|  |  |
| --- | --- |
| 교육 제목 | **데이터프레임의 다양한 응용** |
| 교육 일시 | 2021년 11월 18일 목요일 |
| 교육 장소 | 영우글로벌러닝 2층 |
| **교육 내용** | |
| 오전 | 1. Series와 DataFrame에 함수 적용하기    1. map, apply 함수 : Series에서 사용 가능하며 모든 요소에 함수 일괄 적용    2. apply 함수 : DataFrame에서 사용 가능하며 각각의 행 또는 열(Series)에 함수 일괄 적용    3. applymap 함수 : DataFrame에서 사용하며 모든 요소에 함수 일괄 적용 2. 그룹 객체 만들기 : groupby()    1. groupby()함수는 그룹객체를 만들어주는 함수로 Dataframe객체.groupby(기준이 되는 열이름)로 사용된다.   titanic = sns.load\_dataset('titanic')  df = titanic[['age','sex','class','fare','survived']]  print("승객 수 :", len(df))  print(df.head())   1. class열에는 first, second, third라는 3개의 값들이 들어 있다. 이 열을 기준으로 그룹객체를 생성해보자.    1. grouped = df.groupby('class')    2. print(grouped)    3. print(type(grouped)) 2. 그룹객체는 그룹이름(key)과 그룹으로 쪼개진 데이터(group)를 튜플형태로 묶어서 가지고 있는다. 또한 행 인덱스 번호를 확인해보면 잘라놓은 것 처럼, 기존의 행 인덱스번호를 그대로 가져오는 것을 알 수 있다.    1. for key, group in grouped:    2. print("\* key", key)    3. print("\* count", len(group))    4. print(group.head())    5. print('\n') 3. 그룹 추출 : group객체.get\_group('그룹이름')    1. get\_group() 함수를 그룹객체에 적용해 나눠진 그룹중 가져오고 싶은 그룹만 가져올 수도 있다.    2. First그룹만 추출해보자    3. group1 = grouped.get\_group('First')    4. group1.head() 4. 여러개의 함수를 여러 열에 적용 : agg()함수    1. 사용자 정의함수를 적용       1. def min\_max(x):       2. return x.max()-x.min()    2. 일종의 매핑함수를 적용하는 것과 비슷하다.       1. grouped.agg(min\_max) 5. 그룹객체 필터링 : filter()    1. 그룹객체에 대한 filter()함수는 내장함수의 filter함수와는 전혀 다른 기능을 한다. group객체.filter(조건식 함수)로 사용하며, 개별원소에 대한 필터링이 아니라, group객체를 필터링 한다는 것을 유념해야 한다.즉, 그룹핑한 그룹들중에 가져올 그룹과 안가져올 그룹을 필터링 하는 게 핵심이다.추가로 그룹객체에 매핑함수를 적용할때는 lambda x에 해당하는 x가 각각의 그룹들이라는 것을 생각하자. |
| 오후 | 1. DataFrame Join 2. Join 이란?    1. 두 개의 DataFrame을 합치는 것       1. 열기준 컬럼명으로 합치기 : merge       2. 열기준 Index명로 합치기 : merge, concat       3. 행기준으로 합치기 : concat, append 3. 합치는 방법은?    1. Inner : 두 DataFrame의 기준 컬럼에서 둘 다 존재하는 데이터만 Join    2. Left Outer join : 왼쪽 DataFrame으로 합치기    3. Right Outer Join : 오른쪽 DataFrame으로 합치기    4. Outer Join : 두 DataFrame의 모든 Data를 합치기 4. Merge 명령어 사용법?    1. pd.merge(left, right, how, on, left\_on, right\_on, left\_index, right\_index)    2. left : Merge 할 왼쪽 DataFrame    3. right : Merge 할 오른쪽 DataFrame    4. how : inner, left, right, outer    5. on : 두 DataFrame을 Join할 기준 컬럼명(컬럼명이 동일할 경우)    6. left\_on : Join할 기준 컬럼의 왼쪽 DataFrame의 컬럼명    7. right\_on : Join할 기준 컬럼의 오른쪽 DataFrame의 컬럼명    8. left\_index : 왼쪽 DataFrame index로 Join할 경우 True    9. right\_index : 오른쪽 DataFrame index로 Join할 경우 True 5. concat 명령어 사용법?    1. pd.concat([left,right], axis, join,...)    2. [left,right] : left DataFrame과 right DataFrame으로 이루어진 List    3. axis = 0 : 행기준으로 합치기, 1: 열기준으로 Index명으로 합치기 |