

[Python] Pandas 사용법 - 인덱싱 접근,데이터 조작, 인덱스조작(loc,iloc) — 나무늘보의 개발 블로그

노트북: 첫 번째 노트북

만든 날짜: 2020-11-05 오전 1:32

URL: <https://continuous-development.tistory.com/133?category=736681>

Python

[Python] Pandas 사용법 - 인덱싱 접근,데이터 조작, 인덱스조작(loc,iloc)

2020. 10. 16. 00:20 수정 삭제 공개

인덱싱, 데이터 조작, 인덱스 조작

loc()

location 은 label value 기반의 2차원 인덱싱

iloc()

index location 은 순서를 나타내는 정수 기반의 2차원 인덱싱

```
In [5]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
# df.loc[행 인덱싱 값]
# df.iloc[행 인덱싱 값, 열 인덱싱 값]
```

```
In [7]: sample_df = pd.DataFrame(np.arange(10,22).reshape(3,4),
                                index=['a','b','c'],
                                columns = ['A','B','C','D'])
sample_df
```

```
Out[7]:
```

	A	B	C	D
a	10	11	12	13
b	14	15	16	17
c	18	19	20	21

행 인덱스 접근

```
In [9]: sample_df.loc['a']
```

```
Out[9]: A    10
        B    11
        C    12
        D    13
        Name: a, dtype: int32
```

```
In [13]: type(sample_df.loc['a'])
```

```
Out[13]: pandas.core.series.Series
```

loc를 통해 해당 인덱스를 가져온다. type은 series로 된다.

```
In [11]: sample_df.loc['a'].values
```

```
Out[11]: array([10, 11, 12, 13])
```

```
In [14]: type(sample_df.loc['a'].values)
```

```
Out[14]: numpy.ndarray
```

loc에 values를 하면 array의 형태로 가져오고 type은 array가 된다.

```
In [15]: sample_df.loc['b':'c']
```

```
Out[15]:
```

	A	B	C	D
b	14	15	16	17
c	18	19	20	21

```
In [17]: sample_df.loc[['b','c']]
```

```
Out[17]:
```

	A	B	C	D
b	14	15	16	17
c	18	19	20	21

```
In [20]: sample_df['b':'c']
```

```
Out[20]:
```

	A	B	C	D
b	14	15	16	17
c	18	19	20	21

이 3가지는 같은 값을 가져온다.

```
value.loc [ : ]
```

```
value.loc[[ , ]]
```

```
value [ : ]
```

열 인덱스 접근

```
In [22]: sample_df['A']
```

```
Out[22]: a    10  
        b    14  
        c    18  
        Name: A, dtype: int32
```

```
In [23]: sample_df.A
```

```
Out[23]: a    10  
        b    14  
        c    18  
        Name: A, dtype: int32
```

```
In [25]: type(sample_df.A)
```

```
Out[25]: pandas.core.series.Series
```

Boolean index

```
In [28]: sample_df.A>15
```

```
Out[28]: a    False  
        b    False  
        c     True  
        Name: A, dtype: bool
```

```
In [29]: sample_df[sample_df.A>15]
```

```
Out[29]:
```

	A	B	C	D
c	18	19	20	21

이런 식으로 연산을 통해 boolean index를 만들고 이 값을 조건으로 해서 결과를 출력할 수 있다.

ex) 원하는 행열 가져오기

```
In [32]: sample_df.loc['b':'c'].A
```

```
Out[32]: b    14  
         c    18  
         Name: A, dtype: int32
```

```
In [34]: sample_df.loc['b':, 'A']
```

```
Out[34]: b    14  
         c    18  
         Name: A, dtype: int32
```

```
In [36]: sample_df.loc[:, 'a', :]
```

```
Out[36]: A    10  
         B    11  
         C    12  
         D    13  
         Name: a, dtype: int32
```

```
In [41]: sample_df.loc['b':'c', 'C':'D'] # 행 , 열
```

```
Out[41]:
```

	C	D
b	16	17
c	20	21

```
In [42]: sample_df.loc[['b', 'c'], ['C', 'D']] # 행 , 열
```

```
Out[42]:
```

	C	D
b	16	17
c	20	21

iloc

	A	B	C	D
a	10	11	12	13
b	14	15	16	17
c	18	19	20	21

```
In [45]: sample_df.iloc[0,1]
```

```
Out[45]: 11
```

행열이라고 보면 된다. 0번째 행에 1번째 열 을 가져와서 11을 가져온다.

```
In [50]: sample_df.iloc[ 0,2:]
```

```
Out[50]: C    12
         D    13
         Name: a, dtype: int32
```

```
In [49]: sample_df.iloc[ 0,-2:]
```

```
Out[49]: C    12
         D    13
         Name: a, dtype: int32
```

12 , 13 에 대해 이런 식으로 가져올 수 있다.

```
In [51]: sample_df.iloc[ 2,1:3]
```

```
Out[51]: B    19
         C    20
         Name: c, dtype: int32
```

19, 20 을 가져오는 데 있어서 이런 방식도 가능하다.

가져올 수 있는 방법은 다양하다. 자기가 편한 방식으로 하면 될 것 같다.

```
In [53]: sample_df.iloc[-1]
Out[53]: A    18
         B    19
         C    20
         D    21
         Name: c, dtype: int32
```

-1을 통해 마지막 값을 가져올 수도 있다.

```
In [54]: sample_df.iloc[-1] = sample_df.iloc[-1] * 2
```

```
In [55]: sample_df
```

```
Out[55]:
```

	A	B	C	D
a	10	11	12	13
b	14	15	16	17
c	36	38	40	42

series를 통해 연산도 가능하다.

'Python' 카테고리의 다른 글

[Python] Pandas 사용법 - 두가지의 DataFrame 합치기 (merge, join)

[Python] Pandas 사용법 - 다양한 인덱스 함수(reset_index,set_index,sort_index)

[Python] Pandas 사용법 - 인덱싱 접근,데이터 조작, 인덱스조작(loc,iloc)

[Python] Pandas 사용법 - 다양한 함수 사용(데이터 입출력, 대소문자변환, 공백...

[Python] Pandas 사용법 - DataFrame 생성, 추가 , 수정, 삭제, indexing

[Python] Pandas 사용법 - series 에 대한 추가 , 수정, 삭제, 연산, 결측치

pandas iloc

pandas loc

pandas 열 인덱스 접근

pandas 행 인덱스 접근

pandas 행열접근



나무늘보스

혼자 끄적끄적하는 블로그 입니다.