

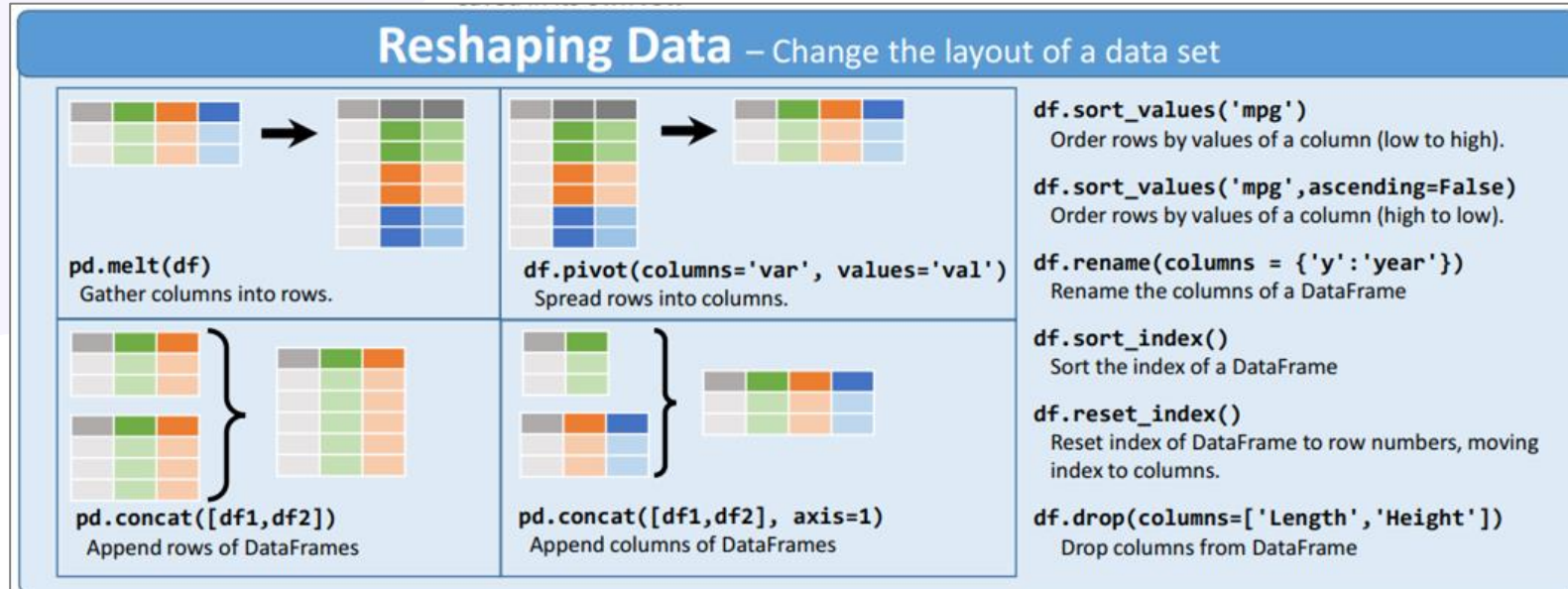


데이터 재형성



Reshaping Data

- 데이터 재형성



<출처 : <http://pandas.pydata.org>>

Reshaping Data

- 데이터 재형성

01 `sort_values`

- 특정 열의 데이터를 기준으로 데이터프레임을 정렬함

02 `df.rename`

- 데이터프레임 내의 행, 열의 이름을 변경함

03 `df.sort_index`

- Index 별로 정렬함

04 `df.reset_index`

- Index가 없을 때, Index를 생성해서 정렬함

Reshaping Data

- 데이터 재형성

06 df.drop

- 특정 컬럼(Column)의 변수를 데이터에서 제거함

```
df.sort_values('mpg')
```

Order rows by values of a column (low to high).

```
df.sort_values('mpg', ascending=False)
```

Order rows by values of a column (high to low).

```
df.rename(columns = {'y': 'year'})
```

Rename the columns of a DataFrame

```
df.sort_index()
```

Sort the index of a DataFrame

```
df.reset_index()
```

Reset index of DataFrame to row numbers, moving index to columns.

```
df.drop(columns=['Length', 'Height'])
```

Drop columns from DataFrame

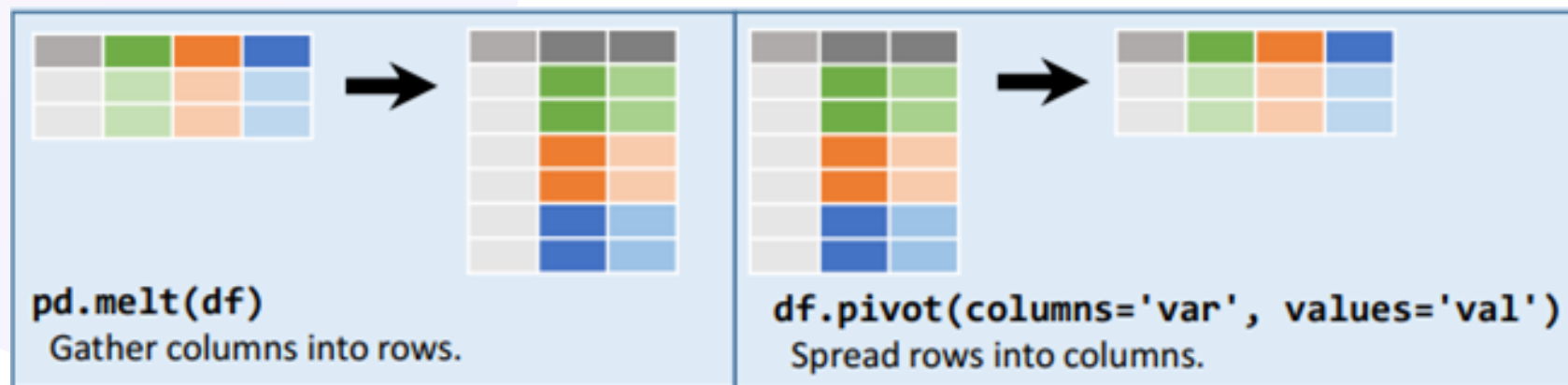


Reshaping Data

이론 영상 후 실습 영상 제시



깔끔한 데이터 만들기(Melt, Pivot)



Method Chaining

Most pandas methods return a DataFrame so that another pandas method can be applied to the result. This improves readability of code.

```
df = (pd.melt(df)
      .rename(columns={
          'variable' : 'var',
          'value' : 'val'})
      .query('val >= 200')
      )
```

깔끔한 데이터 만들기(Melt, Pivot)

- Melt, Pivot 함수

`pd.melt`

Gather columns into rows(열의 값을 모아서 행으로 변경)

`df.pivot`

Spread rows into columns(행의 값을 열의 값으로 변경)



깔끔한 데이터 만들기(Melt, Pivot)

이론 영상 후 실습 영상 제시

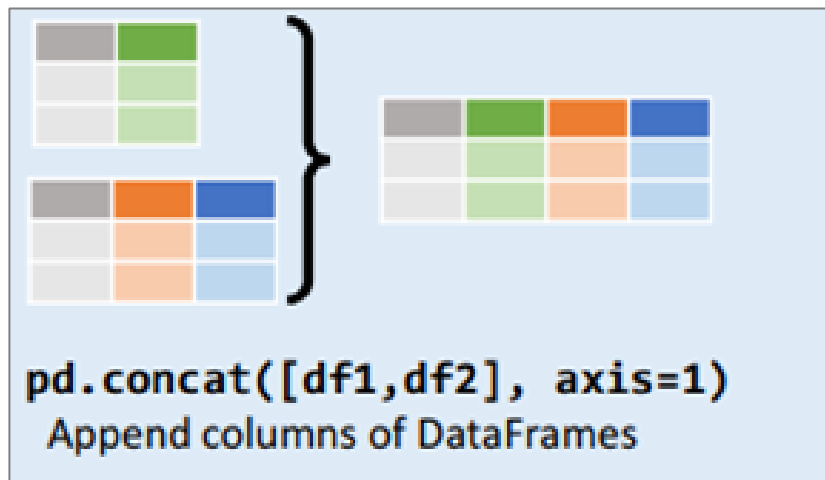
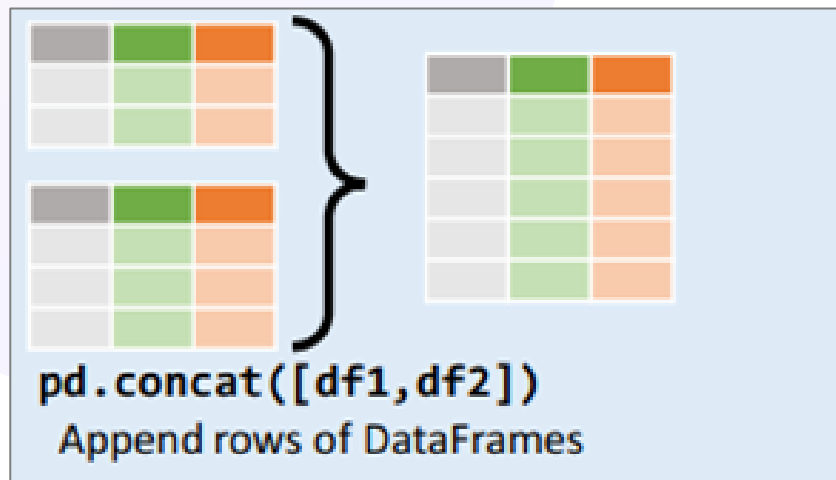




데이터 세트 합치기



데이터 합치기



`pd.concat`

Append rows of DataFrames(데이터 프레임 합치기)



이론 영상 후 실습 영상 제시



Merge로 데이터프레임 합치기

Combine Data Sets

adf			bdf		
x1	x2		x1	x3	
A	1	+	A	T	=
B	2		B	F	
C	3		D	T	

Standard Joins

x1	x2	x3
A	1	T
B	2	F
C	3	NaN

`pd.merge(adf, bdf, how='left', on='x1')`
Join matching rows from bdf to adf.

x1	x2	x3
A	1.0	T
B	2.0	F
D	NaN	T

`pd.merge(adf, bdf, how='right', on='x1')`
Join matching rows from adf to bdf.

x1	x2	x3
A	1	T
B	2	F

`pd.merge(adf, bdf, how='inner', on='x1')`
Join data. Retain only rows in both sets.

x1	x2	x3
A	1	T
B	2	F
C	3	NaN
D	NaN	T

`pd.merge(adf, bdf, how='outer', on='x1')`
Join data. Retain all values, all rows.

Filtering Joins

x1	x2
A	1
B	2

`adf[adf.x1.isin(bdf.x1)]`
All rows in adf that have a match in bdf.

x1	x2
C	3

`adf[~adf.x1.isin(bdf.x1)]`
All rows in adf that do not have a match in bdf.

Merge로 데이터프레임 합치기

pd.merge

- 데이터 프레임 병합
 - How = Left : Left Join
 - How = Right : Right Join
 - How = Inner : Inner Join
 - ➡ 두 데이터프레임이 동시에 해당하는 것만 병합
 - How = Inner : Outer Join
 - ➡ 두 데이터프레임의 모든 값을 병합
 - On = x1 : 'x1' Key를 중심으로 병합

Merge로 데이터프레임 합치기

Filtering Joins

```
adf[adf.x1.isin(bdf.x1)]
```

Indicator = True

어떻게 병합했는지 알려줌

Query

요청한 값만 가지고 옴



Merge로 데이터프레임 합치기

이론 영상 후 실습 영상 제시



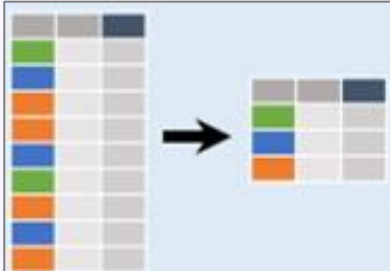


데이터 집계 활용



SQL 문 활용 데이터 집계

- Groupby



df.groupby(by="col")
Return a GroupBy object, grouped by values in column named "col".

df.groupby(level="ind")
Return a GroupBy object, grouped by values in index level named "ind".

All of the summary functions listed above can be applied to a group.
Additional GroupBy functions:

size() Size of each group.	agg(function) Aggregate group using function.
--------------------------------------	---

The examples below can also be applied to groups. In this case, the function is applied on a per-group basis, and the returned vectors are of the length of the original DataFrame.

shift(1) Copy with values shifted by 1.	shift(-1) Copy with values lagged by 1.
rank(method='dense') Ranks with no gaps.	cumsum() Cumulative sum.
rank(method='min') Ranks. Ties get min rank.	cummax() Cumulative max.
rank(pct=True) Ranks rescaled to interval [0, 1].	cummin() Cumulative min.
rank(method='first') Ranks. Ties go to first value.	cumprod() Cumulative product.

SQL 문 활용 데이터 집계

01 df.groupby

- `df.groupby(by="origin")['cylinders'].size()`
 - 특정 컬럼 값의 크기를 보고 싶을 때
- `df.groupby(by="origin")['cylinders'].mean()`
 - 특정 컬럼의 평균을 보고 싶을 때

02 shift()

- `df2.shift(1)` - 아래 행 방향으로 하나씩 Shift(이동)함
- `df2.shift(-1)` - 위 행 방향으로 하나씩 Shift(이동)함
- `df2['b'].shift(-1)` - 특정 Column(열)을 이동할 경우

SQL 문 활용 데이터 집계

03 rank() : 순위

- `df["model_year"].rank(method='min')`
 - 값이 적은 순서대로
- `df["model_year"].rank(pct=True) #pct`
 - 비율이 어느 정도인지 알려줌

04 cummax, cummin, cumprod

- `df2.cumprod()` - 누적 곱(Cumulative Product)



이론 영상 후 실습 영상 제시



학습목차	학습평가				화면설명
<div>평가하기</div> <div>- 학습평가</div> <div>-----</div> <div>정리하기</div> <div>- 학습정리</div>	학습한 내용을 바탕으로 다음 문제를 풀어 보세요.				<div>[학습평가 페이지]</div> <div>▪ 페이지 퀴즈 컴포넌트 사용하여 페이지 개발</div>
	번호	문제	정답	해설	
	1	Index가 없을 때, index를 생성해서 정렬하라는 판다스 명령어는 무엇인가? 1. df.sort_index 2. df.reset_index 3. df.rename 4. df.create_index	2	df.reset_index의 기능에 대한 설명이다.	
	2	df.pivot 판다스 명령어는 어떠한 역할을 하는가? 1. 열의 값을 모아서 행으로 변경 2. 행과 열을 새로 생성 3. 행의 값을 열의 값으로 변경 4. 행의 값을 새로 생성	3	df.pivot 명령어는 행의 값을 열의 값으로 변경하는 역할을 한다.	
	3	df2.shift(1) 판다스 명령어를 입력하면 어떠한 현상이 일어나는가? 1. 아래 행 방향으로 하나씩 shift(이동)함 2. 위 행 방향으로 하나씩 shift(이동)함 3. 아래 열 방향으로 하나씩 shift(이동)함 4. 위 열 방향으로 하나씩 shift(이동)함	1	df2.shift(1)을 입력하면, 아래 행 방향으로 하나씩 shift(이동)한다.	