<https://3months.tistory.com/292>

R의 dataframe 데이터 타입을 참고하여 만든 것이 바로 pandas dataframe이다. pandas는 dataframe을 주로 다루기 위한 라이브러리이며, dataframe을 자유롭게 가공하는 것은 데이터 분석에서 주요한 작업이다. Pandas Dataframe은 테이블 형식의 데이터 (tabular, rectangular grid 등으로 불림)를 처리할 떄 사용한다. pandas dataframe의 주요한 구성요소는 컬럼, 데이터(로우), 인덱스이다.

pandas dataframe은 다양한 데이터 타입으로부터 만들 수 있다. ndarray, dictionary, dataframe, series, list 등이 데이터타입으로 사용될 수 있다.  
  
1. 데이터프레임 생성하기

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

columns = ['name', 'age', 'gender', 'job']

user1 = pd.DataFrame([['alice', 19, "F", "student"],

['john', 26, "M", "student"]],

columns=columns)

df.shape를 통해 dataframe의 row와 column 수를 알 수 있다. .index를 통해 index를 알 수 있으며, len을 통해 dataframe의 길이(row의 갯수)를 알 수 있다.

# Use the `shape` property

print(user1.shape)

(2, 4)

# Or use the `len()` function with the `index` property

print(len(user1.index))

2

print(user1.columns)

Index(['name', 'age', 'gender', 'job'], dtype='object')

user2 = pd.DataFrame([['eric', 22, "M", "student"],

['paul', 58, "F", "manager"]],

columns=columns)

user3 = pd.DataFrame(dict(name=['peter', 'julie'],

age=[33, 44], gender=['M', 'F'],

job=['engineer', 'scientist']))

print(user3)

Kangs-MacBook-Air:~ kangyunhi$ python3 test\_df.py

name age gender job

0 peter 33 M engineer

1 julie 44 F scientist

2. Dataframe에서 특정 컬럼이나 로우(인덱스) 선택하기  
  
 Dataframe에서 특정 컬럼이나 로우를 선택하여 그 값을 보는 것이다. pandas에서는 상당히 다양한 방법들이 있다. 대표적으로 iloc, loc, ix 등을 통해서 할 수 있다. column을 조회할때 df['column'], row를 조회할 때는 df.ix[index]를 많이 사용한다.

# Use `iloc[]` to select a row

print(user1.iloc[0])

print (user1.loc[0])

print (user1.ix[0])

3. Dataframe에 컬럼, 로우, 인덱스 추가하기

pandas는 기본적으로 row에 인덱스를 0부터 차례대로 자연수를 부여한다. 이를 변경하는 방법은 set\_index 함수를 이용하는 것이다. 아래의 df.set\_index('A')는 A 컬럼을 인덱스로 지정하는 것을 뜻한다. 그러면 3개의 row에 대하여 인덱스가 1,4,7이 부여된다.

**Append를 이용해 로우 추가하기**

인덱스를 신경쓰지 않고 그냥 데이터의 가장 뒤에 row를 추가하고 싶을 수도 있다. 이 경우에는 append를 사용하면 좋다. 아래는 df 데이터프레임에 a를 추가하여 row를 추가하는 코드를 보여준다. row를 추가한 후에 reset index를 통해 index를 0부터 새롭게 지정한다. 이는 실제 작업할 때 많이 쓰는 테크닉이다. 

user2 = pd.DataFrame([['eric', 22, "M", "student"],

['paul', 58, "F", "manager"]],

columns=columns)

user1 = user1.append(user2)

user1 = user1.reset\_index(drop=True)

print(user1)

name age gender job

0 alice 19 F student

1 john 26 M student

2 eric 22 M student

3 paul 58 F manager

**4. Dataframe의 인덱스, 컬럼, 데이터 삭제하기**

**인덱스 삭제**

인덱스를 지워야할 경우는 그렇게 많지 않을 것이다. 주로 reset\_index를 이용해서 index를 리셋하는 것을 많이 사용한다. 혹은 index의 이름을 삭제하고 싶다면 del df.index.name을 통해 인덱스의 이름을 삭제할 수 있다.

**컬럼 삭제**

drop 명령어를 통해 컬럼 전체를 삭제할 수 있다. axis=1은 컬럼을 뜻한다. axis=0인 경우, 로우를 삭제하며 이것이 디폴트이다. inplace의 경우 drop한 후의 데이터프레임으로 기존 데이터프레임을 대체하겠다는 뜻이다. 즉, 아래의 inplace=True는 df = df.drop('A', axis=1)과 같다.

user1.drop('name', axis=1, inplace=True)

print(user1)

age gender job

0 19 F student

1 26 M student

2 22 M student

3 58 F manager