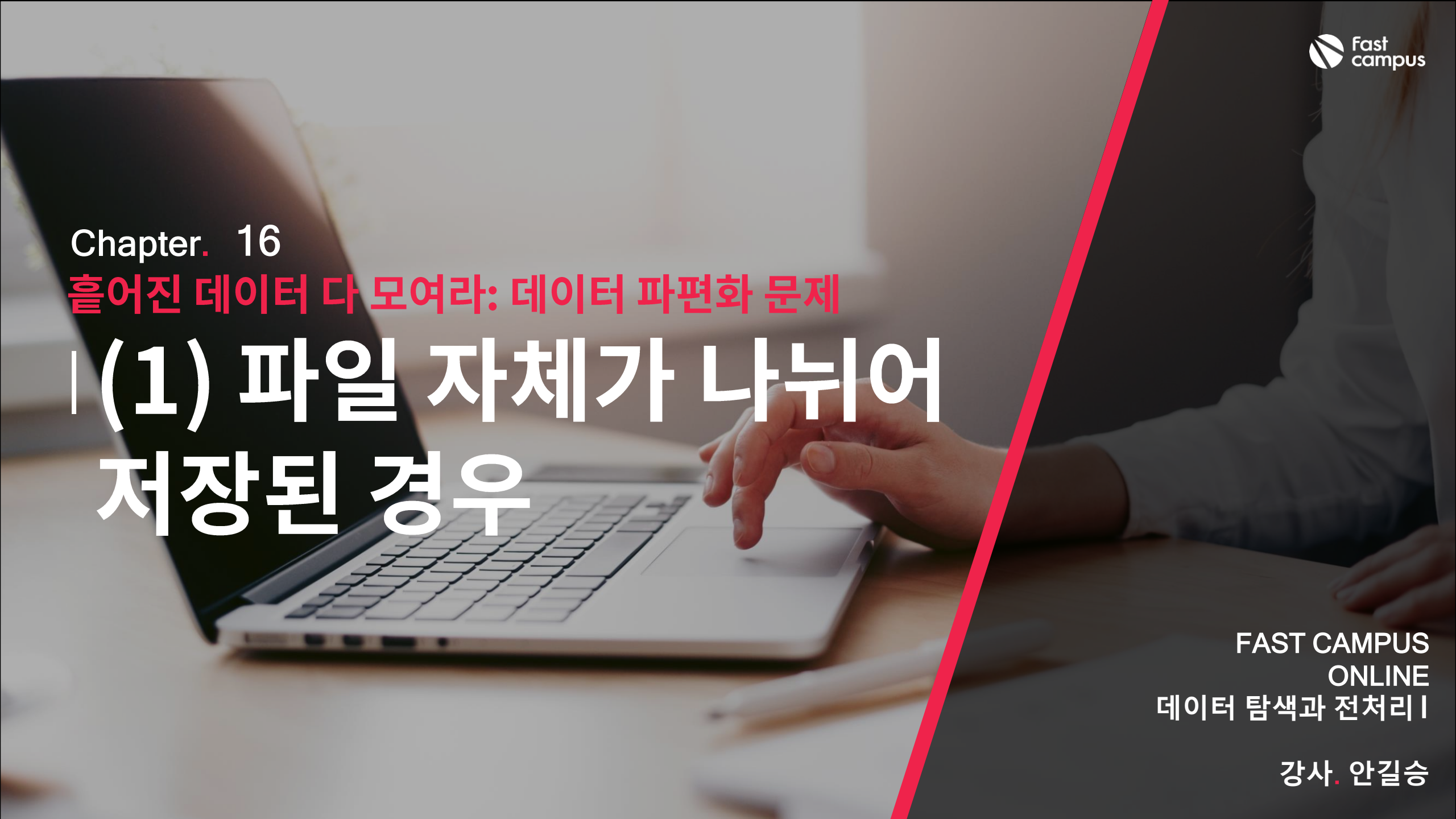


Chapter. 16

흩어진 데이터 다 모여라: 데이터 파편화 문제

(1) 파일 자체가 나뉘어 저장된 경우

FAST CAMPUS
ONLINE
데이터 탐색과 전처리 I

강사. 안길승

I 개요

- 지도학습 모델을 학습하려면 아래와 같이 반드시 **하나의 통합**된 데이터 집합이 필요함

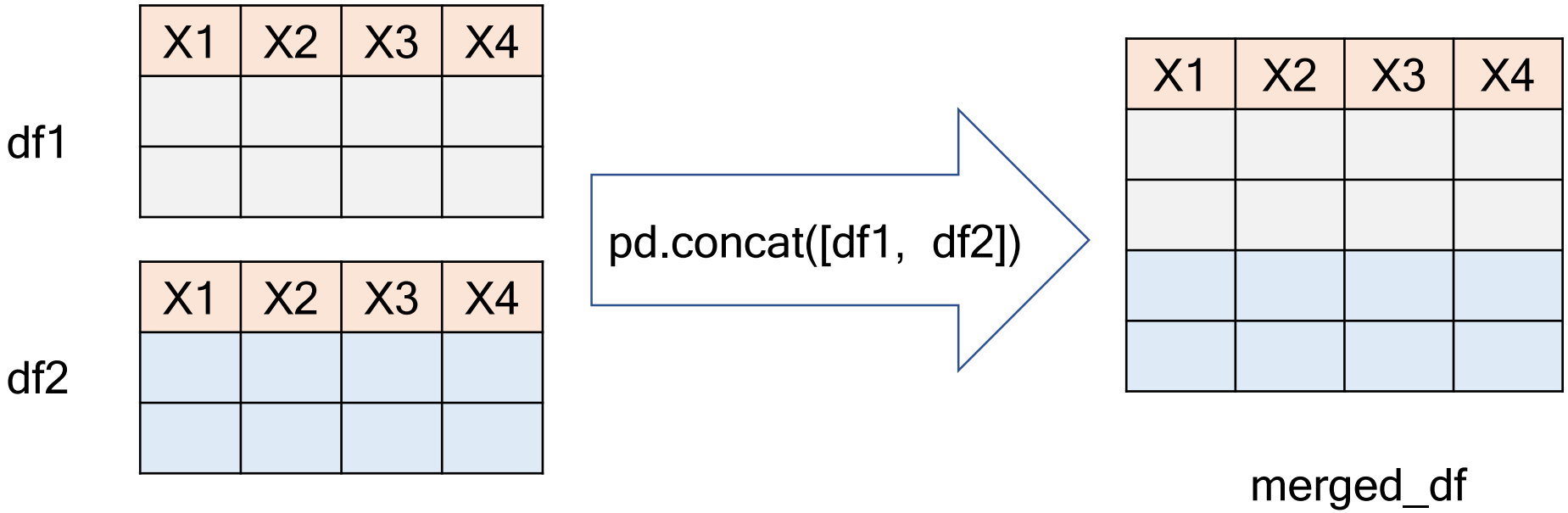
x_1	x_2	x_3	y
$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	y_1
$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	y_2
$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	y_3
$x_{4,1}$	$x_{4,2}$	$x_{4,3}$	y_4
$x_{5,1}$	$x_{5,2}$	$x_{5,3}$	y_5
$x_{6,1}$	$x_{6,2}$	$x_{6,3}$	y_6

- 많은 경우에 데이터가 두 개 이상으로 나뉘어져 있어, 이들을 **반드시 통합하는 전처리**를 수행해야 함

I 문제 정의 및 해결 방안

- 센서, 로그, 거래 데이터 등과 같이 크기가 **매우 큰 데이터는** 시간과 ID 등에 따라 **분할되어 저장**됨

- pandas.concat 함수를 사용하면 손쉽게 해결할 수 있음



- 통합해야 하는 데이터가 많은 경우에는 **빈 데이터프레임**을 생성한 뒤, 이 데이터프레임과 반복문을 사용하여 불러온 데이터를 concat 함수를 이용하면 효율적으로 통합할 수 있음

I 관련 문법: pandas.concat

- 둘 이상의 데이터 프레임을 이어 붙이는데 사용하는 함수

- 주요 입력
  - ➢ objs: DataFrame을 요소로 하는 리스트 (입력 예시: [df1, df2])로 입력 순서대로 병합이 됨
  - ➢ ignore_index: True면 기존 인덱스를 무시하고 새로운 인덱스를 부여하며, False면 기존 인덱스를 사용

df1

Index	X1	X2
a		
b		

df2

Index	X1	X2
a		
b		

pd.concat([df1, df2], ignore_index = True)

Index	X1	X2
0		
1		
2		
3		

  - ➢ axis: 0이면 행 단위로 병합을 수행하며, 1이면 열 단위로 병합을 수행

I Tip. Axis 키워드

- axis 키워드는 numpy 및 pandas의 많은 함수에 사용되는 키워드로, 연산 등을 수행할 때 축의 방향을 결정하는 역할을 함

- axis가 0이면 행을, 1이면 열을 나타내지만 이렇게만 기억하면 논리적으로 이상한 점이 존재함
  - (예시 1) sum(axis = 0): 열 기준 합
  - (예시 2) concat([df1, df2], axis = 0): 행 단위 병합

Index	X1	X2
a	1	4
b	2	5
c	3	6

df.sum(axis = 1)

Index	
a	1+4
b	2+5
c	3+6

df

df.sum(axis = 0)

X1	X2
1+2+3	4+5+6

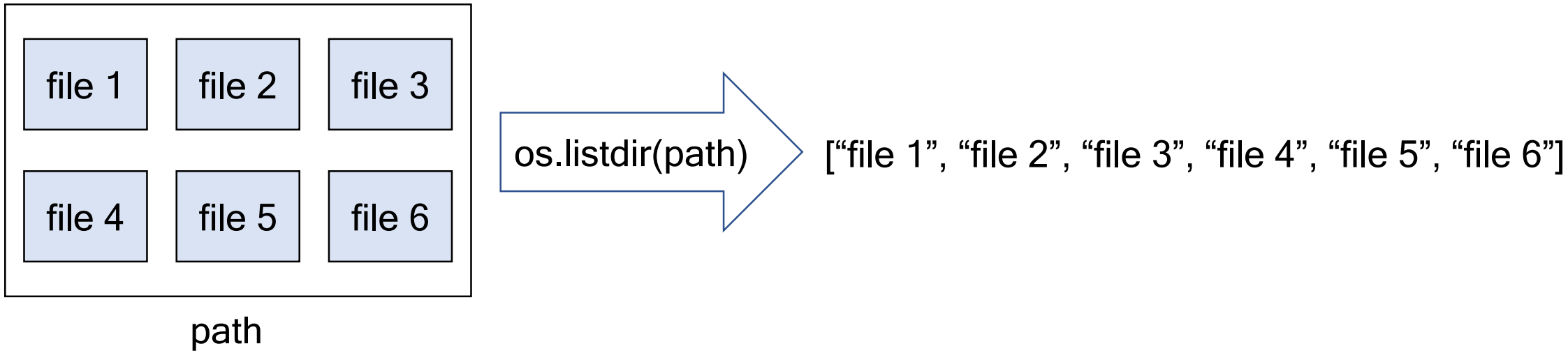
I Tip. Axis 키워드

- axis 키워드는 그 함수의 **결과 구조**가 벡터 형태 (1차원)인지, 행렬 형태 (2차원)인지에 따라, 그 역할이 조금씩 다름

		결과	
		벡터	행렬
axis	0	결과가 행벡터	연산 과정이 행 기준
	1	결과가 열벡터	연산 과정이 열 기준

I 관련 문법: os.listdir

- 주요 입력
  - ➢ path: 입력된 경로(path) 상에 있는 **모든 파일명을 리스트 형태**로 반환



Chapter. 16

흩어진 데이터 다 모여라: 데이터 파편화 문제

(2) 명시적인 키 변수가 있는 경우

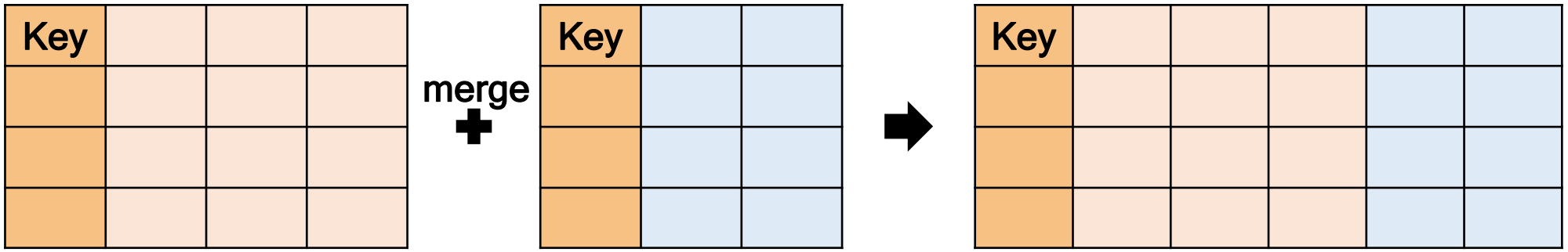
FAST CAMPUS
ONLINE
데이터 탐색과 전처리 I

강사. 안길승

I 문제 정의 및 해결 방안

- 효율적인 데이터 베이스 관리를 위해, 잘 정제된 데이터일지라도 데이터가 **키 변수를 기준으로 나뉘어 저장**되는 경우가 매우 흔함

- **SQL**에서는 **JOIN**을 이용하여 해결하며, **python**에서는 **merge**를 이용하여 해결함



- 일반적인 경우는 해결이 어렵지 않지만, **다양한 케이스가 존재**할 수 있으므로 반드시 핵심을 기억해야 함

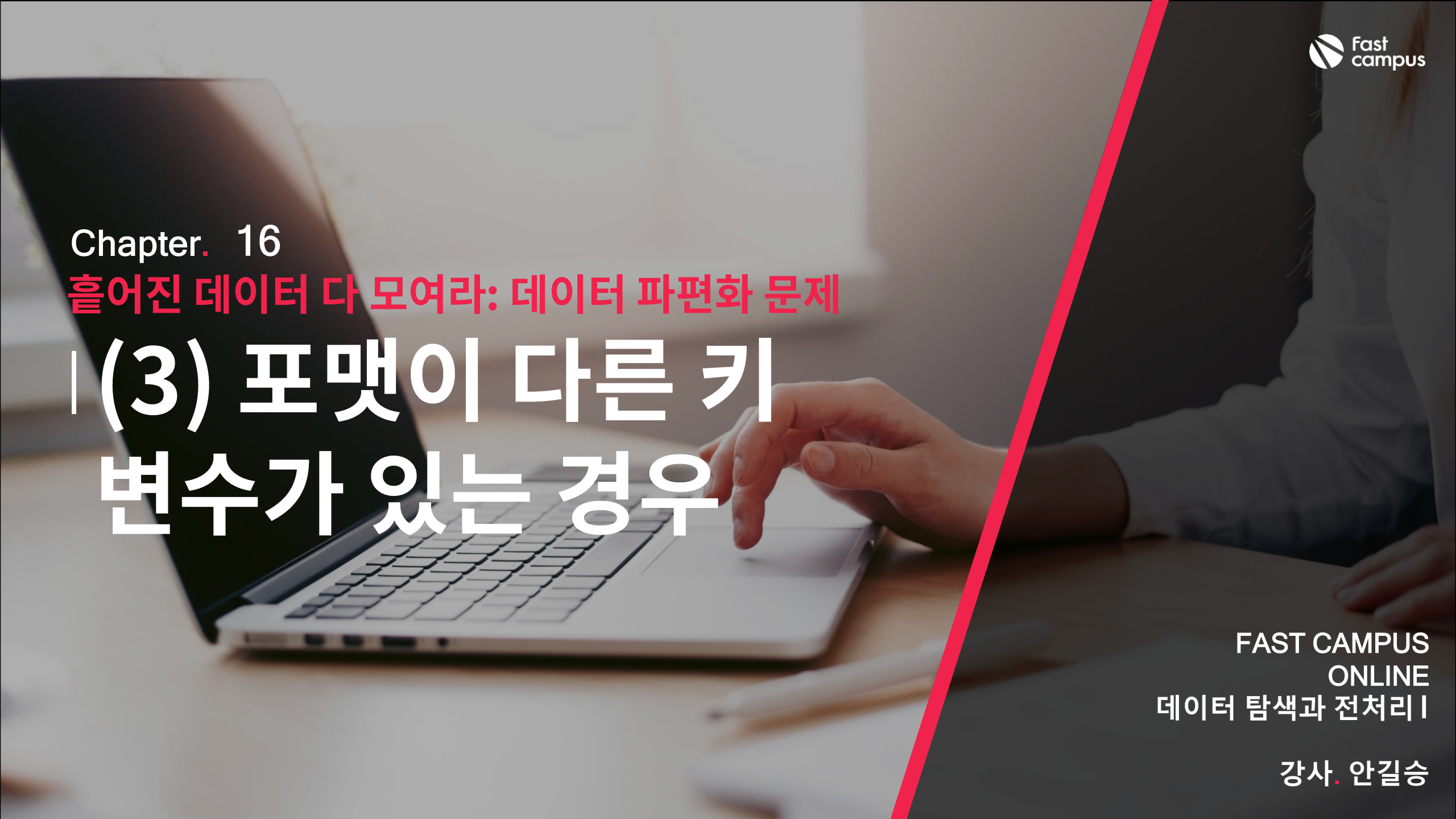
  (1) 어느 컬럼이 **키 변수 역할**을 할 수 있는지 **확인**하고, 키 변수를 **통일**해야 한다.

  (2) **레코드**의 단위를 **명확히** 해야 한다.

I 관련 문법: pandas.merge

- 키 변수를 기준으로 두 개의 데이터 프레임을 병합(join)하는 함수

- 주요 입력
  - left: 통합 대상 데이터 프레임 1
  - right: 통합 대상 데이터 프레임 2
  - on: 통합 기준 key 변수 및 변수 리스트 (입력을 하지 않으면, 이름이 같은 변수를 key로 식별함)
  - left_on: 데이터 프레임 1의 key 변수 및 변수 리스트
  - right_on: 데이터 프레임 2의 key 변수 및 변수 리스트
  - left_index: 데이터 프레임 1의 인덱스를 key 변수로 사용할 지 여부
  - right_index: 데이터 프레임 2의 인덱스를 key 변수로 사용할 지 여부



Chapter. 16

흩어진 데이터 다 모여라: 데이터 파편화 문제

(3) 포맷이 다른 키 변수가 있는 경우

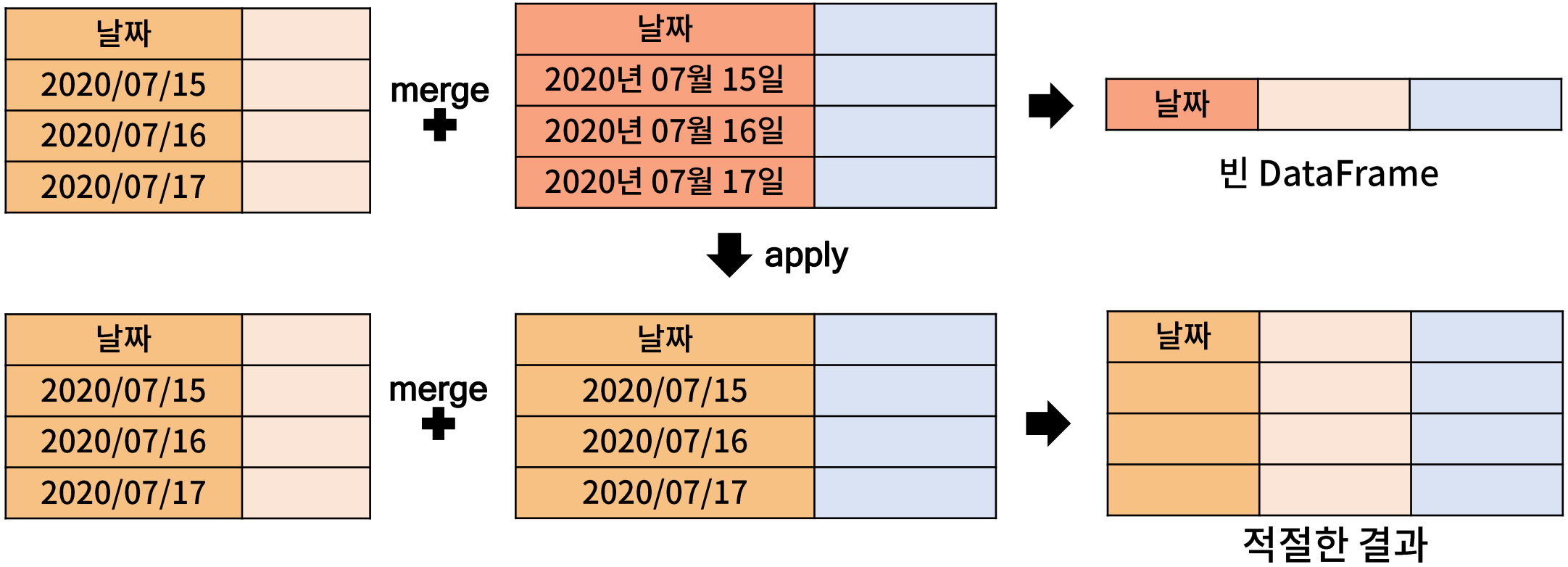
FAST CAMPUS
ONLINE
데이터 탐색과 전처리 I

강사. 안길승

I 참조 데이터가 필요 없는 경우의 병합

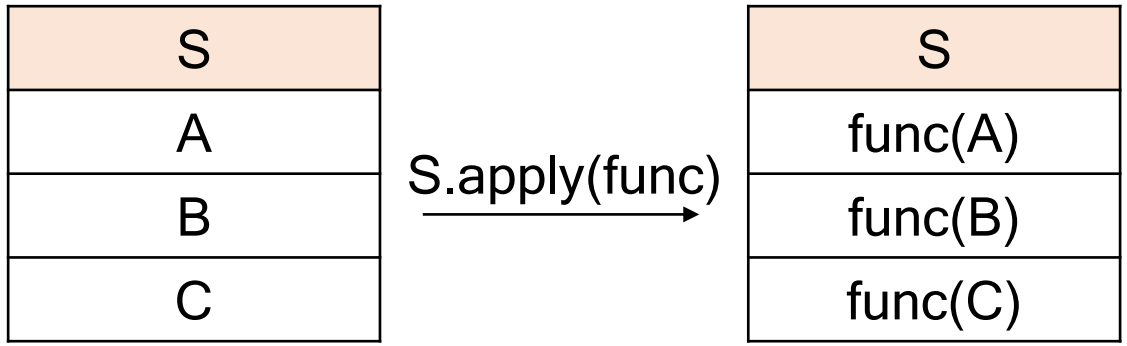
- 시간과 날짜 컬럼 등은 데이터에 따라 포맷이 다른 경우가 잦음

- 키 변수의 포맷이 다른 두 DataFrame에 대해 merge를 적용하면, 비정상적으로 병합이 이뤄질 수 있으므로, **하나의 컬럼을 다른 컬럼의 포맷에 맞게 변경해주는 작업**이 필요함



I 관련 문법: Series.apply

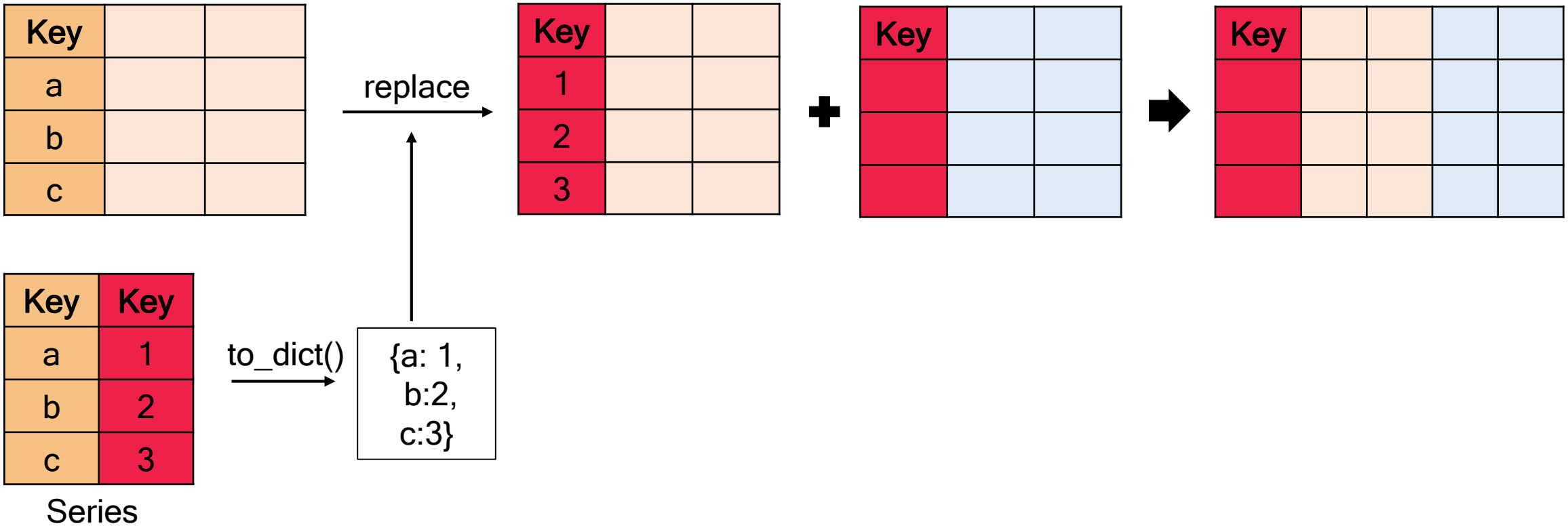
- Series에 있는 모든 요소에 func을 일괄 적용하는 함수 (built-in 함수인 map 함수와 유사)
- 주요 입력
  - ➢ func: Series의 한 요소를 처리하는 함수



- apply 함수는 머신러닝 코드의 **효율성**을 위해 굉장히 자주 사용됨

I 참조 데이터가 필요한 경우의 병합

- 도로명 주소 / 지번 주소, 회원명 / 회원번호 등과 같이 일정한 패턴이 없이 포맷이 다른 경우에는 컬럼 값을 **참조 데이터**를 이용하여 변경해야 함



I 관련 문법: Series.to_dict()

- Series의 Index를 Key로, Data를 Value로 하는 사전으로 변환

Index	Data
a	1
b	2
c	3
d	4

S (type: Series)

S.to_dict() →

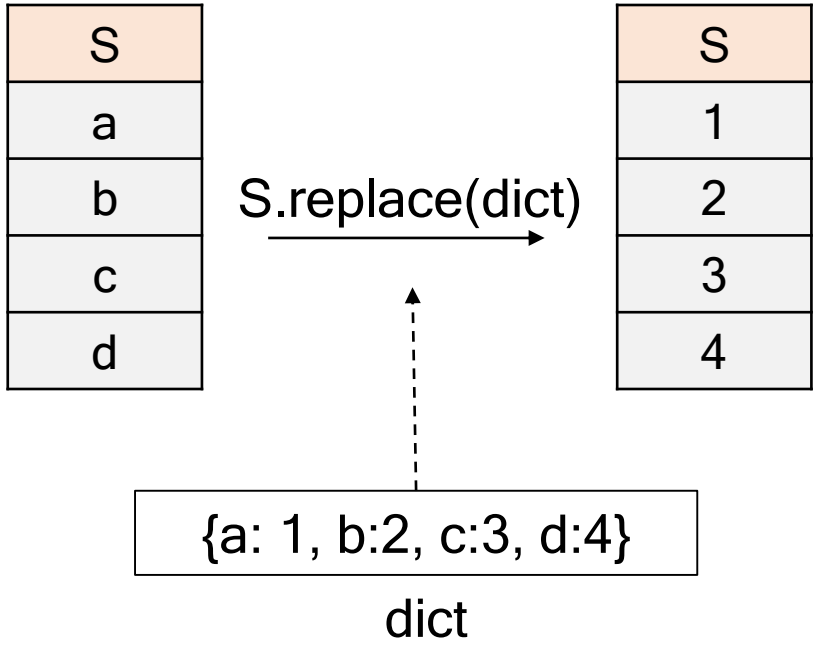
{a: 1, b:2, c:3, d:4}

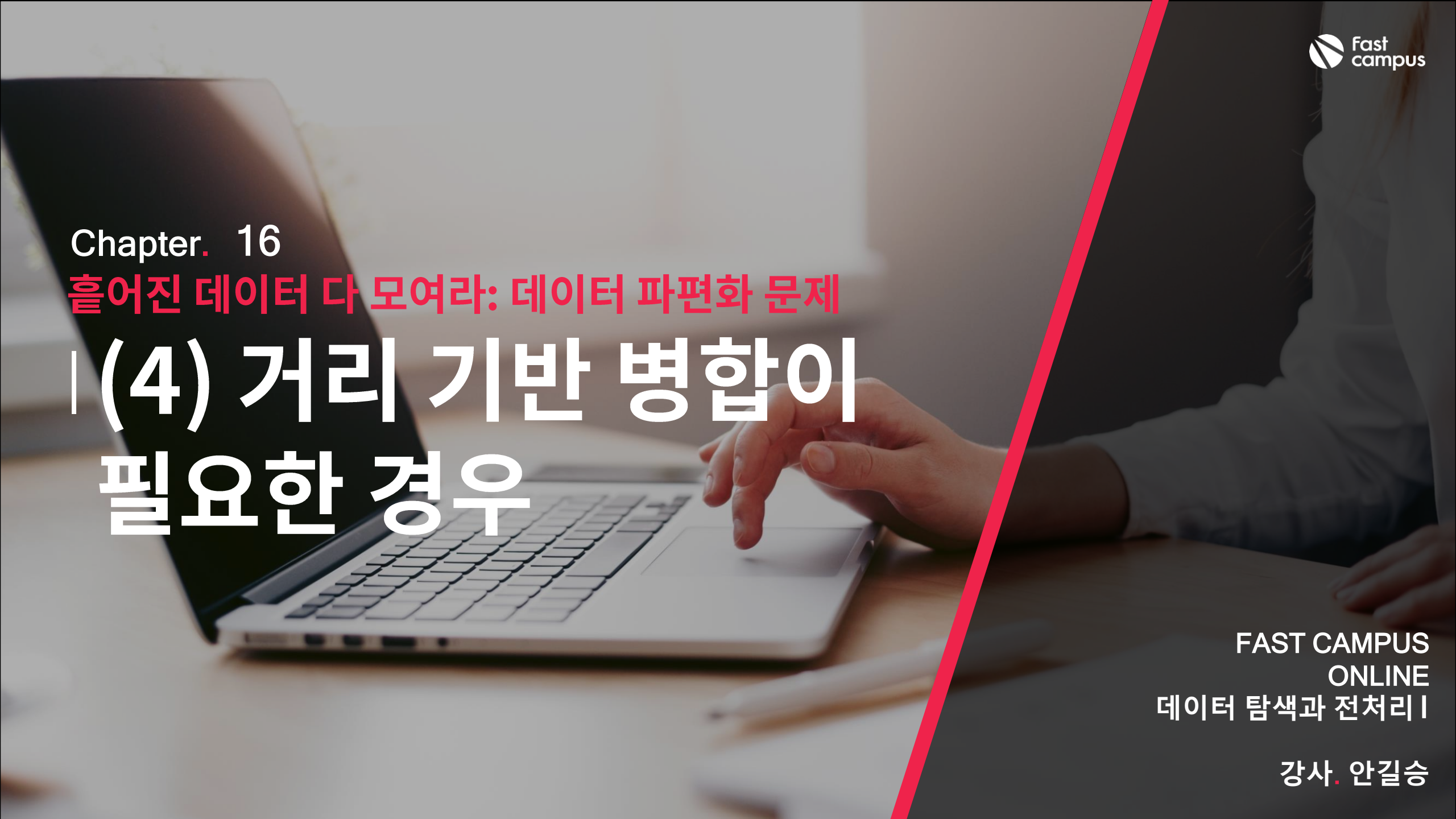
(type: dictionary)

- replace 등 사전을 입력받는 함수를 사용할 때 주로 사용

I 관련 문법: Series.replace

- dict을 입력 받아, Series 내에 있는 요소 가운데 key와 같은 값을 value로 변환해줌





Chapter. 16

흩어진 데이터 다 모여라: 데이터 파편화 문제

(4) 거리 기반 병합이 필요한 경우

FAST CAMPUS
ONLINE
데이터 탐색과 전처리 I

강사. 안길승

I 문제 정의 및 해결 방안

- 아파트 가격 예측 등 지역이 포함되는 문제에서 주소나 위치 변수 등을 기준으로 **거리가 가까운 레코드 및 관련 통계치**를 통합해야 하는 경우가 종종 있음

- 일반적으로 (1) 각 데이터에 포함된 레코드 간 거리를 나타내는 **거리 행렬을 생성**하고, (2) 거리 행렬의 행 혹은 열 기준 최소 값을 가지는 인덱스를 바탕으로 **이웃을 탐색**한 뒤, (3) 이웃을 기존 데이터에 부착하는 방식으로 해결함

주소	값1
A	a
B	b
C	c

주소	값2
X	x
Y	y
Z	z

주소	X	Y	Z
A	2	1.3	1
B	1.1	2.4	3.6
C	1.5	0.9	2.8

거리 행렬

→ Z
→ X
→ Y

이웃 탐색

주소	값1	값2
A	a	z
B	b	x
C	c	y

l 관련 문법: scipy.spatial.distance.cdist

- 두 개의 행렬을 바탕으로 거리 행렬을 반환하는 함수

- 주요 입력
  - XA: 거리 행렬 계산 대상인 행렬 (ndarray 및 DataFrame)로, 함수 출력의 행에 해당
  - XB: 거리 행렬 계산 대상인 행렬 (ndarray 및 DataFrame)로, 함수 출력의 열에 해당
  - metric: 거리 척도 (‘cityblock’, ‘correlation’, ‘cosine’, ‘euclidean’, ‘jaccard’, ‘matching’ 등)

A

Index	X1	X2
a	1	2
b	3	4

B

Index	X1	X2
c	5	6
d	7	8

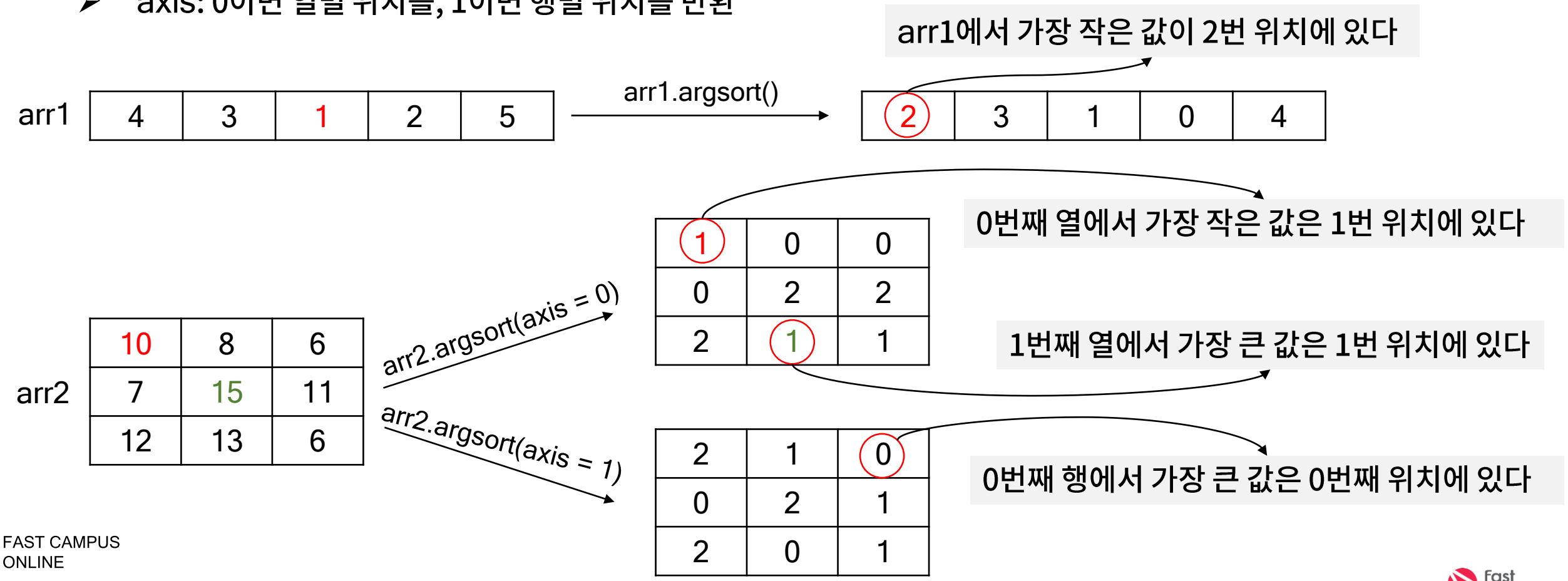
cdist(XA = A, XB = B,
metric = ‘cityblock’)

	c	d
a	$\|1-5\|+\|2-6\|$	$\|1-7\|+\|2-8\|$
b	$\|3-5\|+\|4-6\|$	$\|3-7\|+\|4-8\|$

I 관련 문법: ndarray.argsort

- 작은 값부터 순서대로 데이터의 **위치를 반환**하는 함수로, **이웃을 찾는데 주로 활용**되는 함수

- 주요 입력
  - ➢ axis: 0이면 열별 위치를, 1이면 행별 위치를 반환



Chapter. 16

흩어진 데이터 다 모여라: 데이터 파편화 문제

(5) 데이터 요약이 포함되는 경우

FAST CAMPUS
ONLINE
데이터 탐색과 전처리 I

강사. 안길승

I 문제 정의 및 해결 방안

- 보통 1:N 병합인 경우에 사용되며, 거래 데이터 및 로그 데이터와 병합하는 경우에 주로 사용됨
- 중복 레코드를 포함하는 **데이터를 요약한 후 병합**하는 방식으로 문제를 해결함

ID	B
1	$B_{1,1}$
1	$B_{1,2}$
1	$B_{1,3}$
2	$B_{2,1}$
2	$B_{2,2}$
3	$B_{3,1}$

요약 ➡

ID	B
1	$\frac{B_{1,1}+B_{1,2}+B_{1,3}}{3}$
2	$\frac{B_{2,1}+B_{2,2}}{2}$
3	B_3

✚

ID	A
1	A_1
2	A_2
3	A_3

➡

ID	A	B
1	A_1	$\frac{B_{1,1}+B_{1,2}+B_{1,3}}{3}$
2	A_2	$\frac{B_{2,1}+B_{2,2}}{2}$
3	A_3	B_3

l 관련 문법: DataFrame.groupby()

- 조건부 통계량(조건에 따른 대상의 통계량)을 계산하기 위한 함수로 머신러닝 프로세스 뿐만 아니라, 통계 분석 등에서도 굉장히 자주 활용됨

- 주요 입력
  - ➢ by: 조건 변수 (컬럼명 혹은 컬럼명 리스트로 입력)
  - ➢ as_index: 조건 변수를 index로 설정할 것인지 여부

- 활용 예시
  - ➢ df.groupby(['성별'])['신장'].mean() # 성별 (조건)에 따른 신장 (대상)의 평균 (통계량)

성별	신장
남성	180
여성	160
남성	170
여성	150

df

df.groupby(['성별'])['신장'].mean()

성별	신장
남성	175
여성	155

Chapter.

흩어진 데이터 다 모여라: 데이터 파편화 문제

감사합니다

FAST CAMPUS
ONLINE
데이터 탐색과 전처리 I

강사. 안길승