

Data Analysis with Pandas Pandas 활용

Jinwook Seo, Ph.D.

Human-Computer Interaction Lab

Department of Computer Science and Engineering Seoul National
University

지난 강의 내용 • 기계 학습의 기초

- Pandas를 활용한 결측 데이터 처리 • 결측 데이터 검색
- 결측 데이터 제거

- 결측 데이터 채우기 • Pandas를 이용한 파일 입출력
- 강의 내용 • 스택킹(stacking)과 피보팅(pivoting) • 데이터 결합과 조인 • 데이터 정렬 • 데이터 집계 함수 • 데이터 변환
- 데이터 랭글링 (Data wrangling) • 원자료(raw data)를 또 다른 형태로 가공하는 것

- 2개의 서로 다른 데이터를 연결 • 이종 데이터의 병합
- 데이터 간에 겹치는 부분의 처리 • 데이터 전처리
- 데이터 정리(cleansing)
- 결측값의 처리
- 데이터 변환(transformation)

(실습) DataFrame의 생성

<https://pastebin.com/HJXmCrsy>



```
In [1]: import pandas as pd
import numpy as np

def make_df(cols, ind):
    data = {c: [str(c) + str(i) for i in ind]
            for c in cols}
    return pd.DataFrame(data, ind)

make_df('ABC', range(3))
```

Out[1]:

	A	B	C
0	A0	B0	C0
1	A1	B1	C1
2	A2	B2	C2

실습용 데이터 • 미국 국립 알코올 남용 및 중독

연구소(NIAAA) 자료 • [niaaa-report2009.csv](#)

• [niaaa-report.xlsx](#) (8일차 강의에서 사용한 파일) • 미국 인구통계

데이터 (2016년 2월 기준) • population.csv

- 작업 중인 IPython notebook과 같은 폴더에 넣어주세요
- <https://goo.gl/7cAeSe>
- <https://goo.gl/gTr17p>
- <https://goo.gl/fWb4Xe>

실습용 데이터 읽기 (1) • Excel 파일 열기

- 예제 파일 (Niaaa-report.xlsx)
- <https://goo.gl/gTr17p>

<https://pastebin.com/c5YJqF5f>

```
In [1]: import pandas as pd  
df = pd.read_excel('niaaaa-report.xlsx', sheet_name='niaaaa-report')  
print(df.head())
```

	State	Year	Beer	Wine	Spirits
0	Alabama	1977	0.99	0.13	0.84
1	Alabama	1978	0.98	0.12	0.88
2	Alabama	1979	0.98	0.12	0.84
3	Alabama	1980	0.96	0.16	0.74
4	Alabama	1981	1.00	0.19	0.73

복습 – 계층적 인덱싱 
HUMAN COMPUTER INTERACTION

```
In [1]: import pandas as pd
index = [('California', 2000), ('California', 2010),
          ('New York', 2000), ('New York', 2010),
          ('Texas', 2000), ('Texas', 2010)]
index = pd.MultiIndex.from_tuples(index)
populations = [33871648, 37253956,
                18976457, 19378102,
                20851820, 25145561]
pop = pd.Series(populations, index=index)
print(pop)
```

```
California 2000    33871648
              2010    37253956
New York   2000    18976457
              2010    19378102
Texas      2000    20851820
              2010    25145561
dtype: int64
```

실습용 데이터 읽기 (2)

- 열 이름을 계층화하여 읽기

```
In [1]: import pandas as pd
df = pd.read_excel('niaaa-report.xlsx',
                    sheet_name='niaaa-report',
                    index_col=[0, 1])
print(df.head())
```

		Beer	Wine	Spirits
State	Year			
Alabama	1977	0.99	0.13	0.84
	1978	0.98	0.12	0.88
	1979	0.98	0.12	0.84
	1980	0.96	0.16	0.74
	1981	1.00	0.19	0.73

<https://pastebin.com/Zap1xLPk>

(실습) 스택킹(stacking)



```
In [1]: import pandas as pd  
df = pd.read_excel('niaaaa-report.xlsx',  
                    sheet_name='niaaaa-report',  
                    index_col=[0, 1])  
tall_df = df.stack()  
tall_df.index.names = ['State', 'Year', 'Drink']  
print(tall_df.head(10))
```

State	Year	Drink	
Alabama	1977	Beer	0.99
		Wine	0.13
		Spirits	0.84
	1978	Beer	0.98
		Wine	0.12
		Spirits	0.88
	1979	Beer	0.98
		Wine	0.12
		Spirits	0.84
1980	Beer	0.96	

dtype: float64

스택킹(stacking)과 언스택킹





(실습) 언스택킹(unstacking)







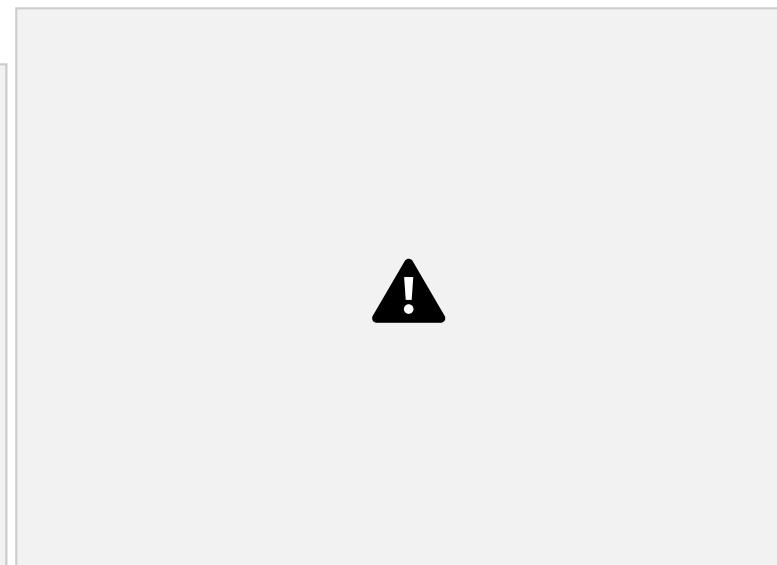
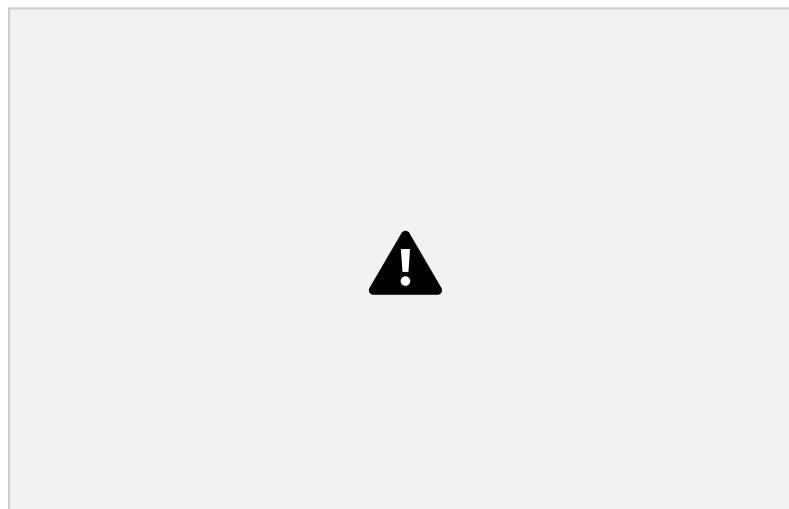
?



(참고) 결과를 엑셀로 저장하기



(실습) 스택킹과 언스택킹



- 스택킹과 언스택킹의 일반화된 연산 • `pivot(index, columns,`

values)

- Index
- 새로운 인덱스로 사용할 열 • Columns

- 새로운 열 이름 리스트

피보팅(Pivoting)

- Values
- 데이터프레임을 채울 데이터

(실습) 피보팅(Pivoting)





실습용 데이터 읽기 (3)

- niaaa-report2009.csv 파일 읽기





인덱스의 초기화 • `reset_index()` 함수

- 기존 인덱스의 지정 취소 및 초기화





실습용 데이터 읽기  (4)



복습 – NumPy 배열의 결합

- `concatenate()` 함수의 사용 •



축 방향의 지정 (axis 값)

설정)



Pandas에서의 결합 • 결합을 위해 merge 함수 사용 • 두 데이터 프레임에서 이름이 같은 열이 있는 경우• merge(df1, df2, on="key")
• 각 데이터 프레임에서 열을 지정하는 경우

- merge(df1, df2, left_on="key1", right_on="key2")

Pandas에서의 결합 • 왼쪽 DataFrame의 각 행이 오른쪽

DataFrame의 각 행에 하

나씩만 매칭되는 경우

• 왼쪽 DataFrame의 각

행이 2개 이상의 오른쪽 행에 매칭되

는 경우

• 왼쪽에 있는 행을 필요한

만큼 복제하고, 복제한만큼 행이 중복 됨

- 일대일(1:1) 결합
- 양쪽에서 필요한 만큼 행을 복제하고, 빈
곳에는 np.nan삽입
- 일대다(1:N) 결합



- 다대다(N:M) 결합

Pandas에서의 결합





Join의 처리

- 아래와 같이 2개의 데이터가 있는 경우를 가정• 일대다 결합:

왼쪽의 고객 정보를 필요한 만큼 복제

이름 핸드폰



도서번호

판매가격 주문일자 1 7000 2 13000 2014-07-03 5 8000 2014-07-03 2 13000 2014-07-04 4 35000 2014-07-05

3 22000 2014-07-07

이름 핸드폰

고객번호			1	박지성	000-5000-0001
			2	김연아	000-6000-0001

3	장미란	000-7000-0001
4	추신수	000-8000-0001

고객번호		
1	박지성	000-5000-0001
1	박지성	000-5000-0001
2	김연아	000-6000-0001
3	장미란	000-7000-0001
4	추신수	000-8000-0001

1	박지성	000-5000-0001
4	추신수	000-8000-0001

고객번호	3
1	
1	

2
3
4
1

4

Join의 처리

고객 주문

고객번호 판매가격 2 7000

- 외부 조인 이름 1 13000

2 8000

1 13000

4 35000

5 22000

4 22000

박지성 2 13000 박지성 4 13000 김연아 1 7000 김연아 3 8000 장미란 NULL 추신수 35000 추신수 22000 판매가격 1 2 7000 2 1 13000 3 2 8000 4 1 13000 5 4 35000 5 22000 4 22000 박지성 2

4 3000 박지성 4 13000 김연아 1 7000 김연아 3 8000 장미란 NULL 추신수 5 35000 추신수 7 22000 NULL 22000

고객 번호	
1	박지성
2	김연아
3	장미란
4	추신수

주문 번호

1		
2		
3		
4		
5		
6		
7		

고객

■

■

고객.고객번호=주문.고객번호

주문

■

고객 고객.고객번호=주문.고객번호

주문

■

고객 고객.고객번호=주문.고객번호

주문

① 왼쪽 외부조인 ② 완전 외부조인 ③ 오른쪽 외부조인

고객번호	이름	주문번호	판매가격
1			
1			
2			
2			
3		NULL	
4		5	
4		7	

고객번호	이름	주문번호	판매가격
1			
1			
2			
2			
3		NULL	
4			
4			
5		6	

이름	주문번호	고객번호
김연아		
박지성		
김연아		
박지성		
추신수		
NULL	6	
추신수	7	

결합 방법의 조정 • `join()`과 `merge()` 함수 모두 결합

옵션인 `how`를 지정 가능 • `left`: 왼쪽에 있는 DataFrame 인덱스를 기준으로 사용

- right: 오른쪽에 있는 DataFrame 인덱스를 기준으로 사용
- outer: 왼쪽과 오른쪽 DataFrame 인덱스들의 합집합을 기준으로 사용
- inner: 왼쪽과 오른쪽 DataFrame 인덱스들의 교집합을 기준으로 사용





how='inner' by default











(실습) DataFrame간의 결합





(참고) DataFrame간의 결합 • `reset_index()`를
수행하지 않을 경우 • `State`가 `index`로 지정되어 결합 시 참고할
열이 없어짐

- `reset_index()`를 수행하여 State column을 복구해야 함



(참고) DataFrame간의 결합 !



(실습) DataFrame간의 결합

- Index를 사용하여 결합하고자 하는 경우





DataFrame간의 결합 • 양쪽 DataFrame에서 index를
사용하여 결합할 경우 앞서
살펴본 merge() 함수 대신 join() 함수 사용 가능



- 결합 방법의 조정
 - 두 DataFrame에서 정확히 같은 이름의 열이 있을 경우 반드시 `lsuffix`와 `rsuffix`를 전달해야 함
 - 지정된 접미사를 각각 붙여줌



- 특정 축을 따라서 여러 DataFrame을 이어붙이는 경우
- concat() 함수 사용

DataFrame 블이기



DataFrame의 자료 정렬하기

- Index 값의 오름차순으로 정렬하기 • `sort_index()` 함수 호출 •
정렬은 기본적으로 사전순 (오름차순) • `ascending` 파라미터를
조정하여 순서 변경 가능 • `inplace=True` 옵션을 줄 경우 원본
데이터 프레임을 직접 변경 함 • 기본값은 `True`

(실습) Index의 순서로 정렬하기

- 인덱스의 값을 기준으로 오름차순 정렬



(실습) Index의 순서로 정렬하기

- 인덱스의 값을 기준으로 내림차순 정렬



DataFrame의 값으로 정렬하기

- 인덱스가 아닌 값을 기준으로 정렬해야 하는 경우
 - `sort_values(by, axis, ascending, inplace, ...)` 함수
 - `by`: 정렬 기준이 될 열 혹은 열의 리스트

- **axis**: 정렬 방향 (0: default)
- **ascending**: 오름차순 정렬이 필요한지에 대한 참/거짓 혹은 그에 대한 리스트 (by의 리스트 요소 수와 동일해야 함)
- **inplace**: 해당 DataFrame을 직접 수정할지에 대한 옵션
(실습) DataFrame의 값으로 정렬하기
- Population을 기준으로 정렬





순위 계산하기

- DataFrame이나 Series에서 순위를 계산할 경우 rank() 함수를 사용 • 같은 순위에 대해서 기본 옵션은 평균적인 순위를 부여 •

`method` 옵션을 지정하여 같은 그룹내의 항목들에 대한 처리 방법

변경 가능 • `average`: 그룹 내의 항목들에 대해 평균 순위 사용

- `min`: 가장 작은 값 사용 • `max`: 가장 큰 값 사용
- `first`: 처음 등장한 값부터 순서대로 순위 배정
- `dense`: `min`과 같지만 group간에 1씩만 차이가 나도록 촘촘하게

순위 배정

순위 계산하기





(실습) 순위 계산하기

- 주의 이름 순으로 정렬한 후, 인구 순위 구하기



(실습) 인구 정보 + 순위 (1) • 기존 DataFrame에

인구 순위를 합친 결과를 구해보세요.

- 같은 그룹의 rank는 min 사용



(실습) 인구 정보 + 순위 (2)

- 위에서 구한 결과를 인구수에 대한 내림차순으로 출력해보세요.



데이터 집계 함수 • Series나 DataFrame의 각 열에서

기술 통계값을 계산• sum(), mean(), median(), std() – 표준편차,
count(), • 결측치의 처리 방법

- skipna를 True로 설정할 경우 nan값을 분석에서 제외• 연산의 방향 지정
- axis 값을 지정하여 수평 혹은 수직 방향으로 연산min(), max()
(실습) 데이터 집계 함수
• 각 주류별로 최대값을 구해보세요.





(실습) 데이터 집계 함수

- 각 주별로 최소값을 구해보세요.



데이터 집계 함수

- 전체에 대한 기술 통계값이 필요한 경우 `describe()`

함수로 한번에 도출 가능





데이터 변환

- 사칙 연산자의 활용 •



(참고)

문자열로 된 column명은 위와 같이 사용 가능

데이터 변환 • groupby() 함수

- 각 열의 키 값을 기준으로 행을 그룹으로 묶어 데이터 프레임을 분리 함



groupby() 함수의 응용

- for문의 활용



DataFrame의 이산화 • 히스토그램 작성을 위하여

연속적인 변수를 이산 변수로 변환 가능

• cut() 함수의 활용



DataFrame의 이산화 • 이산화된 구간에 레이블을

붙이고 싶은 경우 **labels** 활용

- 앞서 지정한 bin의 수만큼 리스트에 **label**을 저장하여 함수 호출



- **labels=False**로 지정할 경우, 각 구간의 소속 정보만 반환
데이터 변환 – Mapping
- 가장 일반적인 형태의

데이터 변환

- 사용자가 지정한 함수를 선택한 열의 모든 항목들에 적용
- 전달 가능한 함수

- Python 내장함수
- Import한 모듈의 함수
- 사용자가 정의한 함수
- 익명 함수

(lambda 함수)

(실습) 데이터 변환 – Mapping • 각 주별로 세글자
약어 부여하기

- 아래의 예제에서는 Lambda 함수 활용





데이터 변환 – 교차 집계 • 교차

집계(Cross-tabulation)

- 각 그룹별 빈도를 산출하고, 서로 다른 두 카테고리 변수를 표현하는 행과 열로 된 DataFrame 반환



- 스택킹(stacking)과 피보팅(pivoting)
- 데이터 결합과 조인
- 데이터 정렬

Summary

- 데이터 집계 함수
- 데이터 변환

