

빅데이터 실습

6주차 2차시

—
데이터 정렬하기

데이터 정렬하기



학습개요

- 1, 데이터 정렬하기(sort_values())
- 2, 인덱스 정렬하기(sort_index())

정렬

어떤 값을 기준으로
어떤 값의 순서를 매기는 행위

sort_values()

데이터의 값을 기준으로
정렬하는 함수

sort_index()

로우 인덱스나 컬럼 인덱스들을
정렬해 주는 함수

01

데이터 정렬과 인덱스 정렬



DataFrame.sort_values

(by=None, → 정렬할 기준 변수
axis=0, → 'index' or 'columns'
(default)
ascending=True,
inplace=False,
kind='quicksort',
na_position='last')

DataFrame.sort_values

(by=None,

axis=0,

ascending=True,

inplace=False,

kind='quicksort',

na_position='last')

→ True : 오름차순
(default)
False : 내림차순

DataFrame.sort_values

(by=None,
axis=0,
ascending=True,
inplace=False,
kind='quicksort',
na_position='last')

→ 정렬 결과를
원본에 반영할 것인지
True : 결과 반영
False : 반영X
(default)

DataFrame.sort_values

(by=None,

axis=0,

ascending=True,

inplace=False,

kind='quicksort', → 정렬 알고리즘

na_position='last') (default)

DataFrame.sort_values

(by=None,
axis=0,
ascending=True,
inplace=False,
kind='quicksort',

na_position='last') → 결측값 위치,
(‘first’, ‘last’)
(default)

DataFrame.sort_values

(by=None, → 필수 지정

axis=0,

ascending=True, → 오름차순, 내림차순

inplace=False,

kind='quicksort',

na_position='last')

DataFrame.sort_index

(by=None,
axis=0,
ascending=True,
inplace=False,
kind='quicksort',
na_position='last')



sort_values와
동일한 인자들을 제공

02

실습



sort_values와 sort_index 함수를 이용한 실습

실습 준비하기

© import

In [1]:

Slide Type

```
import pandas as pd
from pandas import Series, DataFrame
import numpy as np
```

Slide Type

1) Series 정렬

In [2]:

Slide Type

```
# Series 정렬을 위한 샘플 데이터
sr = Series([3.5, 2.1, 7.10], index = list('bcafed'))
```

Series 정렬

◎ Series 정렬을 위한 샘플 데이터

```
# Series 정렬을 위한 샘플 데이터
```

```
sr = Series([3,5,2,1,7,10], index = list('bcafed'))
```

```
sr
```

```
Out[6]: b      3  
       c      5  
       a      2  
       f      1  
       e      7  
       d     10  
       dtype: int64
```

```
In [4]:
```

Slide Type

```
# 로우 인덱스 라벨의 값으로 정렬
```

```
In [1]:
```

Slide Type

Series 정렬

◎ 로우 인덱스 라벨의 값으로 정렬

✓ sort_index 함수 실행 : 인덱스 값을 기준으로 정렬

```
c      5  
e      7  
d     10  
dtype: int64
```

In [7]:

Slide Type

```
# 로우 인덱스 라벨의 값으로 정렬  
sr
```

Out [7]:

```
b      3  
c      5  
a      2  
e      7  
d     10
```

정렬이 되어 있지 않은 상태

Series 정렬

◎ Series의 값으로 정렬

Out [9]:

✓ sort_values 함수 실행 : 기본값 = 오름차순

```
c      5  
d     10  
e      7  
f      1  
dtype: int64
```

In [10]:

Slide Type

```
# Series의 값으로 정렬 (기본값 = 오름차순)  
sr.sort_values()
```

Out [10]:

```
f      1  
a      2  
b      3  
c      5  
e      7
```

Series 정렬

◎ Series의 값으로 정렬

Out[]: sr.sort_values()

Out[]: ✓ sort_values 함수 실행 : ascending = False

```
a    2
b    3
c    5
e    7
d   10
dtype: int64
```

In [11]:

Slide Type

```
# Series의 값으로 내림차순 정렬
sr.sort_values(ascending = False)
```

Out[11]:

```
d    10
e     7
c     5
b     3
a     2
```


DataFrame정렬

◎ 샘플 데이터 생성

dtype: int64

✓ 학생별, 연도별, 과목별 성적을 가지고 있는 데이터 프레임

Slide Type

2) DataFrame 정렬

In []:

Slide Type

Slide

```
# 샘플 데이터 생성
```

```
np.random.seed(0)
```

```
df = pd.DataFrame(np.random.randint(50, 100, (5, 6)),
```

```
                  columns=[2019, 2019, 2019, 2020, 2020, 2020], [ '영어', '수학', '과학' ],
```

```
                  index = [ 'Kim', 'Park', 'Lee', 'Jung', 'Moon' ])
```

```
df.index.set_names('학생명', inplace = True)
```

```
df.columns.set_names(['년도', '과목'], inplace = True)
```

```
df.loc['Moon', (2019, '과학')] = np.nan
```

dtype: int64

Slide Type

2) DataFrame 정렬

In []:

Slide Type Slide

샘플 데이터 생성

np.random.seed(0)

df = pd.DataFrame(np.random.randint(50, 100, (5, 6)),

columns=[2019, 2019, 2019, 2020, 2020, 2020], ['영어', '수학', '과학'])

index=['Kim', 'Park', 'Lee', 'Jung', 'Moon'])

2020년 데이터만 선택을 해서 계층 색인을 제거한 후에 정렬을 해보고,
계층 색인인 경우에 어떻게 정렬을 하는지 살펴보도록 하겠습니다.

df.loc['Moon', (2019, '과학')] = np.nan

© 2020년 성적만 df2020에 저장

Slide Type

- 2차원 DataFrame 정렬
 - df에서 2020년 데이터만 선택하여 df2020 생성한 후 실습

In [14]:

Slide Type

```
df2020 = df[2020]
```

Out [14]:

과목	영어	수학	과학
학생명			

Kim	53	53	89
-----	----	----	----

◎ 학생명(알파벳 순서)으로 정렬

✓ sort_index로 정렬하고, axis = 0 으로 지정

```
# 학생명으로 정렬
```

```
df2020
```

Out[16]:

과목 영어 수학 과학

학생명

Kim	53	53	89
Park	86	73	56
Lee	51	88	89

로우 인덱스인 학생명을 정렬하므로, axis를 0으로 지정합니다.
기본값이 0이므로 생략가능합니다.

◎ 수학 성적으로 오름차순 정렬

주의하기

86 73 56

- ✓ DataFrame의 경우에는 columns이 여러 개 있으므로,
어떤 columns 값을 기준으로 정렬을 할 건지 명시하지 않으면 에러 발생

Slide Type

```
# 수학 성적으로 오름차순 정렬  
df2020.sort_values()
```

TypeError

Traceback (most recent call last)

```
<ipython-input-20-f359b742f531> in <module>
```

```
1 # 수학 성적으로 오름차순 정렬
```

```
----> 2 df2020.sort_values()
```

```
TypeError: sort_values() missing 1 required positional argument: 'by'
```

◎ 수학 성적으로 오름차순 정렬

✓ by 인자에 정렬하고자 하는 columns의 이름 지정

```
# 수학 성적으로 오름차순 정렬  
df2020.sort_values(by = '수학')
```

Out[21]:

과목	영어	수학	과학
학생명			
Kim	53	53	89
Moon	70	66	55
Park	86	73	56
Jung	67	87	75
Lee	51	88	89

◎ 영어 성적의 내림차순 정렬

✓ by = 수학 → 영어로 수정

✓ 내림차순 : ascending = False로 수정

ascending = False)

Out[22]:

과목	영어	수학	과학
학생명			
Park	86	73	56
Moon	70	66	55
Jung	67	87	75
Kim	53	53	89

영어 성적이 높은 사람부터 낮은 사람 순서대로 정렬이 되는 것을 알 수가 있습니다.

계층 색인인 경우 정렬하기

계층 색인은 튜플 형태로 지정

© 2020년 수학 성적 기준으로 내림차순 정렬

✓ 코드에 다 포함되어 있으므로, 코드 가리지 않도록 삭제

```
# 2020년 수학 성적 기준으로 내림차순 정렬
```

```
df.sort_values(by = (2020, '수학'), ascending = False)
```

Out[24]:

년도	2019			2020		
과목	영어	수학	과학	영어	수학	과학
학생명						
Lee	74	74	62.0	51	88	89
Jung	73	96	74.0	67	87	75
Park	59	69	71.0	86	73	56
Moon	63	58	NaN	70	66	55

◎ 2개 이상의 columns으로 정렬하기

✓ 예 2020년 수학 성적으로 정렬하고 점수가 같으면 2019년 수학으로 정렬

2개 이상의 컬럼으로 정렬하기

✓ sort_values의 by인자에 두 개의 컬럼명을 순서대로 지정

Out[25]:

년도	2019			2020		
과목	영어	수학	과학	영어	수학	과학
학생명						
Kim	94	97	50.0	53	53	89
Park	59	69	71.0	86	73	56
Lee	74	74	62.0	51	88	89

파이썬에서는 두 개 이상의 데이터를 전달을 할 때는 항상 리스트로 담아서 전달을 합니다.

◎ 2개 이상의 columns으로 정렬하기

✓ 예 2020년 수학 성적으로 정렬하고 점수가 같으면 2019년 수학으로 정렬

2개 이상의 컬럼으로 정렬하기

```
df.sort_values(by = [(2020, '수학'), (2019, '수학')], ascending = [False, True])
```

Out[27]:

✓ 정렬 방식이 다를 경우, ascending 값을 컬럼 개수에 맞게 지정

과목 영어 수학 과학 영어 수학 과학

학생명

학생명	영어	수학	과학	영어	수학	과학
Lee	74	74	62.0	51	88	89
Jung	73	96	74.0	67	87	75
Park	59	69	71.0	86	73	56
Moon	63	58	NaN	70	66	55

◎ 정렬 결과를 원본에 반영하기

✓ inplace = True

```
(by = (2020, '수학'), ascending = False, inplace = True)
```

In [30]:

Slide Type

df

Out [30]:

년도	2019			2020		
과목	영어	수학	과학	영어	수학	과학
학생명						
Lee	74	74	62.0	51	88	89
Jung	73	96	74.0	67	87	75
Park	59	69	71.0	86	73	56
M	66	59	N/A	79	66	55