데이터 전처리 (3) - 데이터 재구조화

- 피벗 테이블
- melt
- stack, unstack
- 교차표

#01. 패키지 참조

```
from pandas import DataFrame, read_excel
from pandas import pivot_table, crosstab, melt
```

#02. 피벗 테이블

1. 샘플 데이터 가져오기

```
df = read_excel("https://data.hossam.kr/C02/city_people.xlsx")
df
```

	도시	연도	인구	지역
0	서울	2015	9904312	수도권
1	서울	2010	9631482	수도권

	도시	연도	인구	지역
2	서울	2005	9762546	수도권
3	부산	2015	3448737	경상권
4	부산	2010	3393191	경상권
5	부산	2005	3512547	경상권
6	인천	2015	2890451	수도권
7	인천	2010	2632035	수도권

2. 피벗테이블 기본

인덱스, 컬럼, 값으로 사용할 필드를 각각 지정하여 데이터를 재배치

```
pivot_table(df,                 # 피벗할 데이터프레임
            index = '도시',     # 행 위치에 들어갈 열
            columns = '연도',   # 열 위치에 들어갈 열
            values = '인구'     # 데이터로 사용할 열
)
```

연도	2005	2010	2015
도시			
부산	3512547.0	3393191.0	3448737.0
서울	9762546.0	9631482.0	9904312.0

연도	2005	2010	2015
도시			
인천	NaN	2632035.0	2890451.0

3. 중복 데이터의 집계 방법 지정하기

```
a = pivot_table(df,                 # 피벗할 데이터프레임
                index = '지역',    # 행 위치에 들어갈 열
                columns = '연도', # 열 위치에 들어갈 열
                values = '인구',    # 데이터로 사용할 열
                aggfunc='mean'    # 데이터가 두 개 이상일 경우 집계함수 지정
)

a
```

연도	2005	2010	2015
지역			
경상권	3512547.0	3393191.0	3448737.0
수도권	9762546.0	6131758.5	6397381.5

4. 복수 집계 함수 지정

```
pivot_table(df,
                index = '지역',
```

```
            columns = '연도',
            values = '인구',
            aggfunc=['sum', 'mean']
)
```

	sum			mean		
연도	2005	2010	2015	2005	2010	2015
지역						
경상권	3512547	3393191	3448737	3512547.0	3393191.0	3448737.0
수도권	9762546	12263517	12794763	9762546.0	6131758.5	6397381.5

5. 복수 인덱스 지정

```
pivot_table(df,                          # 피벗할 데이터프레임
            index = ['지역', '연도'],    # 행 위치에 들어갈 열
            columns = '도시',            # 열 위치에 들어갈 열
            values = '인구',             # 데이터로 사용할 열
            aggfunc =  ['mean', 'sum']  # 데이터 집계함수
)
```

		mean			sum		
	도시	부산	서울	인천	부산	서울	인천
지역	연도						
경상권	2005	3512547.0	NaN	NaN	3512547.0	NaN	NaN
	2010	3393191.0	NaN	NaN	3393191.0	NaN	NaN
	2015	3448737.0	NaN	NaN	3448737.0	NaN	NaN
수도권	2005	NaN	9762546.0	NaN	NaN	9762546.0	NaN
	2010	NaN	9631482.0	2632035.0	NaN	9631482.0	2632035.0
	2015	NaN	9904312.0	2890451.0	NaN	9904312.0	2890451.0

#03. melt

데이터 테이블의 컬럼 이름을 변수화 한 형태

피벗테이블을 분리한 것으로 볼 수 있다.

샘플 피벗 테이블

```
pivot_df = pivot_table(df,                  # 피벗할 데이터프레임
            index = '연도',    # 행 위치에 들어갈 열
            columns = '지역', # 열 위치에 들어갈 열
            values = '인구',    # 데이터로 사용할 열
            aggfunc='mean'    # 데이터가 두 개 이상일 경우 집계함수 지정
```

```
)

pivot_df
```

지역	경상권	수도권
연도		
2005	3512547.0	9762546.0
2010	3393191.0	6131758.5
2015	3448737.0	6397381.5

2. 피벗 테이블 분리

- id_vars : 인덱스로 사용할 컬럼이름. 반드시 컬럼만 가능(인덱스 불가)
- value_vars: 분리할 컬럼 이름들

```
# 데이터프레임의 인덱스를 일반 컬럼으로 설정
pivot_df2 = pivot_df.reset_index()
pivot_df2
```

	지역	연도	경상권	수도권
0	2005	3512547.0	9762546.0
1	2010	3393191.0	6131758.5
2	2015	3448737.0	6397381.5

```
# 피벗 테이블 분리
mdf = melt(pivot_df2, id_vars=['연도'], value_vars=['경상권', '수도권'])
mdf
```

	연도	지역	value
0	2005	경상권	3512547.0
1	2010	경상권	3393191.0
2	2015	경상권	3448737.0
3	2005	수도권	9762546.0
4	2010	수도권	6131758.5
5	2015	수도권	6397381.5

```
# 피벗 테이블 분리 및 필드 이름 지정
mdf = melt(pivot_df2, id_vars=['연도'], value_vars=['경상권', '수도권'],
           var_name='구분', value_name='인구수')
mdf
```

	연도	구분	인구수
0	2005	경상권	3512547.0
1	2010	경상권	3393191.0
2	2015	경상권	3448737.0

	연도	구분	인구수
3	2005	수도권	9762546.0
4	2010	수도권	6131758.5
5	2015	수도권	6397381.5

#04. stack, unstack

데이터 분리 (stack)

모든 변수를 하나의 변수로 쌓아놓는 처리

샘플 데이터 가져오기

```
df = read_excel("https://data.hossam.kr/C02/body.xlsx", index_col="name"
df
```

	sex	height	weight
name			
Lee	M	175	98.0
Park	F	167	48.0
Hong	M	180	NaN
Kim	F	162	55.0

	sex	height	weight
name			
Nam	M	172	85.0

stack 처리

리턴 결과를 Series 객체가 된다.

```
st = df.stack()
st
```

```
name
Lee   sex          M
      height     175
      weight    98.0
Park  sex          F
      height     167
      weight    48.0
Hong  sex          M
      height     180
Kim   sex          F
      height     162
      weight    55.0
Nam   sex          M
      height     172
      weight    85.0
dtype: object
```

stack 결과를 DataFrame으로 만들기

```
df2 = DataFrame(st)
df2
```

		0
name		
Lee	sex	M
	height	175
	weight	98.0
Park	sex	F
	height	167
	weight	48.0
Hong	sex	M
	height	180
Kim	sex	F
	height	162
	weight	55.0
Nam	sex	M
	height	172

		0
name		
	weight	85.0

빈 값에 대한 처리

`dropna=False` 파라미터를 적용하면 빈값(`NaN`)을 유지한다. (기본값=True, 빈값 삭제)

```
st2 = df.stack(dropna=False)
st2
```

```
name
Lee   sex          M
      height     175
      weight    98.0
Park  sex          F
      height     167
      weight    48.0
Hong  sex          M
      height     180
      weight     NaN
Kim   sex          F
      height     162
      weight    55.0
Nam   sex          M
      height     172
```

```
      weight    85.0
dtype: object
```

stack 결과를 원래대로 되돌림

```
st.unstack()
```

	sex	height	weight
name			
Lee	M	175	98.0
Park	F	167	48.0
Hong	M	180	NaN
Kim	F	162	55.0
Nam	M	172	85.0

unstack을 수행하면서 빈값을 다른 값으로 대체

```
st.unstack(fill_value=100)
```

	sex	height	weight
name			
Lee	M	175	98.0

	sex	height	weight
name			
Park	F	167	48.0
Hong	M	180	100
Kim	F	162	55.0
Nam	M	172	85.0

#04. 교차표(crosstab)

범주형 자료를 갖는 데이터에 대해 각 범주별로 빈도수를 계산하여 표현한 표

1. 샘플 데이터 가져오기

```
df = read_excel("https://data.hossam.kr/C02/score.xlsx")
df
```

	gender	score
0	M	A
1	M	C
2	M	B
3	M	B
4	W	A

	gender	score
5	W	C
6	W	C
7	W	B

2. 교차표 만들기

index 파라미터와 columns 파라미터를 지정한다.

```
crosstab(index=df['gender'], columns=df['score'])
```

score	A	B	C
gender			
M	1	2	1
W	1	1	2

3. 파라미터 설정

- rownames : 인덱스의 이름 설정
- colnames : 컬럼의 이름 설정
- margins : 집계 결과 포함 여부(True/False)

```
crosstab(index=df['gender'], columns=df['score'],
```

```
        rownames=['성별'], colnames=['점수'], margins=True)
```

점수	A	B	C	All
성별				
M	1	2	1	4
W	1	1	2	4
All	2	3	3	8

4. 비율 표시

- `normalize=True` 파라미터 사용

```
crosstab(index=df['gender'], columns=df['score'],
         rownames=['성별'], colnames=['점수'], margins=True,
         normalize=True)
```

점수	A	B	C	All
성별				
M	0.125	0.250	0.125	0.5
W	0.125	0.125	0.250	0.5
All	0.250	0.375	0.375	1.0