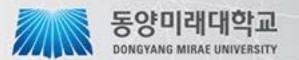


동양미래대학교 DONGYANG MIRAE UNIVERSITY

# 

6주차 2차시

데이터 정렬하기

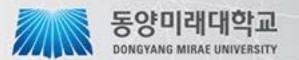


# 데이터 정렬하기



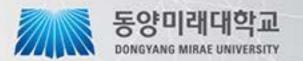
# 학습개요

- 1/ 데이터 정렬하기(sort\_values())
- 2/ 인덱스 정렬하기(sort\_index())



# 정블

어떤 값을 기준으로 어떤 값의 순서를 매기는 행위



# sort\_values()

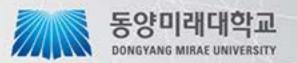
데이터의 값을 기준으로 정렬하는 함수

sort\_index()

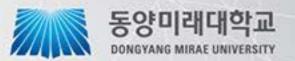
로우 인덱스나 컬럼 인덱스들을 정렬해 주는 함수







(by=None, → 정렬할기준변수 axis=0, → 'index' or 'columns' ascending=True, (default) inplace=False, kind='quicksort', na\_position='last')



(by=None, axis=0,

ascending=True, → True : 오름차순

inplace=False,

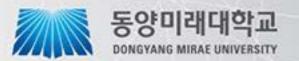
kind='quicksort',

na\_position='last')

(default)

False: 내림차순





(by=None,

axis=0,

ascending=True,

# inplace=False, → 정렬 결과를

kind='quicksort',

na\_position='last')

원본에 반영할 것인지

True: 결과 반영

False : 반영X

(default)





(by=None,

axis=0,

ascending=True,

inplace=False,

kind='quicksort', → 정렬알고리즘
na\_position='last') (default)

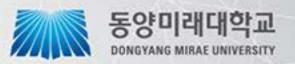




```
(by=None,
axis=0,
ascending=True,
inplace=False,
kind='quicksort',
```

na\_position='last') > 결측값위치, ('first', 'last') (default)





(by=None, → 필수지정

axis=0,

ascending=True, → 오름차순, 내림차순

inplace=False,

kind='quicksort',

na\_position='last')

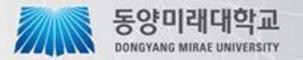


# DataFrame.sort\_index

(by=None, axis=0, ascending=True, inplace=False, kind='quicksort', na\_position='last') —

sort\_values와 동일한인자들을제공





# sort\_values와 sort\_index 함수를 이용한 실습

# 실습 준비하기

# import

```
In [1]:
                                                                           Slide Type Slide
        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 정렬을 위한 샘플 데이터

```
# Series 싱물들 위안 샘들 네이터
       sr = Series([3,5,2,1,7,10], index = list('bcafed'))
       sr
Out[6]:
            10
       dtype: int64
In [4]:
                                                                  Slide Type
       #로우 인덱스 라벨의 값으로 정렬
```

г г 1.

OI: 1 T

# ◎ 로우 인덱스 라벨의 값으로 정렬

```
▼ sort_index 함수실행:인덱스값을기준으로정렬
e 7
d 10
dtype: int64
```

```
In [7]: # 로우 인덱스 라벨의 값으로 정렬 sr
```



# ◎ Series의 값으로 정렬

```
Out Tol.
    ✔ sort_values 함수실행:기본값 = 오름차순
            10
        dtype: int64
In [10]:
                                                                  Slide Type
        # Series의 값으로 정렬 (기본값 = 오름차순)
        sr.sort_values()
Out [10]:
```

# ◎ Series의 값으로 정렬

```
✓ sort_values 함수실행 : ascending = False
        dtype: int64
In [11]:
                                                                      Slide Type
        # Series의 값으로 내림차순 정렬
        sr.sort_values(ascending = False)
Out[11]: d
             10
```

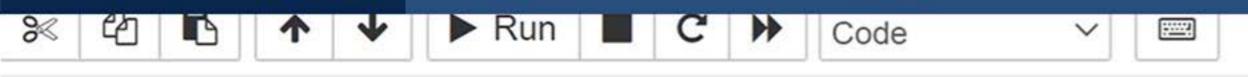
# DataFrame정렬

## ◎ 샘플 데이터 생성

▼ 학생별, 연도별, 과목별 성적을 가지고 있는 데이터 프레임

Slide Type

# 2) DataFrame 정렬



dtype: int64

Slide Type

# 2) DataFrame 정렬

계층 색인인 경우에 어떻게 정렬을 하는지 살펴보도록 하겠습니다.

df.loc['Moon', (2019, '과학')] = np.nan

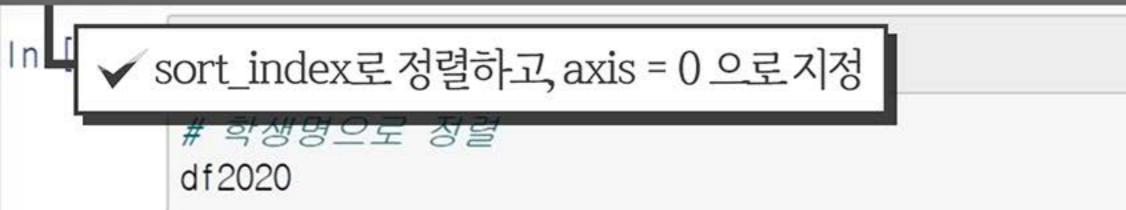
◎ 2020년 성적만 df2020에 저장

```
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 89 53

# ◎ 학생명(알파벳 순서)으로 정렬



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.

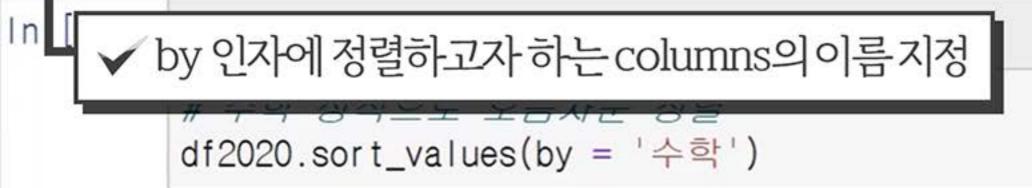
Slide Type

◎ 수학 성적으로 오름차순 정렬

```
86
                    56
주의하기
  DataFrame의경우에는 columns이 여러개있으므로,
                                                             Slide Type
  어떤 columns 값을 기준으로 정렬을 할 건지 명시하지 않으면 에러 발생
   df2020.sort_values()
                                        Traceback (most recent call last)
   TypeError
   <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



Slide Type

### Out[21]:

학생명			
Kim	53	53	₿ 89
Moon	70	66	55
Park	86	73	56
Jung	67	87	75
Lee	51	88	89

과목 영어 수학 과학

영어 수학 과학

# ◎ 영어 성적의 내림차순 정렬

과목

```
Slide Type

✓ by = 수학→영어로 수정

✓ 내림차순: ascending = False로 수정
scending = False)
```

#### Out [22]:

56
55
75
89

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

# 계층 색인인 경우 정렬하기

계층 색인은 튜플 형태로 지정

# DataFrame정렬 3차원 DataFrame 정렬(계층 색인)

# ◎ 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)

Slide Type

#### Out [24]:

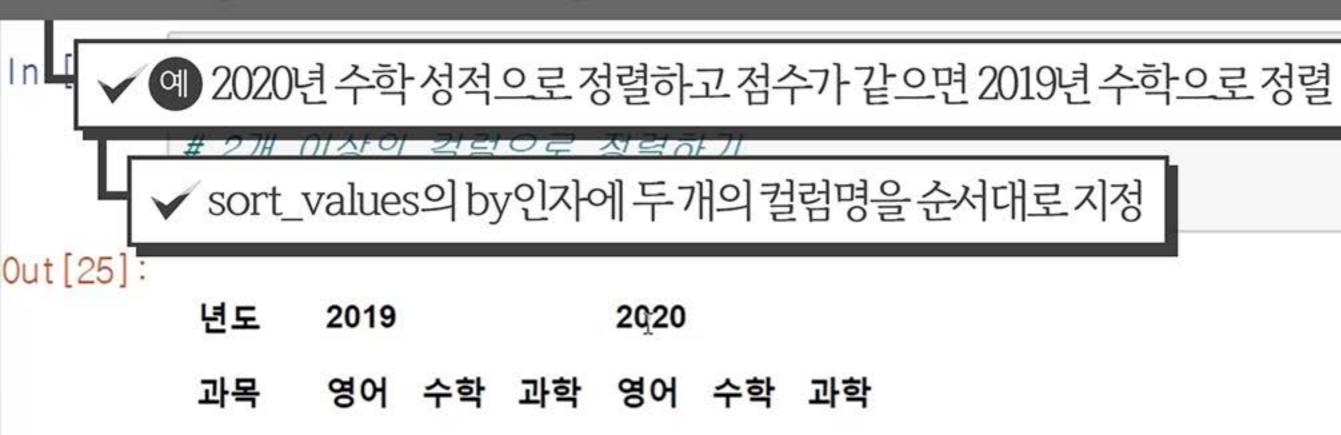
년도	2019			2020		
과목	영어	수학	과학	영어	수학	과학
학생명						
Lee	74	74	62.0	51	88	89
Jung	73	96	74.0	67	87	75
Park	59	69	71.0	86	73	56
Moon	63	58	NaN	70	66	55

# DataFrame정렬

# 3차원 DataFrame 정렬(계층 색인)

Slide Type

◎ 2개 이상의 columns으로 정렬하기



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.

# DataFrame정렬

# 3차원 DataFrame 정렬(계층 색인)

◎ 2개 이상의 columns으로 정렬하기

예 2020년 수학 성적으로 정렬하고 점수가 같으면 2019년 수학으로 정렬 Slide Type # 2개 이상의 걸럼으로 정달하기 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

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

# DataFrame정렬 3차원 DataFrame 정렬(계층 색인)

# ◎ 정렬 결과를 원본에 반영하기

```
결과를 권존에 만영하기
                     by = (2020, '수학'), ascending = False, inplace = True)
       inplace = True
In [30]:
                                                                     Slide Type
        df
```

#### Out [30]:

년도	2019			2020		
과목	영어	수학	과학	영어	수학	과학
학생명						54
Lee	74	74	62.0	51	88	89
Jung	73	96	74.0	67	87	75
Park	59	69	71.0	86	73	56
				70		