6. pandas –연산

이번시간에 학습할 내용은..

Series, DataFrame에 대한 연산, 요약 통계함수 사용방법설명

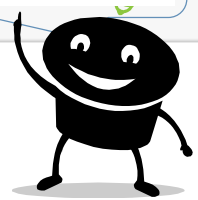
I. 산술연산 작업

➢ 객체내의 원소들에 대한 산술연산작업은 물론, 서로 다른 객체간의 산술연산작업도 지원됨.

객체에서 각 원소에 대한 연산작업

```
In [1]: from pandas import Series,DataFrame

In [2]: p1=Series([10,15,17],index=['1라인','2라인','3라인'])
        p1

Out[2]: 1라인    10
        2라인    15
        3라인    17
        dtype: int64

In [3]: p2=p1*2
        p2

Out[3]: 1라인    20
        2라인    30
        3라인    34
        dtype: int64
```

이렇게 p1객체내의 모든 원소에 대해 산술작업이 실행돼요.



```
In [4]: p2=Series([1.7,2.5,3.3,4.2],index=['1라인','2라인','3라인','4라인'])
        p1+p2

Out[4]: 1라인    11.7
        2라인    17.5
        3라인    20.3
        4라인     NaN
        dtype: float64
```

색인이 일치하지 않는 경우 이렇게 결측값 NaN이 표시돼요.

산술연산 메소드를 이용한 객체간 연산작업

```
In [3]: df2=DataFrame(np.ones((4,4),dtype='int32'),index=['one','two','three','four'])
        df2
```

Out[3]:

	0	1	2	3
one	1	1	1	1
two	1	1	1	1
three	1	1	1	1
four	1	1	1	1

```
In [4]: df1.add(df2)
```

Out[4]:

	0	1	2	3
four	NaN	NaN	NaN	NaN
one	1.0	1.0	1.0	NaN
three	1.0	1.0	1.0	NaN
two	1.0	1.0	1.0	NaN

이렇게 df1에 df1과 df2를 더하는 작업을 메소드 이용하여 실행할 수 있어요.
산술연산 메소드는 아래와 같아요.

- 덧셈 –add
- 뺄셈 – sub
- 곱셈 – mul
- 나눗셈 - div

2. 통계 계산 및 요약작업

➢ pandas는 Series, DataFrame에 대해 컬럼, 행 에 대한 계산 및 요약작업을 가능하게끔 다양한 함수 제공

주요 통계 및 요약 메소드

메소드	기능
count	결측값을 제외한 값의 수를 반환
sum	합 계산
mean	산술 평균 계산
std / var	표준편차 계산 / 분산계산
min / max	최소 값 / 최대 값 계산
Median	중위수 반환
Cumsum	누적값 계산
Cumprod	누적곱 계산
describe	각 컬럼에 대한 요약통계를 계산

이와 같은 데이터프레임객체 df가 있을 때

```
In [1]: from pandas import DataFrame

In [2]: df=DataFrame({'거래처':['a슈퍼','b슈퍼','c슈퍼','a슈퍼','c슈퍼','c슈퍼'],
                      '주문량':[100,25,40,55,123,77],
                      '할인율':[0.1,0.3,0.03,0.07,0.15,0.07]})
        df
```

Out[2]:

	거래처	주문량	할인율
0	a슈퍼	100	0.10
1	b슈퍼	25	0.30
2	c슈퍼	40	0.03
3	a슈퍼	55	0.07
4	c슈퍼	123	0.15
5	c슈퍼	77	0.07

기본 통계정보를 보여주는 –describe

```
In [3]: df.describe()
```

Out[3]:

	주문량	할인율
count	6.00000	6.000000
mean	70.00000	0.120000
std	37.16988	0.096747
min	25.00000	0.030000
25%	43.75000	0.070000
50%	66.00000	0.085000
75%	94.25000	0.137500
max	123.00000	0.300000

합계와 평균을 계산하는 –sum, mean

```
In [4]: df.sum()
```

```
Out[4]: 거래처      a슈퍼b슈퍼c슈퍼a슈퍼c슈퍼c슈퍼
        주문량                        420
        할인율                       0.72
        dtype: object
```

```
In [5]: df.mean()
```

```
Out[5]: 주문량    70.00
        할인율     0.12
        dtype: float64
```

누적을 계산하는cumsum

```
In [11]: df.cumsum()
```

Out[11]:

	거래처	주문량	할인율
0	a슈퍼	100	0.1
1	a슈퍼b슈퍼	125	0.4
2	a슈퍼b슈퍼c슈퍼	165	0.43
3	a슈퍼b슈퍼c슈퍼a슈퍼	220	0.5
4	a슈퍼b슈퍼c슈퍼a슈퍼c슈퍼	343	0.65
5	a슈퍼b슈퍼c슈퍼a슈퍼c슈퍼c슈퍼	420	0.72

특정 값이 몇 개 존재하는지(도수)를 계산하는 –value_counts()

```
In [14]: df['거래처'].value_counts()
```

```
Out[14]: c슈퍼    3
         a슈퍼    2
         b슈퍼    1
         Name: 거래처, dtype: int64
```

중복되는 값을 제거하여 특정컬럼의 값을 보여주는–unique()

```
In [15]: df['거래처'].unique()
```

```
Out[15]: array(['a슈퍼', 'b슈퍼', 'c슈퍼'], dtype=object)
```

3. 결측 데이터 처리

- 결측값의 문제를 어떻게 처리하느냐에 따라 데이터 분석결과에 영향을 미치므로 결측값의 존재유무부터 확인하는 게 중요
- 'none'값도 결측값으로 인식하며, 결측값인 경우 'NaN'으로 표시
- 결측값 존재유무 확인 메소드 – isnull(), notnull()

결측데이터 유무 확인하기

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

In [4]: df=pd.DataFrame(np.ones((4,4)),index=['a','b','c','d'])
        df
```

Out[4]:

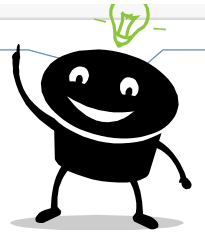
	0	1	2	3
a	1.0	1.0	1.0	1.0
b	1.0	1.0	1.0	1.0
c	1.0	1.0	1.0	1.0
d	1.0	1.0	1.0	1.0

```
In [8]: df[1]['b']=np.nan
        df[3]['b':]=np.nan
        df
```

Out[8]:

	0	1	2	3
a	1.0	1.0	1.0	1.0
b	1.0	NaN	1.0	NaN
c	1.0	1.0	1.0	NaN
d	1.0	1.0	1.0	NaN

이렇게 실습용 데이터셋을 만들어요.
결측치 데이터들이 여러 개 생성됐어요.



```
In [9]: pd.isnull(df)
```

Out[9]:

	0	1	2	3
a	False	False	False	False
b	False	True	False	True
c	False	False	False	True
d	False	False	False	True

결측치면 True표시

결측치를 제외한 정상 데이터만 표시

```
In [12]: df1=df.dropna()
         df1
```

Out[12]:

	0	1	2	3
a	1.0	1.0	1.0	1.0

결측치가 존재하는 행은 제외시키고 결과를 표시함.

결측데이터 다른 값으로 채우기 – fillna()메소드

결측치를 무조건 1로 채우기

```
In [13]: df.fillna(1)
```

Out[13]:

	0	1	2	3
a	1.0	1.0	1.0	1.0
b	1.0	1.0	1.0	1.0
c	1.0	1.0	1.0	1.0
d	1.0	1.0	1.0	1.0

컬럼1의 결측값=3,컬럼3의 결측값=7로 대체

```
In [14]: df.fillna({1:3,3:7})
```

Out[14]:

	0	1	2	3
a	1.0	1.0	1.0	1.0
b	1.0	3.0	1.0	7.0
c	1.0	1.0	1.0	7.0
d	1.0	1.0	1.0	7.0