|  |  |
| --- | --- |
| 교육 제목 | AI 프로젝트 기반 S/W 전문가 양성과정 |
| 교육 일시 | 21.11.08 |
| 교육 장소 | 대면(강의실) |
| **교육 내용** | |
| 오전 | #판다스 https://pandas.pydata.org/pandas-docs/stable/?v=20211108091159  # 시리즈 : 판다스에서 리스크와 같은 개념이지만 동일한 데이터가 1차원 배열로 들어간 구조, 인덱스가 만들어져.  # 딕셔너리 -> 판다스 시리즈 변환  # 시리즈변수=pandas.Series(딕셔너리변수)  # 딕셔너리 키 값이 인덱스로 변환 되어 들어간다  # list data type도 변환 가능. 리스트 인덱스가 곧 시리즈 인덱스  # Series 접근 방법  # 정수형 인덱스 위치와 인덱스 네임 두가지로 접근 가능  #리스트로 인덱싱  #여러개 값을 동시에 출력하고 자 할때 [인덱스명 또는 정수인덱스]  #정수형 인덱스로 슬라이싱 할때 start만 포함 end는 미포함  # 인덱스명으로 슬라이싱 하면 start와 end 둘다 포함 범위  #데이터 프레임  #2차원 배열, R의 데이터 프레임에서 유래함  #행과 열로 구분  #칼럼은 어떤 자료 형을 넣을 지  #Row는 고유 이름  #데이터 프레임은 딕셔너리가 여러개라고 보면 됨  #시리즈가 모여서 2차원 배열  #데이터프레임=pandas.DataFrame(딕셔너리 객체)  #pd.DataFrame(2차원데이터, index=인덱스명, columns=컬럼명)  #datafrmae 제작  #숫자 1~10까지 자료를 가지고 2차원 배열 (5 \* 2)로 구성 한 데이터 프레임 생성  #columns 명 : 'c\_1', 'c\_2' 인덱스명은 'idx1''idx2''idx3''idx4''idx5'  #데이터프레임 변수명은 df  #자료의 인덱스명과 컬럼명을 출력, 전체자료 출력  #인덱스명 변경  df.index = ['id1', 'id2', 'id3', 'id4', 'id5']  #변경할 칼럼명 리스트  #df.columns = [칼럼명 리스트]  #df.rename(index = {기존칼럼명:새로운칼럼명}, inplace=True)  # inplace=True -> 변경사항 적용 한다는 flag  #인덱스 삭제, 컬럼 삭제  #행삭제  #df.drop(삭제할 인덱스명, axis=0, inplace=True)  #열삭제  #df.drop(삭제할 칼럼명, axis=1, inplace=True)  #행 삭제  데이터프레임명.drop(인덱스명,axis=0,inplace=True)  #열 삭제  데이터프레임명.drop(칼럼명,axis=1,inplace=True)  #행 선택  #인덱스 접근 방법  # loc:인덱스명으로 접근  label\_1 = df.loc['서준']  # iloc : 정수 인덱스로 접근  position\_1 = df.iloc[0]  #여러 데이터 동시 가져오기  df.loc['서준':'인아']  #열 접근 방법  # 데이터프레임.컬럼명 or 데이터프레임[컬럼명]  #여러개 동시 출력 가능  #칼럼은 순서가 있는게 아니라 콜론으로 처리하지 않는게 좋다  # 행과 열 접근 방법  # df.loc[인덱스명 , 컬럼명]  #컬럼명에 콜론 입력하면 모든 컬럼을 가져온다  df.loc['서준', :]  #특정 칼럼 데이터만 가져올때  df.loc['서준','영어']  #특정 칼럼 여러 데이터 가져올때  df.loc['서준',['영어','수학']]  df.iloc[0, 0:3]  #모든 데이터 출력  df.loc[:,:]  #데이터 프레임에 신규 인덱스 (행)추가  #데이터프레임명.loc[추가할인덱스] = [리스트형 데이터]  #데이터 프레임에 칼럼 추가  # 데이터프레임명[추가할 칼럼명]= 일괄 삽입 할 데이터  #기존 행을 복사해서 행을 추가하는 방법  # df.loc[인덱스명] = df.loc[기존인덱스]  #컬럼과 인덱스 위치 변경  transpose()  #접근 최종 정리  #df.loc[인덱스명, 칼럼명]  #df.iloc[인덱스 정수, 칼럼 정수] |
| 오후 | #인덱스  #특정한 칼럼(열)을 인덱스 칼럼으로 설정  # df.set\_index(칼럼명, inplace=True)  #인덱스 재배열  #df.reindex(새로운 인덱스, fill\_value = 0, copy = True)  #기존의 자료는 유지하고 새로운 자료를 생성 하면서 인덱스 추가할 때  fill\_value = 0 => 빈곳을 0으로 채운다는 매개변수 전달  #인덱스를 칼럼으로 변환, 인덱스 제거하면 정수 인덱스가 나옴  df\_reset = df.reset\_index()  #열 기준으로 정렬  #df.sort\_values(by='c0',ascending=False)  #인덱스 기준으로 정렬  #df.sort\_index(ascending=False)  #sort\_index -> 인덱스 기준으로  #sort\_values -> 열 기준으로  #(by = 컬럼명, ascending = True or False)  #산술연산  #시리즈 type 연산  #시리즈를 숫자와 연산  #시리즈 끼리 연산  #연산 메소드 이용하기  sr\_add = A.add(B, fill\_value = 0)  add, sub, mul, div  #데이터 프레임 연산  Series와 같음.    #head(n)  #n만큼 보여주고 기본 5,  #tail(n)  #마지막 부분부터 보여주는 것  기타 데이터 프레임 관련  #데이터 프레임의 칼럼과 데이터 갯수 등 정보 보여줌  데이터프레임.info()  #통계 정보 보여줌. 합게, count 등등  데이터프레임. describe()  #데이터 입출력  #파일 형식에 따라 read 가능  #<format> = csv, excel, Json, html  읽기 : 데이터프레임.read\_<format>(경로)  저장 : 데이터프레임.to\_<format>(경로)  Csv 형식  #인덱스 칼럼을 지정 하면서 데이터를 가져옴  df2 = pd.read\_csv(file\_name,index\_col='c0')    #인덱스 칼럼과 칼럼명을 지정 하면서 데이터 가져옴  df3 = pd.read\_csv(file\_name,index\_col='a',skiprows=1, names=['a','b','c','d'])  예제)  #seaborn의 taitanic 자료의 age, fare , class 칼럼만 10번째 데이터 부터  #60번째 데이터 추출, titanic.csv파일로 저장  import pandas as pd  import seaborn as sns  titanic = sns.load\_dataset('titanic')  df = titanic.loc[10:60,['age','fare', 'class']]  #csv 저장  df.to\_csv("./data/titanic.csv")  #sample 20개만 추출  df\_random = df.sample(20, random\_state=100)  df\_random.to\_csv("./data/titanic\_random.csv") |