결측치 처리

- 결측치 : 비어있는 값. 분석 결과 왜곡
- 결측치 처리

1) 행제거 또는 2) 값을 채워넣음(대표값 또는 예측값으로)

In [31]:

```
import pandas as pd

age      = pd.Series([ None,  42, 27,  25,  20]  )
score    = pd.Series([3.8, 4.2, 2.6, 1.0, 3.0] )
salary =  pd.Series([2700,4000,3000,2700,3200])
stu_class = pd.Categorical([ 1, 1, 2, None, 2])   # None 결측치
gender    = pd.Categorical([ 'F', 'M', 'M', 'M', None ])   # None 결측치
```

In [32]:

```
df = pd.DataFrame ( {'age': age,
                     'score' :  score  ,
                     'salary' : salary,
                      'class' :stu_class,
                       'gender' : gender}
)
df
```

Out[32]:

	age	score	salary	class	gender
0	NaN	3.8	2700	1	F
1	42.0	4.2	4000	1	M
2	27.0	2.6	3000	2	M
3	25.0	1.0	2700	NaN	M
4	20.0	3.0	3200	2	NaN

In [33]:

```
# df 복제
df_new = df.copy()
```

- 결측치값 확인

In [34]:

```
#데이터프레임의 모든 값이 boolean 형태로 표시되도록 하며, nan인 값에만 True가 표시되게 하는 함수
pd.isna(df_new)
```

Out[34]:

	age	score	salary	class	gender
0	True	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	True	False
4	False	False	False	False	True

In [35]:

```
df_new.isnull().sum()
# 열별로 결측치 갯수 확인
```

Out[35]:

```
age       1
score     0
salary    0
class     1
gender    1
dtype: int64
```

- 결측치 처리

1) 행제거

In [36]:

```
#결측치를 가지고 있는 행들을 삭제
df_new = df_new.dropna(how='any')
df_new
```

Out[36]:

	age	score	salary	class	gender
1	42.0	4.2	4000	1	M
2	27.0	2.6	3000	2	M

- 결측치 처리

2) 다른 값으로 채우기

In [44]:

```
# df 복제
df_new = df.copy()
```

In [45]:

```
# age 열의 결측치를 다른 값으로 채우기 (대표값 또는 예측값을 구한 후)
df_new["age"] = df_new["age"].fillna(value=30)
```

In [46]:

```
df_new
```

Out[46]:

	age	score	salary	class	gender
0	30.0	3.8	2700	1	F
1	42.0	4.2	4000	1	M
2	27.0	2.6	3000	2	M
3	25.0	1.0	2700	NaN	M
4	20.0	3.0	3200	2	NaN