

Pandas 데이터 변형

pandas 단일 데이터프레임 변형을 다룬다.

3

단일 데이터프레임

예제 데이터

http://203.247.53.31/Stat_Notes/example_data/US_crime.csv

A	B	C	D	E	F	G	H	I	J	K	L
주이름	도시이름	인구수	폭력사건	살인	강간	강도	폭행	재산범죄	주거침입	절도	차량절도
ALABAMA	Abbeville	2645	11	1	1	2	7	63	21	39	3
ALABAMA	Adamsville	4481	19	1	0	7	11	321	58	252	11
ALABAMA	Addison	744	1	0	1	0	0	25	6	17	2
ALABAMA	Alabaster	31170	44	0	2	11	31	640	70	544	26
ALABAMA	Alexander Cit	14692	119	2	16	12	89	661	121	510	30
ALABAMA	Aliceville	2419	7	0	0	2	5	48	16	26	6

```
import pandas as pd
url='http://203.247.53.31/Stat_Notes/example_data/US_crime.csv'
crime=pd.read_csv(url, encoding='ms949')
```

```
crime.columns

Index(['주이름', '도시이름', '인구수', '폭력사건', '살인', '강간', '강도', '폭행', '재산범죄', '주거침입', '절도', '차량절도'],
      dtype='object')
```

```
crime.head(2)
```

	주이름	도시이름	인구수	폭력사건	살인	강간	강도	폭행	재산범죄	주거침입	절도	차량절도
0	ALABAMA	Abbeville	2645.0	11.0	1	1.0	2	7.0	63.0	21.0	39.0	3
1	ALABAMA	Adamsville	4481.0	19.0	1	0.0	7	11.0	321.0	58.0	252.0	11

변수 이름 변경

```
df.rename(columns={'기존이름':'새이름'}, inplace=True)
```

```
crime.rename(columns={'주이름':'State'},inplace=True)
crime.columns

Index(['State', '도시이름', '인구수', '폭력사건', '살인', '강간', '강도', '폭행', '재산범죄', '주거침입', '절도', '차량절도'],
      dtype='object')
```

빈도분석 freq table (문자형 변수)

```
PDSeries.value_counts()
```

문자형 변수는 빈도분석 결과를 출력한다.

```
crime.State.value_counts()
```

PENNSYLVANIA	853
TEXAS	604
ILLINOIS	506
NEW JERSEY	479
CALIFORNIA	462
MISSOURI	445
NEW YORK	348
MICHIGAN	343
OHIO	343
WISCONSIN	304

결과를 저장하면 pandas Series 형식으로 저장된다.

인덱스 숫자 0, 1, ... 아닌 주 이름으로 된다.

```
freq=crime.State.value_counts()
freq.head(3)
```

```
PENNSYLVANIA 853
TEXAS        604
ILLINOIS     506
Name: State, dtype: int64
```

주이름 인덱스를 열변수로 만들려면 우선 Series 형식을 데이터프레임 형식으로 변환 한 후 인덱스를 열 이름으로 해야 한다.

```
pd.DataFrame(PDSeries)
```

```
DF.reset_index(inplace=True)
```

```
freq=pd.DataFrame(crime.State.value_counts())
freq.reset_index(inplace=True)
freq.head(3)
```

	index	State
0	PENNSYLVANIA	853
1	TEXAS	604
2	ILLINOIS	506

데이터 크기 순 정렬

```
DF.sort_values(by=['정렬변수','정렬변수2'], ascending=False,
na_position='first', inplace=True)
```

- by=[] : by=을 사용하지 않아도 된다.
- inplace=True : 정렬 결과가 동일 데이터프레임 이름으로 저장된다.
- ascending=True : 디폴트이므로 크기 순 정렬이면 사용하지 않아도 된다. 단 지 역순이면 ascending=False 사용한다.
- na_position='first' : 결측치가 정렬 제일 앞에 나오게 함, 디폴트는 마지막에 출력함

```
crime.sort_values(by=['주이름','인구수'], inplace=True)
```

```
crime.head(2)
```

	주이름	도시이름	인구수	폭력사건	살인	강간	강도	폭행	재산범죄
113	ALABAMA	McIntosh	228.0	10.0	0	0.0	0	10.0	22.0
10	ALABAMA	Arley	353.0	3.0	0	0.0	0	3.0	27.0

```
crime.sort_values(['주이름','인구수'], ascending=False, inplace=True)
```

```
crime.head(2)
```

	주이름	도시이름	인구수	폭력사건	살인	강간	강도	폭행	재산범죄
9259	WYOMING	Cheyenne	62149.0	133.0	1	NaN	12	108.0	2065.0
9258	WYOMING	Casper	58688.0	88.0	2	NaN	15	64.0	2036.0

df.groupby() 활용

1) 범주형 데이터에 의해 데이터 분리

`newdf=df.groupby(['변수명1', '변수명2', ...])` - 테이블 형식으로 데이터 분리

`newdf.first()` - 각 그룹의 첫 관측치 보여줌

```
state=crime.groupby('주이름')
```

```
state.first()
```

주이름	도시이름	인구수	폭력사건	살인	강간
ALABAMA	Mobile4	250557.0	1541.0	29	88.0
ALASKA	Anchorage	299455.0	2435.0	14	408.0
ARIZONA	Phoenix	1502120.0	2402.0	118	14.0

`state.get_group('범주')`

```
state.get_group('WYOMING')
```

	주이름	도시이름	인구수	폭력사건	살인	강간	강도	폭행
9259	WYOMING	Cheyenne	62149.0	133.0	1	NaN	12	108.0
9258	WYOMING	Casper	58688.0	88.0	2	NaN	15	64.0

2) 기초통계량 활용

`crime.groupby(['그룹변수명']).통계량()` 그룹 변수에 의해 통계량 요약

- 통계량 mean, std, var, std, max, min, mode, skew, kurt, cumsum

```
state.get_group('WYOMING')
```

	주이름	도시이름	인구수	폭력사건	살인	강간	강도	폭행	재산
9259	WYOMING	Cheyenne	62149.0	133.0	1	NaN	12	108.0	206
9258	WYOMING	Casper	58688.0	88.0	2	NaN	15	64.0	203

`crime.groupby(['주이름']).quantile(q=?)` ?%퍼센트 백분위 값

- q=0.75는 3사분위 값, 즉 75%백분위값

```
crime.groupby(['주이름']).quantile(q=0.75)
```

0.75	인구수	폭력사건	살인	강간	강도	폭행
주이름						
ALABAMA	12005.00	47.00	1.00	5.00	8.00	30.25
ALASKA	7863.50	59.50	0.00	8.00	2.00	51.00
ARIZONA	47464.50	124.50	1.00	30.00	19.50	87.50

요약 통계량 향후 분석을 위한 인덱스 변수화 방법

- 주이름으로 그룹 만들어 그룹 내 사분위값을 계산하였음

```
crime00=crime.groupby('주이름').quantile(q=0.75)
crime00.head(3)
```

	0.75	인구수	폭력사건	살인	강간	강도	폭행	재산범죄
주이름								
ALABAMA		12005.0	47.0	1.0	5.0	8.0	30.25	529.5
ALASKA		7863.5	59.5	0.0	8.0	2.0	51.00	258.0
ARIZONA		47464.5	124.5	1.0	30.0	19.5	87.50	1474.5

- 요약 테이블 데이터프레임을 보면 '주이름'이 인덱스 이름으로 가 있어 변수 리스트에는 없다.

```
crime00.columns
```

```
Index(['인구수', '폭력사건', '살인', '강간', '강도', '폭행', '재산범죄', '주거침입', '차량절도'], name=0.75)
```

- 인덱스 이름을 변수이름으로 하려면 reset_index() 사용하면 된다.
inplace=True는 결과가 바로 동일 이름으로 저장된다.

```
crime00.reset_index(inplace=True)
crime00.columns
```

```
Index(['주이름', '인구수', '폭력사건', '살인', '강간', '강도', '폭행', '재산범죄', '주거침입', '차량절도'], dtype='object', name=0.75)
```

행 단위 변형

행 (개체 관측값) Subset

1) 조건 활용

DF.head(n) - 첫 n개 행 지정, DF.drop(n) - 마지막 n개 행 지정

```
import pandas as pd
url='http://203.247.53.31/Stat_Notes/example_data/US_crime.csv'
crime=pd.read_csv(url, encoding='ms949')
```

```
crime.head(3)
```

	주이름	도시이름	인구수	폭력사건	살인	강간	강도	폭행	재산범죄
0	ALABAMA	Abbeville	2645.0	11.0	1	1.0	2	7.0	63.0
1	ALABAMA	Adamsville	4481.0	19.0	1	0.0	7	11.0	321.0
2	ALABAMA	Addison	744.0	1.0	0	1.0	0	0.0	25.0

2) 행 인덱스 활용 (필요 행 선택)

DF[행 번호]

```
crime[0:2]
```

	주이름	도시이름	인구수	폭력사건	살인	강간	강도	폭행	재산범죄
0	ALABAMA	Abbeville	2645.0	11.0	1	1.0	2	7.0	63.0
1	ALABAMA	Adamsville	4481.0	19.0	1	0.0	7	11.0	321.0

DF.iloc[행 번호]

```
crime.iloc[0:2]
```

	주이름	도시이름	인구수	폭력사건	살인	강간	강도	폭행	재산범죄
0	ALABAMA	Abbeville	2645.0	11.0	1	1.0	2	7.0	63.0
1	ALABAMA	Adamsville	4481.0	19.0	1	0.0	7	11.0	321.0

DF.iloc[[행 번호, 행번호], :] - 행 번호가 일련 번호가 아닌 경우

```
crime.iloc[[0,2],:]
```

	주이름	도시이름	인구수	폭력사건	살인	강간	강도	폭행	재산범죄
0	ALABAMA	Abbeville	2645.0	11.0	1	1.0	2	7.0	63.0
2	ALABAMA	Addison	744.0	1.0	0	1.0	0	0.0	25.0

3) 열변수 조건 : 단일 조건 df[df.변수명 조건]

- 조건에 맞는 행만 저장된다. 인덱스를 보면 0, 2 행은 제외 되었음

```
crime0=crime[crime.인구수>3000]
crime0.head(3)
```

	주이름	도시이름	인구수	폭력사건	살인	강간	강도	폭행	재산범죄
1	ALABAMA	Adamsville	4481.0	19.0	1	0.0	7	11.0	321.0
3	ALABAMA	Alabaster	31170.0	44.0	0	2.0	11	31.0	640.0
4	ALABAMA	Alexander City	14692.0	119.0	2	16.0	12	89.0	661.0

- 문자열 변수 조건은 '=='을 사용해야 한다.

```
crime0=crime[crime.주이름=='OHIO']
crime0.head(3)
```

	주이름	도시이름	인구수	폭력사건	살인	강간	강도	폭행	재산범죄
5828	OHIO	Ada	5755.0	7.0	0	2.0	0	5.0	76.0
5829	OHIO	Akron	198405.0	1570.0	23	160.0	528	859.0	9649.0
5830	OHIO	Albany	877.0	0.0	0	0.0	0	0.0	22.0

4) 열변수 2개 이상 조건 df[(df.변수명 조건1) & 혹은 | (df.변수명 조건2)]

- 2개 조건을 사용하는 경우 조건은 ()으로 분리하고 and=& or= | 그리고 ==은 동일, !=은 같지 않음

```
crime00=crime[(crime.인구수>3000) & (crime.강간>2)]
crime00.head(3)
```

	주이름	도시이름	인구수	폭력사건	살인	강간	강도	폭행	재산범죄
4	ALABAMA	Alexander City	14692.0	119.0	2	16.0	12	89.0	661.0
6	ALABAMA	Andalusia	9079.0	34.0	1	4.0	6	23.0	491.0
7	ALABAMA	Anniston	22648.0	461.0	5	39.0	78	339.0	1988.0

```
crime0=crime[(crime.주이름!='OHIO') & (crime.강간>2)]
crime0.head(3)
```

	주이름	도시이름	인구수	폭력사건	살인	강간	강도	폭행	재산범죄
4	ALABAMA	Alexander City	14692.0	119.0	2	16.0	12	89.0	661.0
6	ALABAMA	Andalusia	9079.0	34.0	1	4.0	6	23.0	491.0
7	ALABAMA	Anniston	22648.0	461.0	5	39.0	78	339.0	1988.0

5) 행 인덱스 활용 (불필요 행 제거)

DF.drop[[행 번호, 행번호]] - 행 번호가 하나인 경우

```
crime.drop([0, 2], axis=0).head(2)
```

	주이름	도시이름	인구수	폭력사건	살인	강간	강도	폭행	재산범죄
1	ALABAMA	Adamsville	4481.0	19.0	1	0.0	7	11.0	321.0
3	ALABAMA	Alabaster	31170.0	44.0	0	2.0	11	31.0	640.0

- axis=0는 행을 의미, 사용하지 않아도 동일한 결과를 얻는다.

```
crime.drop([0, 2]).head(2)
```

	주이름	도시이름	인구수	폭력사건	살인	강간	강도	폭행	재산범죄
1	ALABAMA	Adamsville	4481.0	19.0	1	0.0	7	11.0	321.0
3	ALABAMA	Alabaster	31170.0	44.0	0	2.0	11	31.0	640.0

열 단위 변형

열 (변수) subset

1) 필요 변수 선택

```
df[['변수명1', '변수명2', ...]]
```

```
df.iloc[:, [1, 3, 4]] - 열 인덱스 활용
```

```
crime[['주이름', '인구수', '절도']].head(3)
```

	주이름	인구수	절도
0	ALABAMA	2645.0	39.0
1	ALABAMA	4481.0	252.0
2	ALABAMA	744.0	17.0

```
crime.iloc[:, [0,2,10]].head(3)
```

	주이름	인구수	절도
0	ALABAMA	2645.0	39.0
1	ALABAMA	4481.0	252.0
2	ALABAMA	744.0	17.0

```
df.loc[:, ['변수명1', '변수명2', ...]]
```

```
crime.loc[:, ['주이름', '인구수', '절도']].head(3)
```

	주이름	인구수	절도
0	ALABAMA	2645.0	39.0
1	ALABAMA	4481.0	252.0
2	ALABAMA	744.0	17.0

2) 불필요 변수 제거

```
df.drop(['변수명1', '변수명2', ...], axis=1)
```

```
crime.drop(['살인', '강간'], axis=1).head(2)
```

	주이름	도시이름	인구수	폭력사건	강도	폭행	재산범죄	주거침입
0	ALABAMA	Abbeville	2645.0	11.0	2	7.0	63.0	21.0
1	ALABAMA	Adamsville	4481.0	19.0	7	11.0	321.0	58.0

행, 열 동시 subset

```
df.iloc[[번호들], [번호들]]
```

```
crime.iloc[[0,3],[0,2]]
```

	주이름	인구수
0	ALABAMA	2645.0
3	ALABAMA	31170.0

만약 인덱스 번호와 이름(label)를 혼용하여 사용한다면 .iloc대신 .ix 사용

```
crime.iloc[[0,3], ['주이름', '인구수']]
```

```
TypeError                                Traceback (most recent call last)
<ipython-input-90-b78b0a21296c> in <module>
----> 1 crime.iloc[[0,3], ['주이름', '인구수']]
```

```
crime.ix[[0,3], ['주이름', '인구수']]
```

	주이름	인구수
0	ALABAMA	2645.0
3	ALABAMA	31170.0

데이터 변형

데이터 쌓기 (melt)

http://203.247.53.31/Stat_Notes/elem_stat/Stat_methods/Fastfoods_Drivethrough.csv 예제 데이터

- 5개 패스트푸드 점 드라이브스루 오더 후 서빙까지 걸린 시간(초)

```
import pandas as pd
url='http://203.247.53.31/Stat_Notes/elem_stat/Stat_methods/Fastfoods_Drivethrough.csv'
fastfood=pd.read_csv(url,encoding='ms949')
```

fastfood.head(3)

	파파이치킨	웬디스	맥도날드	버거킹	잭인더박스
0	150	173	198	158	289
1	197	95	131	195	273
2	186	154	158	177	254

```
pd.melt(fastfood, id_vars=['쌓는기준'], value_vars=['쌓는 변수명1','변수명2'], var_name='id변수명', value_name='쌓은변수명')
```

```
fastfood2=pd.melt(fastfood, value_vars=['웬디스','버거킹'],
var_name='지점명', value_name='시간(초)')
```

fastfood2.head(2)

	지점명	시간(초)
0	웬디스	173
1	웬디스	95

fastfood2.tail(2)

	지점명	시간(초)
198	버거킹	154
199	버거킹	202

-> 쌓는 기준이 없는 경우

쌓는 기준이 있는 경우 id_vars=[] 사용 가능

예제 : http://203.247.53.31/Stat_Notes/example_data/US_crime.csv

```
import pandas as pd
url='http://203.247.53.31/Stat_Notes/example_data/US_crime.csv'
crime=pd.read_csv(url, encoding='ms949')
```

crime.columns

Index(['주이름', '도시이름', '인구수', '폭력사건', '살인', '강간', '강도', '폭행', '재산범죄', '주거침입', '절도', '차량절도'], dtype='object')

crime.head(3)

	주이름	도시이름	인구수	폭력사건	살인	강간	강도	폭행	재산범죄
0	ALABAMA	Abbeville	2645.0	11.0	1	1.0	2	7.0	6
1	ALABAMA	Adamsville	4481.0	19.0	1	0.0	7	11.0	32
2	ALABAMA	Addison	744.0	1.0	0	1.0	0	0.0	2

```
crime0=pd.melt(crime, id_vars=['도시이름'], value_vars=['강도','폭행'],
               var_name='사건내용',value_name='사건건수')
```

```
crime0.head(3)
```

	도시이름	사건내용	사건건수
0	Abbeville	강도	2.0
1	Adamsville	강도	7.0
2	Addison	강도	0.0

```
crime0.tail(3)
```

	도시이름	사건내용	사건건수
18581	Torrington	폭행	17.0
18582	Wheatland	폭행	11.0
18583	Worland	폭행	7.0

데이터 열 나누기 (pivot) - melt

```
df.pivot_table(index='인덱스 변수', columns='범주형', values='측정형')
```

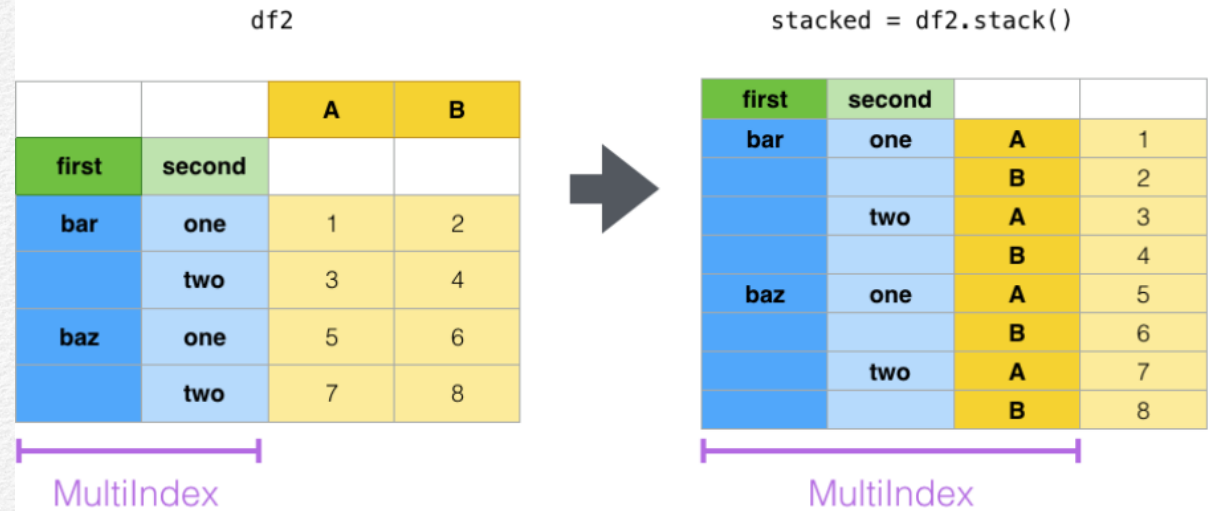
- melt에 의해 합쳐진 데이터 다시 나누기

```
crime0.pivot_table(index='도시이름',columns='사건내용',values='사건건수')
```

	사건내용	강도	폭행
도시이름			
Abbeville		4.250000	35.500000
Aberdeen		7.833333	22.166667
Aberdeen Township		13.000000	6.000000

데이터 stack(쌓기) / unstack(나누기)

Stack



Unstack

