Làm việc với `DataFrame`

23-12-2020

1 Dữ liệu mẫu

In [1]:
```
# DataFrame mẫu
import numpy as np
import pandas as pd

date = np.datetime64('2019-11-25', 'D') +
    np.random.randint(-100, 100, size = 100)
product = np.random.choice(['Apple', 'Banana', 'Cherry'], size = 100)
quantity = np.random.randint(100, size = 100)

df1 = pd.DataFrame({
    'Date' : date,
    'Product' : product,
    'Quantity' : quantity
})
df2 = pd.DataFrame(
    ['Apple', 'Banana', 'Cherry'],
    columns = ['Product']
)
```

In [2]:
```
df1.head(10)
```

Out[2]:

	Date	Product	Quantity
0	2019-09-05	Apple	77
1	2020-01-08	Cherry	82
2	2019-12-12	Banana	2
3	2020-01-28	Banana	67
4	2020-01-30	Cherry	24
5	2019-11-03	Apple	27
6	2020-02-28	Banana	39
7	2019-11-25	Cherry	90
8	2020-02-08	Apple	1
9	2019-11-28	Apple	95

In [3]:
```
df2
```

Out[3]:

	Product
0	Apple
1	Banana
2	Cherry

2 Thêm cột mới vào DataFrame

Để thêm một cột mới vào DataFrame, ta có thể làm như sau :

```
<tên_DataFrame>[<tên_cột_mới>] = <giá_trị>
```

hoặc dùng phương thức .assign()

```
.assign(<tên_cột_mới> = <giá_trị>)
```

Trong đó, giá_trị có thể là một list, pandas.Series

```
In [4]: # Ví dụ :
        df2['Price'] = [10, 15, 20] # df2 = df2.assign(Price = [10, 15, 20])
        df2

Out[4]:   Product  Price
        0   Apple     10
        1  Banana     15
        2  Cherry     20
```

Câu hỏi : - Nếu số lượng của giá trị khác với số lượng của dòng của DataFrame, phép gán trên có thể thực hiện được hay không? - Trong trường hợp giá_trị là một Series có label không nằm trong DataFrame, việc thêm cột mới có thể được thực hiện hay không?

3 Thêm dòng mới vào DataFrame

Để thêm dòng mới vào DataFrame, ta có thể dùng phương thức .append() như sau :

```
.append(<giá_trị>, ignore_index)
```

Trong đó : - giá_trị có thể là một dictionary hoặc DataFrame. - ignore_index dùng để báo cho pandas cách đánh label của dòng mới. Có hai giá trị là True và False. True sẽ bỏ qua label (nếu có) của các dòng mới, False sẽ sử dụng label (nếu có) của các dòng mới. Mặc định là False.

```
In [5]: # Ví dụ
        df2.append({'Product' : 'Durian', 'Price' : 25}, ignore_index = True)

Out[5]:   Product  Price
        0   Apple     10
        1  Banana     15
        2  Cherry     20
        3  Durian     25

In [6]: df3 = pd.DataFrame([['Durian', 25], ['Fig', 30]], columns = ['Product', 'Pric
        df2.append(df3)

Out[6]:   Product  Price
        0   Apple     10
        1  Banana     15
        2  Cherry     20
        0  Durian     25
        1     Fig     30
```

4 Xoá dòng, cột trong DataFrame

Để xoá dữ liệu trong DataFrame, ta có thể dùng phương thức .drop() như sau :

```
.drop(index = <dòng_cần_xoá>, columns = <cột_cần_xoá>)
```

```
In [7]: df2.drop(index = [1])
```

```
Out[7]:   Product  Price
        0   Apple     10
        2  Cherry     20
```

```
In [8]: df2.drop(columns = ['Product'])
```

```
Out[8]:    Price
        0     10
        1     15
        2     20
```

5 Thay đổi giá trị của DataFrame

5.1 Thay đổi giá trị cột

Bạn có thể thực hiện giống như việc thêm cột mới, nhưng thay vì dùng tên_cột_mới, bạn dùng tên_cột_đã_có.

```
In [9]: # ví dụ
        df2.assign(Price = df2.Price * 2)
```

```
Out[9]:   Product  Price
        0   Apple     20
        1  Banana     30
        2  Cherry     40
```

5.2 Thay đổi giá trị dòng

Để thay đổi giá trị một dòng, thực hiện như sau:

```
<dòng_cần_thay_đổi> = <giá_trị_mới>
```

Lưu ý : kích thước của giá_trị_mới phải bằng với kích thước của dòng_cần_thay_đổi

```
In [10]: # ví dụ
         # tạo bản copy
         df4 = df2.copy()
         # thay đổi giá trị 2 dòng đầu tiên
         df4.iloc[0:2] = [['Mango', 12], ['Longan', 14]]
         df4
```

```
Out[10]:   Product  Price
         0   Mango     12
         1  Longan     14
         2  Cherry     20
```

6 Sắp xếp `DataFrame` theo giá trị cột

Để sắp xếp một `DataFrame` theo giá trị của cột, ta có thể dùng phương thức `.sort_values()` như sau:

```
.sort_values(by = <tên_cột>, ascending)
```

Trong đó `ascending` là tham số để chỉ cách sắp xếp. Có 2 giá trị: * `True` để sắp xếp tăng dần. * `False` để sắp xếp giảm dần.

Mặc định là `True`.

```
In [11]: # ví dụ :
         df1.sort_values(by = 'Date', ascending = False)
```

```
Out[11]:          Date Product  Quantity
         57 2020-03-03  Banana        82
         35 2020-03-02  Banana        22
         28 2020-03-02   Apple         1
         6  2020-02-28  Banana        39
         52 2020-02-27  Cherry        30
         ..         …     …        …
         64 2019-08-28   Apple        87
         81 2019-08-27   Apple        57
         58 2019-08-27  Banana        60
         32 2019-08-21  Cherry        44
         17 2019-08-17  Cherry        86

         [100 rows x 3 columns]
```