|  |
| --- |
| **ㅇ쇼SSAC**  **파이썬을 활용한 비즈니스 빅데이터 분석가 양성 과정**  **(TEMPLATE) 자가학습 워크시트** |

|  |  |
| --- | --- |
| **이름** | 백승재 |
| **날짜** | 2021-10-25 |
| **온라인 콘텐츠(인강)**  **학습 범위** | * 직장인을 위한 Python : 전처리 CH02\_04~11 |
| **복습 노트 작성 (자유 양식)** | Ch-전처리-04.DataFrame 합치기 (concat)  횡방향(column) df.concat([df,df1],axis=1)  종방향(index) df.concat([df,df1],sort=False).reset\_indext(drop=True)  Ch-전처리-05.DataFrame 병합하기 (merge) : JOIN(left,right,inner,outer)  pd.merge (df,df2, on = “key“, how = “left(right,outer,inner “)  Ch-전처리-06.Series의 Type변환하기  - Step1: 데이터 이해(df.info) 결측값, 데이터 타입 확인  - Step2: 결측값 처리(df[‘키’].fillan(df[‘키’.mean])  - Step3 : df[‘키’]=df[‘키’].astype(int) (astype은 inplace기능이 없으므로 재정의 필요)  - series datatype(object, float, int, category, datetime)  Ch-전처리-07.시간을 다루는 Datatime 타입  Step1: 문자열 타입- -> 데이터 타입으로 변경  Step2: datetime(dt) 메소드를 이용하여 날짜 정보 추출  df['생일\_년'] = df['생년월일'].dt.year  df['생일\_월'] = df['생년월일'].dt.month  df['생일\_일'] = df['생년월일'].dt.day  Ch-전처리-08. Apply  Step1: 함수 만들기  Step2: 함수를 전체 Dataframe 혹은 특정 열에 적용하기(Apply)  Ch-전처리-09.Lamda,map  df['키/2'] = df['키'].apply(lambda x:x/2)  my\_map = { '남자': 1, '여자': 0 } -> df['성별'].map(my\_map)  Ch-전처리-10.데이터프레임의 산술연산  - series type과 data frame의 broadcasting 을 통한 산술 연산  Ch-전처리-11.select\_dtypes  Step1: df.info  Step2: df.select\_dtypes(include="object"), df.select\_dtypes(exclude="object") |
| **퀴즈 제출** | Q1. DataFramed cp\_df, df을 컬럼 기준으로 합치고 결측값을 평균값으로 채워준 것으로 적절한 것은?  1) df\_concat = pd.concat([df,df\_copy],sort=False).reset\_index(drop=True)  2) df1 =pd.concat([cp\_df,df]),axis=1)  3) df1 =pd.concat([cp\_df,df],axis=1)  df1.fillna(df3.mean())  4) df\_concat = pd.concat([df,cp\_df],sort=False)  정답 3 : 컬럼 기준합치고 결측값 평균값으로 채워줌  Q2. 데이터 프레임 df,cp\_df을 이름(성함)순으로 공통된 컬럼만 merge한 것은?  # df는 ‘이름’ 컬럼에 cp\_df은 ‘성함’ 컬럼에 그룹 멤버의 구성원의 이름이 요소로 구성되어 있다.  1) pd.merge(df,cp\_dft,on="이름", how = "inner")  2) pd.merge(df,cp\_df,left\_on="이름",right\_on="성함", how = "inner")  3) pd.merge(df,cp\_df,left\_on="이름",right\_on="성함", how = "outer")  4) pd.merge(df,cp\_df,on="성함", how = "right")  정답 : 2  - 두 데이터 프레임의 키는 이름이 다르기 때문에 left\_on과 right\_on으로 키값을 각각 설정해준다.  - 1) 4) Key Error 발생    - 두 데이터 프레임의 공통 컬럼이므로 inner Join 방식이다.  Q3.데이터 프레임 중 결측값이 포함된 ‘키’컬럼의 데이터 타입을 int32로 변경하는 절차로 적절한 것은?  a. df.info()  b. df['키'] = df['키'].fillna(df['키'].interpolate(method="linear"))  c. df = df.astype({"키" : "int"})  d. df[키].astype(int)  e. df = df[키].astype(int)  1) a -> d -> c  2) a-> b -> d  3) a-> b -> c  4) d->a-> b  정답 : 3 데이터 타입 확인 -> 결측값 확인 -> 데이터 타입 변경 후 변경 열 재정의  - 데이터 타입 변경은 C와 d 중 택 1하여 작업  3) df['키'].fillna(df['키'].median())  4) df3['키']=df3['키'].fillna(df['키'].max())  정답 : 3 원본에 적용이 안되며 변경 시 inplace옵션 또는 fillna을 이용해 값을 넣어줘야 한다.  Q4. 데이터 프레임에 ‘성별’ 컬럼에서 남자는 1, 여자는 0으로 변경하려고 하는 코드로 적절하지 않은 것은?  1) my\_map = {'남자':1, '여자':0}, df['성별'] = df['성별'].map(my\_map)  2) df.loc[df.성별=='여자'] =0, df.loc[df.성별 == '남자] = 1  3) f = lambda x:1 if x=='남자' else 0  df['성별']=df['성별'].apply(f)  4) def male\_or\_female(x):  if x == '남자':  return 1  elif x == '여자':  return 0  df['성별\_NEW'] = df['성별'].apply(male\_or\_female)  정답 : 2 loc를 이용하여 성별 컬럼에 각각 인데싱을 한후 해당 컬럼을 지정한 후에 값을 부여해야함  df.loc[df.성별=='여자','성별'] =0, df.loc[df.성별=='남자','성별'] =1, df  Q5. 데이터 프레임에서 컬럼 데이터 타입 별로 분류할 때 문자열로 분류한 것은?    1) df = df.select\_dtypes(include="object")  2) df.info() -> df.drop(columns= ['브랜드평판지수'],axis=1)  3) df= df.iloc[:,:7]  4) df.loc[:,'이름':'혈액형']  정답 : 1 |
| **교육생 사진 (교육생 모습 + 교육 화면)** |  |
| **화면 캡쳐**  **(교육 화면 + 교육일시)** |  |