

8. .loc, .iloc, .ix

.iloc, .loc .ix

Có ba lựa chọn chính có thể selecting một dữ liệu của các hàng và cột trong Pandas, điều này có thể gây nhầm lẫn. Ba trường hợp selecting và phương pháp được bao gồm trong bài đăng này là:

1. **.iloc** giúp selecting hàng và cột qua các row và column numbers. Cú pháp `data.iloc[<row selection>, <column selection>]`. Những input được phép truyền vào là một số nguyên (5), một list của các số nguyên ([1,2,3]), một slice object với các số nguyên (1:5), một boolean array hay một callable function.
2. **.loc** giúp selecting hàng và cột qua hai cách: Cách 1 qua các row và column index hoặc nhãn. Cách 2 qua [boolean/conditional lookup](#), tôi sẽ chỉ rõ qua các ví dụ. Cú pháp là `data.loc[<row selection>, <column selection>]`.
3. **.ix** là lai của hai cách phía trên. Nếu truyền vào là một label không phải số nguyên thì nó sẽ hoạt động giống .loc còn nếu truyền vào kiểu số nguyên nó sẽ hoạt động giống iloc.

Trước tiên ta tạo một dataframe để demo cho bài học này.

```
>>> import pandas as pd

>>> crimes_rates = {"year": [1960, 1961, 1962, 1963, 1964], "Population": [179323175, 182992000, 185771000, 188483000, 191141000], "Total": [3384200, 3488000, 3752200, 4109500, 4564600], "Violent": [288460, 289390, 301510, 316970, 364220]}

>>> crimes_dataframe = pd.DataFrame(crimes_rates)

>>> crimes_dataframe
   Population  Total  Violent year
0  179323175  3384200  288460  1960
1  182992000  3488000  289390  1961
2  185771000  3752200  301510  1962
3  188483000  4109500  316970  1963
4  191141000  4564600  364220  1964

>>>
```

.iloc

```
>>> print "select row having row number is 1"

select row having row number is 1

>>> crimes_dataframe.iloc[1]

Population    182992000
Total         3488000
Violent       289390
year          1961
Name: 1, dtype: int64

>>> crimes_dataframe.loc[1]

Population    182992000
Total         3488000
Violent       289390
year          1961
Name: 1, dtype: int64

>>> print "select data of Violent when index of row is 3"

select data of Violent when index of row is 3

>>> crimes_dataframe.iloc[3,2]

316970
```

Sử dụng .iloc hay .loc cho cùng một kết quả trong trường hợp này vì index được tạo ra mặc định cũng là row numbers.

Ta sẽ chỉnh lại nhãn cho các hàng. Để chứng minh .iloc không làm việc với nhãn.

```
>>> ordinals = ["first", "second", "third", "fourth", "fifth",]

>>> crimes_dataframe = pd.DataFrame(crimes_rates, index=ordinals)

>>> crimes_dataframe.iloc["first"]

Traceback (most recent call last):

...

TypeError: cannot do positional indexing on <class 'pandas.indexes.base.Index'> with these indexers [first] of <type 'str'>
```

.loc

Tiếp tục ví dụ trên thay vì dùng iloc với nhãn 'first' ta dùng .loc.

```
>>> crimes_dataframe.loc["first"]
```

```
Population    179323175
Total          3384200
Violent        288460
year           1960
Name: first, dtype: int64
>>>
```

Tương tự ta cũng không thể dùng row number khi dùng `.loc`.

```
>>> crimes_dataframe.loc[0]

Traceback (most recent call last):

...

TypeError: cannot do label indexing on <class 'pandas.indexes.base.Index'> with these indexers [0] of <type 'int'>

>>>
```

Với `.loc` ta có thể truyền vào [boolean/conditional lookup](#) (Boolean filter vector). Bằng cách đó ta có thể lọc được các hàng thỏa mãn điều kiện nào đó.

```
>>> crimes_dataframe.loc[crimes_dataframe.year > 1962]

   Population    Total  Violent  year
fourth  188483000  4109500   316970  1963
fifth   191141000  4564600   364220  1964

>>>
```

Chúng ta không thể truyền một (Boolean vector) vào `iloc` như ví dụ trên.

```
>>> crimes_dataframe.iloc[crimes_dataframe.year > 1962]

Traceback (most recent call last):

...

ValueError: iLocation based boolean indexing cannot use an indexable as a mask
```

`.ix`

Ta thấy `.ix` làm được kết hợp của `iloc` và `loc` ở phía trên.

```
>>> crimes_dataframe.ix[0]

Population    179323175
Total          3384200
Violent        288460
year           1960
Name: first, dtype: int64

>>> crimes_dataframe.ix['first']

Population    179323175
Total          3384200
Violent        288460
year           1960
Name: first, dtype: int64

>>> crimes_dataframe.ix[crimes_dataframe.year > 1962]

   Population    Total  Violent  year
fourth  188483000  4109500   316970  1963
fifth   191141000  4564600   364220  1964

>>>
```

Chú ý ta có thể cùng truyền một list các row number, index/label vào `.iloc`, `loc` hay `ix` để nhận được nhiều hàng trong một lần selecting. Ví dụ:

```
>>> crimes_dataframe.iloc[[0,4],:]

   Population    Total  Violent  year
first  179323175  3384200   288460  1960
fifth  191141000  4564600   364220  1964

>>> crimes_dataframe.loc[['first','fifth'],:]

   Population    Total  Violent  year
```

```
first 179323175 3384200 288460 1960
fifth 191141000 4564600 364220 1964

>>> crimes_dataframe.ix[['first','fifth'],:]

      Population  Total Violent year
first 179323175 3384200 288460 1960
fifth 191141000 4564600 364220 1964
```

Kĩ thuật slicing với hàng.

```
>>> crimes_dataframe[0:2]

      Population  Total Violent year
first 179323175 3384200 288460 1960
second 182992000 3488000 289390 1961

>>>
```

Kết Luận

.iloc và .loc là các phương thức để truy vấn dữ liệu hàng và cột từ Pandas dataframes. Bảng dưới đây sẽ tổng hợp cách sử dụng 2 phương thức này [1]:

.iloc selections		
data.iloc	[<row selection> , <column selection>]	
	Integer list of rows: [0,2]	Integer list of columns: [0,2]
	Slice of rows: [0:4]	Slice of columns: [0:4]
	Single values: 1	Single column selections: 1
.loc selections		
data.loc	[<row selection> , <column selection>]	
	Index/label value: 'first'	Named column: 'year'
	List of labels: ['first','second']	List of col names:['Total','year']
	Logical/Boolean index: data['year'] > 1962	Slice of columns: 'total':'year'

.ix là kết hợp của hai phương pháp trên.

[1] <https://www.shanelynn.ie/select-pandas-dataframe-rows-and-columns-using-iloc-loc-and-ix/>