Chuyển đổi kiểu dữ liệu

23-12-2020

1 Giới thiệu

Trong bài học này, chúng ta học về cách chuyển đổi kiểu dữ liệu trong DataFrame.

2 Chuyển đổi chuỗi

```
In [1]: import pandas as pd

        # đọc file Data\vnm.txt
        # dùng tham số names để đổi đặt tên cho các cột đọc vào
        df = pd.read_csv(
            'https://raw.githubusercontent.com/Levytan/MIS.2019/master/Data/vnm.csv',
            header = None,
            names = ['Date', 'Close', 'Volume', 'PercentChange']
        )

        df
```

Out[1]:

	Date	Close	Volume	PercentChange
0	2018-06-01	136.1k	1051610	?%
1	2018-06-04	141.6k	965230	4.04%
2	2018-06-05	144.4k	879160	1.98%
3	2018-06-06	142.8k	463720	-1.11%
4	2018-06-07	144.7k	401900	1.33%
5	2018-06-08	144.4k	505520	-0.21%
6	2018-06-11	147.0k	721050	1.8%
7	2018-06-12	147.7k	873690	0.48%
8	2018-06-13	147.7k	472470	0.0%
9	2018-06-14	146.9k	767750	-0.54%
10	2018-06-15	146.9k	1114730	0.0%

Bài tập : In ra kiểu dữ liệu của từng cột của `df`.

```
In [2]: df.dtypes
```

```
Out[2]: Date             object
        Close            object
        Volume            int64
        PercentChange    object
        dtype: object
```

Như các bạn có thể thấy, cột `Date`, cột `Close` và cột `PercentChange` đều là `str`. Điều này sẽ gây khó khăn cho việc xử lý dữ liệu. Vì vậy, chúng ta cần phải chuyển đổi kiểu dữ liệu về dạng phù hợp.

2.1 Chuyển đổi chuỗi ngày tháng thàng kiểu dữ liệu thời gian

Để chuyển đổi từ chuỗi ngày tháng thành kiểu thời gian, chúng ta sử dụng:
`pandas.to_datetime()`.

```
In [3]: pd.to_datetime(df.Date)
```

```
Out[3]: 0    2018-06-01
        1    2018-06-04
        2    2018-06-05
        3    2018-06-06
        4    2018-06-07
        5    2018-06-08
        6    2018-06-11
        7    2018-06-12
        8    2018-06-13
        9    2018-06-14
        10   2018-06-15
        Name: Date, dtype: datetime64[ns]
```

```
In [4]: # chuyển đổi cột Date sang kiểu thời gian
        df = df.assign(Date = pd.to_datetime(df.Date))

        # kiểm tra lại
        df.dtypes
```

```
Out[4]: Date              datetime64[ns]
        Close                     object
        Volume                     int64
        PercentChange             object
        dtype: object
```

Như vậy, bạn đã chuyển đổi thành công thành kiểu thời gian, bây giờ có thể làm nhiều thứ hay ho hơn rồi. Chẳng hạn như đặt cột `Date` làm `index`, hoặc

```
In [5]: # dùng accessor .dt để dùng các phương thức của kiểu dữ liệu thời gian
        df.assign(Fancy = df.Date.dt.strftime('%d/%m/%Y'))
```

```
Out[5]:         Date   Close   Volume PercentChange       Fancy
        0  2018-06-01  136.1k  1051610            ?%  01/06/2018
        1  2018-06-04  141.6k   965230         4.04%  04/06/2018
        2  2018-06-05  144.4k   879160         1.98%  05/06/2018
        3  2018-06-06  142.8k   463720        -1.11%  06/06/2018
        4  2018-06-07  144.7k   401900         1.33%  07/06/2018
        5  2018-06-08  144.4k   505520        -0.21%  08/06/2018
        6  2018-06-11  147.0k   721050          1.8%  11/06/2018
        7  2018-06-12  147.7k   873690         0.48%  12/06/2018
        8  2018-06-13  147.7k   472470          0.0%  13/06/2018
        9  2018-06-14  146.9k   767750        -0.54%  14/06/2018
        10 2018-06-15  146.9k  1114730          0.0%  15/06/2018
```

Chú ý :
`accessor` là một tính năng của `pandas` cho phép dùng các phương thức tương ứng với kiểu dữ liệu trong một `Series`.
Các `accessor` thường dùng:

- `.str` dành cho kiểu chuỗi.
- `.dt` dành cho kiểu thời gian.
- `.cat` dành cho kiểu phân loại.

2.2 Chuyển đổi chuỗi số thành kiểu số

Để chuyển đổi chuỗi số, chúng ta dùng đến `pandas.to_numeric()`.

```
In [7]: # ví dụ, chuyển đổi cột Close thành kiểu số
        # trước tiên, bỏ ký tự k
        df = df.assign(Close = df.Close.str.replace('k', ''))

        # kế tiếp, chuyển đổi thành kiểu số
        df = df.assign(Close = pd.to_numeric(df.Close))

        df
```

Out[7]:

	Date	Close	Volume	PercentChange
0	2018-06-01	136.1	1051610	?%
1	2018-06-04	141.6	965230	4.04%
2	2018-06-05	144.4	879160	1.98%
3	2018-06-06	142.8	463720	-1.11%
4	2018-06-07	144.7	401900	1.33%
5	2018-06-08	144.4	505520	-0.21%
6	2018-06-11	147.0	721050	1.8%
7	2018-06-12	147.7	873690	0.48%
8	2018-06-13	147.7	472470	0.0%
9	2018-06-14	146.9	767750	-0.54%
10	2018-06-15	146.9	1114730	0.0%

```
In [8]: df = df.assign(Close = df.Close * 1000)
        df
```

Out[8]:

	Date	Close	Volume	PercentChange
0	2018-06-01	136100.0	1051610	?%
1	2018-06-04	141600.0	965230	4.04%
2	2018-06-05	144400.0	879160	1.98%
3	2018-06-06	142800.0	463720	-1.11%
4	2018-06-07	144700.0	401900	1.33%
5	2018-06-08	144400.0	505520	-0.21%
6	2018-06-11	147000.0	721050	1.8%
7	2018-06-12	147700.0	873690	0.48%
8	2018-06-13	147700.0	472470	0.0%
9	2018-06-14	146900.0	767750	-0.54%
10	2018-06-15	146900.0	1114730	0.0%

Bài tập : Chuyển đổi cột `PercentChange` thành chuỗi số.

```
In [9]: pd.to_numeric(
            df.PercentChange.str.replace('%', ''),
            errors = 'coerce'
        )

Out[9]: 0      NaN
        1     4.04
        2     1.98
        3    -1.11
        4     1.33
        5    -0.21
        6     1.80
        7     0.48
        8     0.00
        9    -0.54
        10    0.00
        Name: PercentChange, dtype: float64
```

Sau khi chuyển đổi kiểu, nhớ kiểm tra xem có tồn tại giá trị `NaN` nào hay không và sử dụng các biện pháp phù hợp với những giá trị đó.