

## 7장 깔끔한 데이터

## **07-1 열과 피벗**



데이터의 열 이름이 어떤 값을 의미하면 열의 폭의 넓은 경우가 많음

|   | religion            | <\$10k | \$10-20k | \$20-30k | \$30-40k | \$40-50k | \$50-75k | # |
|---|---------------------|--------|----------|----------|----------|----------|----------|---|
| 0 | Agnostic            | 27     | 34       | 60       | 81       | 76       | 137      |   |
| 1 | Atheist             | 12     | 27       | 37       | 52       | 35       | 70       |   |
| 2 | Buddhist            | 27     | 21       | 30       | 34       | 33       | 58       |   |
| 3 | Catholic            | 418    | 617      | 732      | 670      | 638      | 1116     |   |
| 4 | Don' t know/refused | 15     | 14       | 15       | 11       | 10       | 35       |   |

|   | \$75-100k | \$100-150k | >150k | Don't know/refused |
|---|-----------|------------|-------|--------------------|
| 0 | 122       | 109        | 84    | 96                 |
| 1 | 73        | 59         | 74    | 76                 |
| 2 | 62        | 39         | 53    | 54                 |
| 3 | 949       | 792        | 633   | 1489               |
| 4 | 21        | 17         | 18    | 116                |



## 데이터 재구조화 (Reshaping data by melt)



`pd.melt(data, id_vars, var_name, value_name)`

*Original data*

| cust_ID | prd_CD | pch_amt | pch_cnt |
|---------|--------|---------|---------|
| C_001   | P_001  | 100     | 1       |
| C_001   | P_002  | 200     | 2       |
| C_002   | P_001  | 300     | 3       |
| C_002   | P_002  | 400     | 4       |

*melt*

*Melted data*

| cust_ID | prd_CD | variable | value |
|---------|--------|----------|-------|
| C_001   | P_001  | pch_amt  | 100   |
| C_001   | P_002  | pch_amt  | 200   |
| C_002   | P_001  | pch_amt  | 300   |
| C_002   | P_002  | pch_amt  | 400   |
| C_001   | P_001  | pch_cnt  | 1     |
| C_001   | P_002  | pch_cnt  | 2     |
| C_002   | P_001  | pch_cnt  | 3     |
| C_002   | P_002  | pch_cnt  | 4     |

# melt 메서드(1)



## 1개의 열 고정하고 나머지 열을 행으로 바꾸기(Pivot)

```
import pandas as pd
pew = pd.read_csv('../data/pew.csv')
print(pew.head( ))
```

|   | religion           | <\$10k | \$10-20k | \$20-30k | \$30-40k | \$40-50k | \$50-75k | \ |
|---|--------------------|--------|----------|----------|----------|----------|----------|---|
| 0 | Agnostic           | 27     | 34       | 60       | 81       | 76       | 137      |   |
| 1 | Atheist            | 12     | 27       | 37       | 52       | 35       | 70       |   |
| 2 | Buddhist           | 27     | 21       | 30       | 34       | 33       | 58       |   |
| 3 | Catholic           | 418    | 617      | 732      | 670      | 638      | 1116     |   |
| 4 | Don't know/refused | 15     | 14       | 15       | 11       | 10       | 35       |   |

```
pew_long = pd.melt(pew, id_vars='religion')
print(pew_long.head( ))
```

|   | religion           | variable | value |
|---|--------------------|----------|-------|
| 0 | Agnostic           | <\$10k   | 27    |
| 1 | Atheist            | <\$10k   | 12    |
| 2 | Buddhist           | <\$10k   | 27    |
| 3 | Catholic           | <\$10k   | 418   |
| 4 | Don't know/refused | <\$10k   | 15    |

# melt 메서드(2)



2개 이상의 열을 고정하고 나머지 열을 행으로 바꾸기  
year, artist, track, time, date.entered 컬럼을 제외한 부분을 행으로 바꾸기

```
billboard = pd.read_csv('../data/billboard.csv')  
print billboard.iloc[0:5, 0:16]
```

|   | year | artist       | track                   | time | date.entered | wk1 | wk2  | ₩ |
|---|------|--------------|-------------------------|------|--------------|-----|------|---|
| 0 | 2000 | 2 Pac        | Baby Don't Cry (Keep... | 4:22 | 2000-02-26   | 87  | 82.0 |   |
| 1 | 2000 | 2Ge+her      | The Hardest Part Of ... | 3:15 | 2000-09-02   | 91  | 87.0 |   |
| 2 | 2000 | 3 Doors Down | Kryptonite              | 3:53 | 2000-04-08   | 81  | 70.0 |   |
| 3 | 2000 | 3 Doors Down | Loser                   | 4:24 | 2000-10-21   | 76  | 76.0 |   |
| 4 | 2000 | 504 Boyz     | Wobble Wobble           | 3:35 | 2000-04-15   | 57  | 34.0 |   |

|   | wk3  | wk4  | wk5  | wk6  | wk7  | wk8  | wk9  | wk10 | wk11 |
|---|------|------|------|------|------|------|------|------|------|
| 0 | 72.0 | 77.0 | 87.0 | 94.0 | 99.0 | NaN  | NaN  | NaN  | NaN  |
| 1 | 92.0 | NaN  | NaN  | NaN  | NaN  | NaN  | NaN  | NaN  | NaN  |
| 2 | 68.0 | 67.0 | 66.0 | 57.0 | 54.0 | 53.0 | 51.0 | 51.0 | 51.0 |
| 3 | 72.0 | 69.0 | 67.0 | 65.0 | 55.0 | 59.0 | 62.0 | 61.0 | 61.0 |
| 4 | 25.0 | 17.0 | 17.0 | 31.0 | 36.0 | 49.0 | 53.0 | 57.0 | 64.0 |

# melt 메서드(2)



`id_vars` 인자에는 고정할 column들을 넣는다.

`var_name` 인자에는 column들의 이름들이 들어갈 새 column의 이름을 넣는다.

`value_name` 인자에는 원래 value였던 자료들이 들어갈 새 column의 이름을 넣는다.

고정 column

```
billboard_long = pd.melt(billboard, id_vars=['year', 'artist', 'track', 'time', 'date.entered'],  
                        var_name='week', value_name='rating')  
print(billboard_long.head())
```

|   | year | artist       | track                   | time | date.entered | week | rating |
|---|------|--------------|-------------------------|------|--------------|------|--------|
| 0 | 2000 | 2 Pac        | Baby Don't Cry (Keep... | 4:22 | 2000-02-26   | wk1  | 87.0   |
| 1 | 2000 | 2Ge+her      | The Hardest Part Of ... | 3:15 | 2000-09-02   | wk1  | 91.0   |
| 2 | 2000 | 3 Doors Down | Kryptonite              | 3:53 | 2000-04-08   | wk1  | 81.0   |
| 3 | 2000 | 3 Doors Down | Loser                   | 4:24 | 2000-10-21   | wk1  | 76.0   |
| 4 | 2000 | 504 Boyz     | Wobble Wobble           | 3:35 | 2000-04-15   | wk1  | 57.0   |

## **07-2 열 이름 관리하기**



# ebola 데이터 집합 살펴보기



Cases : 발병 / Deaths : 죽음

```
ebola = pd.read_csv('../data/country_timeseries.csv')  
print(ebola.columns)
```

```
Index(['Date', 'Day', 'Cases_Guinea', 'Cases_Liberia', 'Cases_SierraLeone',  
      'Cases_Nigeria', 'Cases_Senegal', 'Cases_UnitedStates', 'Cases_Spain',  
      'Cases_Mali', 'Deaths_Guinea', 'Deaths_Liberia', 'Deaths_SierraLeone',  
      'Deaths_Nigeria', 'Deaths_Senegal', 'Deaths_UnitedStates',  
      'Deaths_Spain', 'Deaths_Mali'],  
      dtype='object')
```

# ebola 데이터 집합 살펴보기



‘Date’ 와 ‘Day’ Column을 제외하고 녹여버리기~!

```
ebola_long = pd.melt(ebola, id_vars=['Date', 'Day'])  
print(ebola_long.head())
```

variable이라는 하나의 열이 여러 의미를 가지게 되었다!!!

|   | Date       | Day | <u>variable</u> | <u>value</u> |
|---|------------|-----|-----------------|--------------|
| 0 | 1/5/2015   | 289 | Cases_Guinea    | 2776.0       |
| 1 | 1/4/2015   | 288 | Cases_Guinea    | 2775.0       |
| 2 | 1/3/2015   | 287 | Cases_Guinea    | 2769.0       |
| 3 | 1/2/2015   | 286 | Cases_Guinea    | NaN          |
| 4 | 12/31/2014 | 284 | Cases_Guinea    | 2730.0       |

# split 메서드로 열 이름 분리하기



## < variable\_split : 자료형은 Series >

|      |                 |
|------|-----------------|
| 0    | [Cases, Guinea] |
| 1    | [Cases, Guinea] |
| 2    | [Cases, Guinea] |
| 3    | [Cases, Guinea] |
| 4    | [Cases, Guinea] |
| ...  | ...             |
| 1947 | [Deaths, Mali]  |
| 1948 | [Deaths, Mali]  |
| 1949 | [Deaths, Mali]  |
| 1950 | [Deaths, Mali]  |
| 1951 | [Deaths, Mali]  |

```
status_values = variable_split.str.get(0)
country_values = variable_split.str.get(1)
print(status_values[:5])
print(country_values[:5])
```

```
0    Cases
1    Cases
2    Cases
3    Cases
4    Cases
Name: variable, dtype: object
0    Guinea
1    Guinea
2    Guinea
3    Guinea
4    Guinea
Name: variable, dtype: object
```

# split 메서드로 열 이름 분리하기



각 component의 위치에 해당하는 원소를 추출한다.

component : lists, tuples or strings

```
pandas.Series.str.get
```

```
Series.str.get(self, i) ¶
```

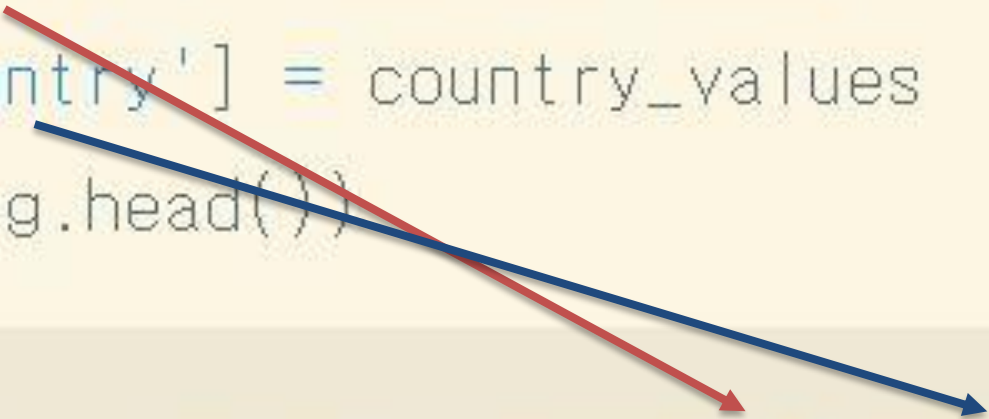
Extract element from each component at specified position.

Extract element from lists, tuples, or strings in each element in the Series/Index.

# 정돈된 열을 다시 DF에 추가하기



```
ebola_long['status'] = status_values  
ebola_long['country'] = country_values  
print(ebola_long.head())
```

A diagram with two arrows. A red arrow originates from the 'status' column in the code above and points to the 'status' column in the table below. A blue arrow originates from the 'country' column in the code above and points to the 'country' column in the table below.

|   | Date       | Day | variable     | value  | status | country |
|---|------------|-----|--------------|--------|--------|---------|
| 0 | 1/5/2015   | 289 | Cases_Guinea | 2776.0 | Cases  | Guinea  |
| 1 | 1/4/2015   | 288 | Cases_Guinea | 2775.0 | Cases  | Guinea  |
| 2 | 1/3/2015   | 287 | Cases_Guinea | 2769.0 | Cases  | Guinea  |
| 3 | 1/2/2015   | 286 | Cases_Guinea | NaN    | Cases  | Guinea  |
| 4 | 12/31/2014 | 284 | Cases_Guinea | 2730.0 | Cases  | Guinea  |



# #다른 방법



```
variable_split = ebola_long.variable.str.split('_', expand=True)
print(variable_split.head())
variable_split.columns = ['status', 'country'] #column 추가
ebola_parsed = pd.concat([ebola_long, variable_split], axis=1)
print(ebola_parsed.head())
```

열 방향 추가!!

|   | 0     | 1      |
|---|-------|--------|
| 0 | Cases | Guinea |
| 1 | Cases | Guinea |
| 2 | Cases | Guinea |
| 3 | Cases | Guinea |
| 4 | Cases | Guinea |

|   | Date       | Day | variable     | value  | status | country | status | country |
|---|------------|-----|--------------|--------|--------|---------|--------|---------|
| 0 | 1/5/2015   | 289 | Cases_Guinea | 2776.0 | Cases  | Guinea  | Cases  | Guinea  |
| 1 | 1/4/2015   | 288 | Cases_Guinea | 2775.0 | Cases  | Guinea  | Cases  | Guinea  |
| 2 | 1/3/2015   | 287 | Cases_Guinea | 2769.0 | Cases  | Guinea  | Cases  | Guinea  |
| 3 | 1/2/2015   | 286 | Cases_Guinea | NaN    | Cases  | Guinea  | Cases  | Guinea  |
| 4 | 12/31/2014 | 284 | Cases_Guinea | 2720.0 | Cases  | Guinea  | Cases  | Guinea  |

## **07-3 여러 열을 하나로 정리하기**

# 기상 데이터 집합 살펴보기



```
weather = pd.read_csv('../data/weather.csv')
print(weather.iloc[:5, :11])
```

|   | id      | year | month | element | d1  | d2   | d3   | d4  | d5   | d6  | d7  |
|---|---------|------|-------|---------|-----|------|------|-----|------|-----|-----|
| 0 | MX17004 | 2010 | 1     | tmax    | NaN | NaN  | NaN  | NaN | NaN  | NaN | NaN |
| 1 | MX17004 | 2010 | 1     | tmin    | NaN | NaN  | NaN  | NaN | NaN  | NaN | NaN |
| 2 | MX17004 | 2010 | 2     | tmax    | NaN | 27.3 | 24.1 | NaN | NaN  | NaN | NaN |
| 3 | MX17004 | 2010 | 2     | tmin    | NaN | 14.4 | 14.4 | NaN | NaN  | NaN | NaN |
| 4 | MX17004 | 2010 | 3     | tmax    | NaN | NaN  | NaN  | NaN | 32.1 | NaN | NaN |

날짜 열에 각 월별  
최고, 최저 온도 데이터 저장

```
weather_melt = pd.melt(weather, id_vars=['id', 'year', 'month', 'element'],
                        var_name='day', value_name='temp')
print(weather_melt.head())
```

|   | id      | year | month | element | day | temp |
|---|---------|------|-------|---------|-----|------|
| 0 | MX17004 | 2010 | 1     | tmax    | d1  | NaN  |
| 1 | MX17004 | 2010 | 1     | tmin    | d1  | NaN  |
| 2 | MX17004 | 2010 | 2     | tmax    | d1  | NaN  |
| 3 | MX17004 | 2010 | 2     | tmin    | d1  | NaN  |



# 기상 데이터의 여러 열을 하나로 정리하기



## pivot\_table 메소드

```
weather_tidy = weather_melt.pivot_table(  
    index=['id', 'year', 'month', 'day'],  
    columns='element',  
    values='temp'  
)  
  
weather_tidy
```

| id      | year | month | day | element | tmax | tmin |
|---------|------|-------|-----|---------|------|------|
|         |      |       |     |         |      |      |
| MX17004 | 2010 | 1     | d30 |         | 27.8 | 14.5 |
|         |      |       | 2   | d11     | 29.7 | 13.4 |
|         |      |       |     | d2      | 27.3 | 14.4 |
|         |      |       |     | d23     | 29.9 | 10.7 |
|         |      | 3     |     | d3      | 24.1 | 14.4 |
|         |      |       | d10 |         | 34.5 | 16.8 |
|         |      |       |     | d16     | 31.1 | 17.6 |
|         |      |       |     | d5      | 32.1 | 14.2 |
|         |      | 4     | d27 |         | 36.3 | 16.7 |
|         |      | 5     | d27 |         | 33.2 | 18.2 |
|         |      | 6     | d17 |         | 28.0 | 17.5 |
|         |      |       |     | d29     | 30.1 | 18.0 |
|         |      | 7     |     | d3      | 28.6 | 17.5 |
|         |      |       |     | d14     | 29.9 | 16.5 |
|         |      | 8     |     | d23     | 26.4 | 15.0 |
|         |      |       |     | d5      | 29.6 | 15.8 |
|         |      |       |     | d29     | 28.0 | 15.3 |

# 기상 데이터의 여러 열을 하나로 정리하기



## reset\_index 메소드

```
1 weather_tidy_flat = weather_tidy.reset_index()  
2 print(weather_tidy_flat.head())
```

| element | id      | year | month | day | tmax | tmin |
|---------|---------|------|-------|-----|------|------|
| 0       | MX17004 | 2010 | 1     | d30 | 27.8 | 14.5 |
| 1       | MX17004 | 2010 | 2     | d11 | 29.7 | 13.4 |
| 2       | MX17004 | 2010 | 2     | d2  | 27.3 | 14.4 |
| 3       | MX17004 | 2010 | 2     | d23 | 29.9 | 10.7 |
| 4       | MX17004 | 2010 | 2     | d3  | 24.1 | 14.4 |

## **07-4 중복 데이터 처리하기**

# 빌보드 차트 데이터 집합 살펴보기



```
1 import pandas as pd
2
3 billboard = pd.read_csv('../data/billboard.csv')
4 billboard_long = pd.melt(billboard, id_vars=['year', 'artist', 'track', 'time', 'date.entered'],
5                           var_name='week', value_name='rating')
6
7 print(billboard_long.shape)
8 billboard_long.head(10)
```

(24092, 7)

|   | year | artist       | track                   | time | date.entered | week | rating |
|---|------|--------------|-------------------------|------|--------------|------|--------|
| 0 | 2000 | 2 Pac        | Baby Don't Cry (Keep... | 4:22 | 2000-02-26   | wk1  | 87.0   |
| 1 | 2000 | 2Ge+her      | The Hardest Part Of ... | 3:15 | 2000-09-02   | wk1  | 91.0   |
| 2 | 2000 | 3 Doors Down | Kryptonite              | 3:53 | 2000-04-08   | wk1  | 81.0   |
| 3 | 2000 | 3 Doors Down | Loser                   | 4:24 | 2000-10-21   | wk1  | 76.0   |
| 4 | 2000 | 504 Boyz     | Wobble Wobble           | 3:35 | 2000-04-15   | wk1  | 57.0   |
| 5 | 2000 | 98^0         | Give Me Just One Nig... | 3:24 | 2000-08-19   | wk1  | 51.0   |
| 6 | 2000 | A*Teens      | Dancing Queen           | 3:44 | 2000-07-08   | wk1  | 97.0   |
| 7 | 2000 | Aaliyah      | I Don't Wanna           | 4:15 | 2000-01-29   | wk1  | 84.0   |
| 8 | 2000 | Aalivah      | Trv Again               | 4:03 | 2000-03-18   | wk1  | 59.0   |

# 빌보드 차트 데이터 집합 살펴보기



# year, artist, track, time에 중복이 많다는 것을 알 수 있다.

```
1 billboard_long[billboard_long.track == 'Loser'].head()
```

|      | year | artist       | track | time | date.entered | week | rating |
|------|------|--------------|-------|------|--------------|------|--------|
| 3    | 2000 | 3 Doors Down | Loser | 4:24 | 2000-10-21   | wk1  | 76.0   |
| 320  | 2000 | 3 Doors Down | Loser | 4:24 | 2000-10-21   | wk2  | 76.0   |
| 637  | 2000 | 3 Doors Down | Loser | 4:24 | 2000-10-21   | wk3  | 72.0   |
| 954  | 2000 | 3 Doors Down | Loser | 4:24 | 2000-10-21   | wk4  | 69.0   |
| 1271 | 2000 | 3 Doors Down | Loser | 4:24 | 2000-10-21   | wk5  | 67.0   |

# 빌보드 차트 – 중복 데이터 처리하기



중복이 많은 year, artist, track, time 열을 추출한다.

```
1 billboard_songs = billboard_long[['year', 'artist', 'track', 'time']]
2 print(billboard_songs.shape)
```

(24092, 4)

**중복된 행을 제거!**

```
1 billboard_songs = billboard_songs.drop_duplicates()
2 print(billboard_songs[billboard_songs.track == 'Loser'].head())
3 print(billboard_songs.shape)
```

|   | year |   | artist     | track | time |
|---|------|---|------------|-------|------|
| 3 | 2000 | 3 | Doors Down | Loser | 4:24 |

(317, 4)

# 빌보드 차트 – 중복 데이터 처리하기



‘id’라는 새로운 column을 만들!

```
1 billboard_songs['id'] = range(len(billboard_songs))  
2 print(billboard_songs.head(n=10))
```

|   | year | artist         | track                   | time | id |
|---|------|----------------|-------------------------|------|----|
| 0 | 2000 | 2 Pac          | Baby Don't Cry (Keep... | 4:22 | 0  |
| 1 | 2000 | 2Gether        | The Hardest Part Of ... | 3:15 | 1  |
| 2 | 2000 | 3 Doors Down   | Kryptonite              | 3:53 | 2  |
| 3 | 2000 | 3 Doors Down   | Loser                   | 4:24 | 3  |
| 4 | 2000 | 504 Boyz       | Wobble Wobble           | 3:35 | 4  |
| 5 | 2000 | 98^0           | Give Me Just One Nig... | 3:24 | 5  |
| 6 | 2000 | A*Teens        | Dancing Queen           | 3:44 | 6  |
| 7 | 2000 | Aaliyah        | I Don't Wanna           | 4:15 | 7  |
| 8 | 2000 | Aaliyah        | Try Again               | 4:03 | 8  |
| 9 | 2000 | Adams, Yolanda | Open My Heart           | 5:30 | 9  |



# 빌보드 차트 - 중복 데이터 처리하기



```
1 print billboard_ratings.shape
2
3 billboard_ratings = billboard_long.merge( billboard_songs,
4                                           on=['year', 'artist', 'track', 'time'])
5
```

billboard\_long에 billboard\_songs를 merge한다.

↳ 인자 on을 기준으로 merge한다.

(24092, 8)

```
1 print billboard_ratings.shape
2 billboard_ratings.head(20)
```

(24092, 8)

|   | year | artist | track                   | time | date.entered | week | rating | id |
|---|------|--------|-------------------------|------|--------------|------|--------|----|
| 0 | 2000 | 2 Pac  | Baby Don't Cry (Keep... | 4:22 | 2000-02-26   | wk1  | 87.0   | 0  |
| 1 | 2000 | 2 Pac  | Baby Don't Cry (Keep... | 4:22 | 2000-02-26   | wk2  | 82.0   | 0  |
| 2 | 2000 | 2 Pac  | Baby Don't Cry (Keep... | 4:22 | 2000-02-26   | wk3  | 72.0   | 0  |
| 3 | 2000 | 2 Pac  | Baby Don't Cry (Keep... | 4:22 | 2000-02-26   | wk4  | 77.0   | 0  |
| 4 | 2000 | 2 Pac  | Baby Don't Cry (Keep... | 4:22 | 2000-02-26   | wk5  | 87.0   | 0  |