     8
1    21
2    14
3    15
4    22
5     2
6    22
7    23
8     9
9    13
10   20
11   23
12   12
13   11
14   16
15    5
16   20
17    0
18    9
Name: Date, dtype: int64
>>>
```

```
>>> sales.Date.dt.minute
0    30
1     0
2     0
3    30
4     0
5     0
6     0
7     0
8     0
9     0
10    0
11    0
12    0
13    0
14    0
15    0
16   30
17   30
18    0
Name: Date, dtype: int64
>>>
```

```
>>> sales.Date.dt.year
0    2015
1    2015
2    2015
3    2015
4    2015
5    2015
6    2015
7    2015
8    2015
9    2015
10   2015
11   2015
12   2015
13   2015
14   2015
15   2015
16   2015
17   2015
18   2015
Name: Date, dtype: int64
>>>
```

Chúng ta cũng có thể thiết lập hoặc chuyển đổi timezone qua phương thức .tz_localize() và tz_convert().

```
>>> central = sales.Date.dt.tz_localize("US/Central")
>>> central
0   2015-02-02 08:30:00-06:00
1   2015-02-02 21:00:00-06:00
2   2015-02-03 14:00:00-06:00
3   2015-02-04 15:30:00-06:00
4   2015-02-04 22:00:00-06:00
5   2015-02-05 02:00:00-06:00
6   2015-02-05 22:00:00-06:00
7   2015-02-07 23:00:00-06:00
8   2015-02-09 09:00:00-06:00
9   2015-02-09 13:00:00-06:00
10  2015-02-11 20:00:00-06:00
11  2015-02-11 23:00:00-06:00
12  2015-02-16 12:00:00-06:00
13  2015-02-19 11:00:00-06:00
14  2015-02-19 16:00:00-06:00
15  2015-02-21 05:00:00-06:00
16  2015-02-21 20:30:00-06:00
17  2015-02-25 00:30:00-06:00
18  2015-02-26 09:00:00-06:00
Name: Date, dtype: datetime64[ns, US/Central]
>>>
```

```
>>> central.dt.tz_convert("US/Eastern")
```

```
0   2015-02-02 09:30:00-05:00
1   2015-02-02 22:00:00-05:00
2   2015-02-03 15:00:00-05:00
3   2015-02-04 16:30:00-05:00
4   2015-02-04 23:00:00-05:00
5   2015-02-05 03:00:00-05:00
6   2015-02-05 23:00:00-05:00
7   2015-02-08 00:00:00-05:00
8   2015-02-09 10:00:00-05:00
9   2015-02-09 14:00:00-05:00
10  2015-02-11 21:00:00-05:00
11  2015-02-12 00:00:00-05:00
12  2015-02-16 13:00:00-05:00
13  2015-02-19 12:00:00-05:00
14  2015-02-19 17:00:00-05:00
15  2015-02-21 06:00:00-05:00
16  2015-02-21 21:30:00-05:00
17  2015-02-25 01:30:00-05:00
18  2015-02-26 10:00:00-05:00
Name: Date, dtype: datetime64[ns, US/Eastern]
>>>
```

Kết Luận

Qua bài học này ta cần nắm được một số nội dung chính sau: (1) Cách thức chuyển đổi (convert) qua lại giữa các kiểu dữ liệu string và datetime. (2) Cách sử dụng cột date là index và các kĩ thuật selection/slice tương ứng. (3) Kĩ thuật resample dữ liệu để có thể khai phá dữ liệu được phân định theo giờ, ngày, tháng, năm …