데이터 정제 및 가공

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

In [2]:
```
#연습문제 가져오기
```

In [2]:
```
df = pd.read_csv("연습데이터셋1.csv", encoding = 'cp949')
df.head()
```

Out[2]:

	student_id	name	age	score	grade	city	registration_date
0	S001	이민준	23.0	85.5	B	서울	2023-01-15
1	S002	박서연	25.0	92.0	A	부산	2023-02-20
2	S003	김지훈	NaN	78.0	C	서울	2023-03-10
3	S004	최예은	22.0	NaN	B	인천	2024-04-05
4	S005	정하준	28.0	95.5	A	미상	2024-05-12

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8 entries, 0 to 7
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   student_id         8 non-null      object 
 1   name               8 non-null      object 
 2   age                7 non-null      float64
 3   score              7 non-null      float64
 4   grade              7 non-null      object 
 5   city               8 non-null      object 
 6   registration_date  7 non-null      object 
dtypes: float64(2), object(5)
memory usage: 576.0+ bytes
```

df.isnull(), df.dropna(), df.fillna ()

In [4]:
```
#null값 총합
df.isnull().sum()
```

Out[4]:
```
student_id          0
name                0
age                 1
score               1
grade               1
city                0
registration_date    1
dtype: int64
```

In [ ]:
```

```

In [5]:
```
#행 전체 삭제 (dropna)
drop = df.dropna()
drop.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 5 entries, 0 to 7
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   student_id         5 non-null      object 
 1   name               5 non-null      object 
 2   age                5 non-null      float64
 3   score              5 non-null      float64
 4   grade              5 non-null      object 
 5   city               5 non-null      object 
 6   registration_date  5 non-null      object 
dtypes: float64(2), object(5)
memory usage: 320.0+ bytes
```

In [6]:
```
# 전체 열 삭제 (dropna)
co_drop = df.dropna(axis=1)
co_drop.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8 entries, 0 to 7
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   student_id  8 non-null      object
 1   name        8 non-null      object
 2   city        8 non-null      object
dtypes: object(3)
memory usage: 320.0+ bytes
```

In [ ]:

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8 entries, 0 to 7
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   student_id         8 non-null      object 
 1   name               8 non-null      object 
 2   age                7 non-null      float64
 3   score              7 non-null      float64
 4   grade              7 non-null      object 
 5   city               8 non-null      object 
 6   registration_date  7 non-null      object 
dtypes: float64(2), object(5)
memory usage: 576.0+ bytes
```

In [8]:
```
df.head()
```

Out[8]:

	student_id	name	age	score	grade	city	registration_date
0	S001	이민준	23.0	85.5	B	서울	2023-01-15
1	S002	박서연	25.0	92.0	A	부산	2023-02-20
2	S003	김지훈	NaN	78.0	C	서울	2023-03-10
3	S004	최예은	22.0	NaN	B	인천	2024-04-05
4	S005	정하준	28.0	95.5	A	미상	2024-05-12

Fillna 예제

예제 1: 전체 컬럼

- df2=df.fillna('None')

예제 2: 컬럼 하나

- df2['Discount'] = df['Discount'].fillna(0)

예제 3: 그룹 컬럼

- df2[['Discount','Fee']] = df[['Discount','Fee']].fillna(0)

예제 4: 그룹 컬럼인데 다른 값

- df2 = df.fillna(value={'Discount':0,'Fee':10000})

예제 5: 제한(limit)

- df2=df.fillna(value={'Discount':0,'Fee':0},limit=1)

In [9]:
```
df_fill = df.fillna(value = {'age':30, 'score':80.0, 'grade': 'None', 'registration_date' :'2023-06-30' })
df_fill.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8 entries, 0 to 7
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   student_id         8 non-null      object 
 1   name               8 non-null      object 
 2   age                8 non-null      float64
 3   score              8 non-null      float64
 4   grade              8 non-null      object 
 5   city               8 non-null      object 
 6   registration_date  8 non-null      object 
dtypes: float64(2), object(5)
memory usage: 576.0+ bytes
```

Astype

Astype

In [10]:
```
df_fill.dtypes
```

Out[10]:
```
student_id            object
name                  object
age                  float64
score                float64
grade                 object
city                  object
registration_date     object
dtype: object
```

- object ('Kelly','James')
- float64 (3.14,2.5)
- int(4,5,6,7,8,9)

In [11]:
```
da = df_fill.astype({'score':'object','age':'object'})
da.dtypes
```

Out[11]:
```
student_id           object
name                 object
age                  object
score                object
grade                object
city                 object
registration_date    object
dtype: object
```

df.replace

In [12]:
```
df.head()
```

Out[12]:

	student_id	name	age	score	grade	city	registration_date
0	S001	이민준	23.0	85.5	B	서울	2023-01-15
1	S002	박서연	25.0	92.0	A	부산	2023-02-20
2	S003	김지훈	NaN	78.0	C	서울	2023-03-10
3	S004	최예은	22.0	NaN	B	인천	2024-04-05
4	S005	정하준	28.0	95.5	A	미상	2024-05-12

In [13]:
```
df['city'].value_counts()
```

Out[13]:
```
city
서울    3
부산    2
인천    1
미상    1
광주    1
Name: count, dtype: int64
```

In [14]:
```
df['city'] = df['city'].replace({'서울':'Seoul', '부산': 'Busan', '인천':'Incheon','미상':'None','광주':'Gwangju'})
df.head()
```

Out[14]:

	student_id	name	age	score	grade	city	registration_date
0	S001	이민준	23.0	85.5	B	Seoul	2023-01-15
1	S002	박서연	25.0	92.0	A	Busan	2023-02-20
2	S003	김지훈	NaN	78.0	C	Seoul	2023-03-10
3	S004	최예은	22.0	NaN	B	Incheon	2024-04-05
4	S005	정하준	28.0	95.5	A	None	2024-05-12

필터링

In [15]:
```
df_fill[df_fill['score']>80]
```

Out[15]:

	student_id	name	age	score	grade	city	registration_date
0	S001	이민준	23.0	85.5	B	서울	2023-01-15
1	S002	박서연	25.0	92.0	A	부산	2023-02-20
4	S005	정하준	28.0	95.5	A	미상	2024-05-12
6	S007	한도윤	29.0	88.0	B	광주	2024-07-21

In [16]:
```
#And
```

In [16]:
```
#And
df_fill[(df_fill['score'] > 80) & (df_fill['age'] >= 25)]
```

Out[16]:

	student_id	name	age	score	grade	city	registration_date
1	S002	박서연	25.0	92.0	A	부산	2023-02-20
4	S005	정하준	28.0	95.5	A	미상	2024-05-12
6	S007	한도윤	29.0	88.0	B	광주	2024-07-21

In [17]:
```
#Or
df_fill[(df_fill['score'] > 80) | (df_fill['age'] >= 25)]
```

Out[17]:

	student_id	name	age	score	grade	city	registration_date
0	S001	이민준	23.0	85.5	B	서울	2023-01-15
1	S002	박서연	25.0	92.0	A	부산	2023-02-20
2	S003	김지훈	30.0	78.0	C	서울	2023-03-10
4	S005	정하준	28.0	95.5	A	미상	2024-05-12
5	S006	윤채원	25.0	-1.0	None	부산	2023-06-30
6	S007	한도윤	29.0	88.0	B	광주	2024-07-21

인덱스 설정 및 초기화

In [16]:
```
df.head()
```

Out[16]:

	student_id	name	age	score	grade	city	registration_date
0	S001	이민준	23.0	85.5	B	Seoul	2023-01-15
1	S002	박서연	25.0	92.0	A	Busan	2023-02-20
2	S003	김지훈	NaN	78.0	C	Seoul	2023-03-10
3	S004	최예은	22.0	NaN	B	Incheon	2024-04-05
4	S005	정하준	28.0	95.5	A	None	2024-05-12

In [18]:
```
ds = df.set_index('student_id')
ds
```

Out[18]:

student_id	name	age	score	grade	city	registration_date
S001	이민준	23.0	85.5	B	Seoul	2023-01-15
S002	박서연	25.0	92.0	A	Busan	2023-02-20
S003	김지훈	NaN	78.0	C	Seoul	2023-03-10
S004	최예은	22.0	NaN	B	Incheon	2024-04-05
S005	정하준	28.0	95.5	A	None	2024-05-12
S006	윤채원	25.0	-1.0	NaN	Busan	NaN
S007	한도윤	29.0	88.0	B	Gwangju	2024-07-21
S008	신유나	21.0	76.5	C	Seoul	2024-08-30

In [19]:
```
dr = df.reset_index(drop = True)
dr
```

Out[19]:

	student_id	name	age	score	grade	city	registration_date
0	S001	이민준	23.0	85.5	B	Seoul	2023-01-15
1	S002	박서연	25.0	92.0	A	Busan	2023-02-20
2	S003	김지훈	NaN	78.0	C	Seoul	2023-03-10
3	S004	최예은	22.0	NaN	B	Incheon	2024-04-05
4	S005	정하준	28.0	95.5	A	None	2024-05-12
5	S006	윤채원	25.0	-1.0	NaN	Busan	NaN
6	S007	한도윤	29.0	88.0	B	Gwangju	2024-07-21
7	S008	신유나	21.0	76.5	C	Seoul	2024-08-30

시계열 데이터 처리

In [20]:
```
df_fill.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8 entries, 0 to 7
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   student_id         8 non-null      object
 1   name               8 non-null      object
 2   age                8 non-null      float64
 3   score              8 non-null      float64
 4   grade              8 non-null      object
 5   city               8 non-null      object
 6   registration_date  8 non-null      object
dtypes: float64(2), object(5)
memory usage: 576.0+ bytes
```

In [21]:
```
df_fill['registration_date'] = pd.to_datetime(df_fill['registration_date'])
df_fill.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8 entries, 0 to 7
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   student_id         8 non-null      object
 1   name               8 non-null      object
 2   age                8 non-null      float64
 3   score              8 non-null      float64
 4   grade              8 non-null      object
 5   city               8 non-null      object
 6   registration_date  8 non-null      datetime64[ns]
dtypes: datetime64[ns](1), float64(2), object(4)
memory usage: 576.0+ bytes
```

In [ ]:
```

```