22.02.04

Pandas(Series + DataFrame)

* 데이터 수가 많아 몇 번째 인지는 모르고 일정한 시간에 대한 정보를 가져오고 싶어서 … 시계열 데이터 (numpy는 정수 인덱스를 이용해서 찾기 때문에 어려움)
* numpy : 벡터 연산 🡪 모든 원소에 일괄 적용 🡪 블리언 인덱싱 🡪 조건 검색
* Pandas는 인덱스도 같이 출력 🡪 padas인덱스는 정수 인덱스 or 내가 만든 인덱스를 지정할 수 있음 🡪 정수 인덱스 : 여러 개, 슬라이싱 내가 만든 인덱스도 가능
* pandas는 1차원 데이터 : Series, 2차원 데이터 : DataFrame 3차원 데이터 : Panel 대문자 주의!!
* 변수.index : 인덱스만 출력 / 변수.values : 값들만 출력
* TF(TensorFlow)는 기본적으로 numpy의 Array 기반 / Scikit 기본적으로 numpy의 Array 기반 🡪 pandas로도 처리가 자연히 되도록
* 전체 데이터 중에서 값이 누락된 것들만 조건 검색 : 변수[변수.isnull( )]
* 전체 데이터 중에서 값이 누락이 없이 잘 들어 있는 것들만 조건 검색 : 변수[변수.notnull( )]
* FM적인 방법 : 접근하고자 하는 값이 1개 :  at(내가 만든 인덱스), iat(정수 인덱스) 접근하고자 하는 값이 여러 개 : loc(내가 만든 인덱스), iloc(정수 인덱스) EX) 변수.at[“APPLE”] 변수.iat[0] / 변수.loc[“APPLE” : “SM”] 변수.iloc[0 : 3]
* 이중 인덱스 가능함 🡪 인덱스 추출해보면 type달라짐 주의해야함
* df[“없는 컬럼”] = 1 : 없는 컬럼은 만들어서 추가됨
* df[“있는 컬럼”] = 0 : 있는 컬럼은 안에 원소 0으로 바뀜
* df [“없는 컬럼“] = df[“있는 컬럼“] \*100 도 가능
* del df[“ “] : 다이렉트로 컬럼 지우기
* df.drop(“컬럼” , axis = 1) : 원본은 건들이지 않으면서 컬럼 지우기 🡪 axis = 1 : 행 / df.drop(“인덱스”, axis = 0) 열 지우기 🡪 axis = 0 : 열
* 컬럼 이름 변경 : df.rename(columns = {“컬럼 이름1” : “변경하고 싶은 이름1”, “컬럼 이름2” : “변경하고 싶은 이름2”,… }, inplace = True) (inplace = True : 따로 재할당 하지 않아도 됨 변수에 바로 변경 가능)
* 인덱싱 : df.iat[0,3] df.at[“2022-01”,”가격”] df.iloc[0:2,0:4] df.loc[[“2022-01”,”2022-05”], : ]
* 인덱싱할 때 슬라이싱, 원하는 것만 가능한데 원하는 것만 컴마를 이용해서 쓰려면 리스트로 묶어야 함
* df.T : 트렌스포즈 가로 세로 축 뒤집음
* df.shape : (행, 열) 개수
* df.reindex([ “있는 컬럼”, “없는 컬럼” ], fill\_value = 10) : 있는 컬럼은 안에 원소 넣어서 나오고 없는 컬럼은 NaN으로 빈값으로 나옴 🡪 NaN을 10으로 채우러면 fill\_value
* 혹은 빈값을 두 지점 간격 사이의 값으로 채우려면 🡪 df.iloc[ : ,0].interpolate( ) 맨위나 맨 아래 값은 한 지점 밖에 없기 때문에 값을 채울 수 없음
* 정렬 : df.sort\_index(axis = , ascending = ) 🡪 axis = 1: 행 , 0 : 열 / ascending = True : 오름차순 , False : 내림차순
* 정렬조건 여러 개 예제 EX) df.sort\_index(axis = 0, ascending=False).sort\_index(axis = 1, ascending = True)
* 인덱스(가로,세로)가 아니라 값 자체를 기준으로 정렬 : df.sort\_values(by = “ “ ) 🡪 조건이 여러 개라면 [“ “, “ “]리스트로 묶어주기
* df[“시가”].rank( ) : 순위 매기기 (동일하면 순위의 평균으로 나옴 🡪 rank(method = “max”) 써주면 1,2,3,4,5 ~로 나옴) / ascending = False 적어주면 내림차순
* 줄 단위로 일괄 처리 : apply, lambda 이용 EX) 날짜별로 날짜의 여러 값중에서 max – min의 차이를 알고 싶을 때 🡪 DF.apply(lambda x : x.max() - x.min(), axis = 1)
* df.mean( ) :평균 / df.describe( ) : 통계적 수치들 / df.corr( ) : 상관계수 / df["시가"].unique( ) : 유니크한 값들 확인
* Nan 데이터 처리 🡪 제거 (dropna) or 채우기(fillna , apply, lambda)
* df.dropna(how = “all”, axis = 0) #전체 다 Nan로 데이터가 없는 열만 제거 (없는 값 기준)
* df.dropna(thresh = 2, axis = 0) # 줄에 데이터 중에서 정상적인 값이 2개 이상인 것들을 살릴 때 (정상적인 값이 기준)
* df.fillna(0) : Nan을 0으로 채움 🡪 컬럼마다 다른 값으로 채울 때 EX) df.fillna({ 컬럼 : 바꿀 수 ,컬럼 : 바꿀 수 …})