데이터 전처리를 위한

Pandas (II)

박성호 (neowizard2018@gmail.com)

Contents

csv 파일로부터 데이터프레임 생성 pd.read_csv(...)

```
import pandas as pd

df = pd.read_csv('./test_dataframe_with_header_without_index.csv')

df
```

	Name	Country	Age	Job
0	John	USA	31	Student
1	Sabre	France	33	Lawyer
2	Kim	Korea	28	Developer
3	Sato	Japan	40	Chef
4	Lee	Korea	36	Professor
5	Smith	USA	55	CEO
6	David	USA	48	Banker

1. 열(column) 데이터 추출하기

➢ 데이터프레임(DataFrame)에서 열(column) 단위 데이터를 추출하기 위해서는 대괄호 안에 열 이름을 사용함

```
df_job = df[ 'Job' ]

df_job
```

← Series

```
0       Student
1        Lawyer
2     Developer
3          Chef
4     Professor
5           CEO
6        Banker
Name: Job, dtype: object
```

```
df_job = df[ ['Job'] ]

df_job
```

← DataFrame

	Job
0	Student
1	Lawyer
2	Developer
3	Chef
4	Professor
5	CEO
6	Banker

```
df_country_job = df[ ['Country', 'Job'] ]

df_country_job
```

	Country	Job
0	USA	Student
1	France	Lawyer
2	Korea	Developer
3	Japan	Chef
4	Korea	Professor
5	USA	CEO
6	USA	Banker

```
cols = [ 'Country', 'Job' ]

df_country_job = df[ cols ]

df_country_job
```

	Country	Job
0	USA	Student
1	France	Lawyer
2	Korea	Developer
3	Japan	Chef
4	Korea	Professor
5	USA	CEO
6	USA	Banker

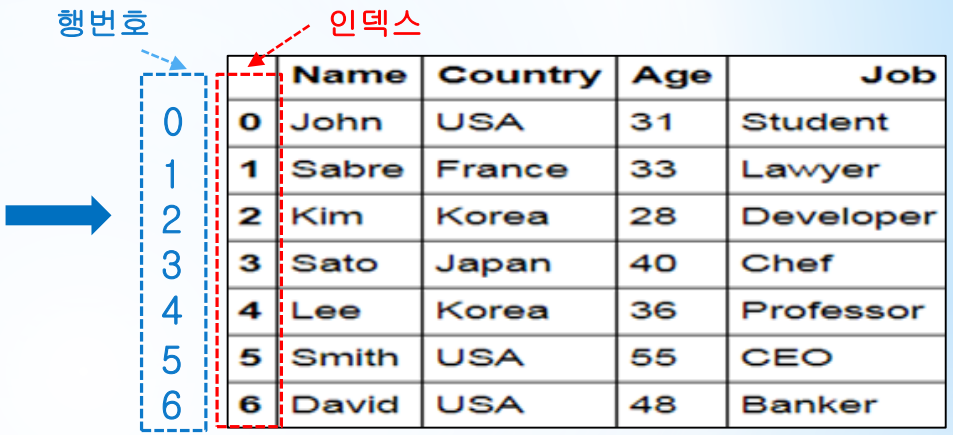
2. 인덱스, 행번호 개념

- 판다스에서는 df.loc[인덱스], df.iloc[행번호] 사용하여 행 단위로 데이터를 가져옴. 초보자라면 조건 지정이 용이한 df.loc[인덱스] 사용법부터 학습하는 것이 좋을것으로 판단됨

loc	인덱스 기준으로 행 데이터 읽기
iloc	행 번호를 기준으로 행 데이터 읽기

현재는 **인덱스**가 **행번호** 처럼 보이지만, 사실 인덱스는 문자열이나 임의의 숫자를 지정해도 무방함

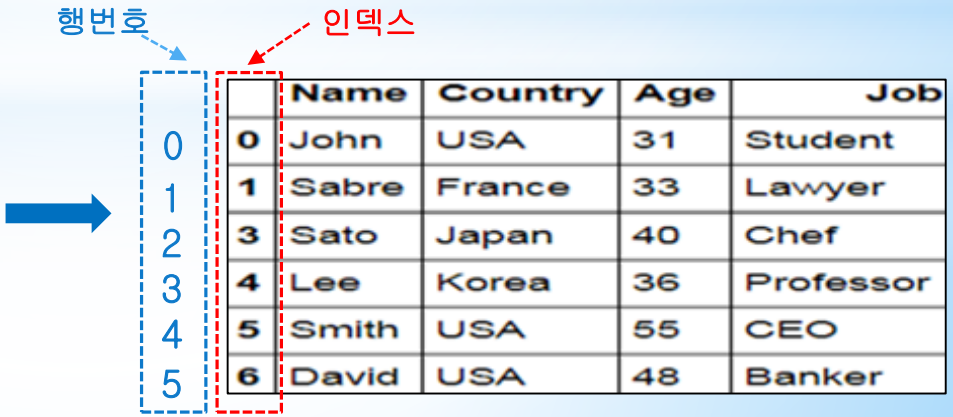
인덱스는 보통 0 부터 시작하지만 행 데이터를 추가, 삭제하면 언제든 변할 수 있음.



	Name	Country	Age	Job
0	John	USA	31	Student
1	Sabre	France	33	Lawyer
2	Kim	Korea	28	Developer
3	Sato	Japan	40	Chef
4	Lee	Korea	36	Professor
5	Smith	USA	55	CEO
6	David	USA	48	Banker

df.drop([2]) 명령을 통해 보기와 같이 2번 인덱스를 삭제하면,

행번호는 원래와 같이 0부터 시작해서 순서대로 이어지지만, **인덱스**는 연속적인 순서가 아닌 것을 확인 할 수 있음



	Name	Country	Age	Job
0	John	USA	31	Student
1	Sabre	France	33	Lawyer
3	Sato	Japan	40	Chef
4	Lee	Korea	36	Professor
5	Smith	USA	55	CEO
6	David	USA	48	Banker

2.1 df.loc[] 이용하여 행(row) 데이터 추출

	Name	Country	Age	Job
0	John	USA	31	Student
1	Sabre	France	33	Lawyer
2	Kim	Korea	28	Developer
3	Sato	Japan	40	Chef
4	Lee	Korea	36	Professor
5	Smith	USA	55	CEO
6	David	USA	48	Banker

[Self Study]

df.loc[0] 데이터 타입
df.loc[[0]] 데이터 타입
df.loc[0, :] 결과
df.loc[[0], :] 결과
df.loc[[0, 3], :] 결과
df.loc[0:3, :] 결과

```
df_1st_row = df.loc[ [0] ]

df_1st_row
```

	Name	Country	Age	Job
0	John	USA	31	Student

```
df_1st_4th_row = df.loc[ [0, 3] ]

df_1st_4th_row
```

	Name	Country	Age	Job
0	John	USA	31	Student
3	Sato	Japan	40	Chef

```
df_slice = df.loc[ 0:3 ]

df_slice
```

	Name	Country	Age	Job
0	John	USA	31	Student
1	Sabre	France	33	Lawyer
2	Kim	Korea	28	Developer
3	Sato	Japan	40	Chef

loc 속성의 슬라이싱은 일반적인 슬라이싱과는 다르므로 주의 필요.
즉 [0:3] 인덱스 0~2 행 추출이 아닌 0~3 까지의 행을 추출함

2.1 df.loc[] 이용하여 조건에 맞는 행(row) 데이터 추출

	Name	Country	Age	Job
0	John	USA	31	Student
1	Sabre	France	33	Lawyer
2	Kim	Korea	28	Developer
3	Sato	Japan	40	Chef
4	Lee	Korea	36	Professor
5	Smith	USA	55	CEO
6	David	USA	48	Banker

[Self Study]

df.loc[df['Age']>30, :]

df.loc[df['Age']>30, ['Job']]

df.loc[(df['Age']>30) & (df['Job']=='Chef')]

df.loc[(df['Age']>30) | (df['Job']=='Chef')]

```
df.loc[df['Country']=='USA']
```

	Name	Country	Age	Job
0	John	USA	31	Student
5	Smith	USA	55	CEO
6	David	USA	48	Banker

```
df.loc[df['Age']>30]
```

	Name	Country	Age	Job
0	John	USA	31	Student
1	Sabre	France	33	Lawyer
3	Sato	Japan	40	Chef
4	Lee	Korea	36	Professor
5	Smith	USA	55	CEO
6	David	USA	48	Banker

3. df.loc[] 이용한 행과 열 데이터 동시 추출

	Name	Country	Age	Job
0	John	USA	31	Student
1	Sabre	France	33	Lawyer
2	Kim	Korea	28	Developer
3	Sato	Japan	40	Chef
4	Lee	Korea	36	Professor
5	Smith	USA	55	CEO
6	David	USA	48	Banker

[Self Study]

```
for index in range(len(df)):

    print(type(df.loc[ index, ['Name' ,'Country']]))
    print(df.loc[ index, ['Name' ,'Country']])
    print('=====================================')
    print(type(df.loc[ [index], ['Name' ,'Country']]))
    print(df.loc[ [index], ['Name' ,'Country']])
    print('=====================================')
```

```
df.loc[ [1], : ]
```

	Name	Country	Age	Job
1	Sabre	France	33	Lawyer

```
df.loc[ [1, 3], ['Name', 'Job'] ]
```

	Name	Job
1	Sabre	Lawyer
3	Sato	Chef

```
df.loc[ :, : ]
```

	Name	Country	Age	Job
0	John	USA	31	Student
1	Sabre	France	33	Lawyer
2	Kim	Korea	28	Developer
3	Sato	Japan	40	Chef
4	Lee	Korea	36	Professor
5	Smith	USA	55	CEO
6	David	USA	48	Banker

4. 데이터프레임 행과 열 삭제 – df.drop()

4.1 행 삭제 df.drop(index, axis=0) # axis = 0 행, axis = 1 열

```
df = df.drop(1, axis=0)

df
```

	Name	Country	Age	Job
0	John	USA	31	Student
2	Kim	Korea	28	Developer
3	Sato	Japan	40	Chef
4	Lee	Korea	36	Professor
5	Smith	USA	55	CEO
6	David	USA	48	Banker

```
df = df.drop([3, 5], axis=0)

df
```

	Name	Country	Age	Job
0	John	USA	31	Student
2	Kim	Korea	28	Developer
4	Lee	Korea	36	Professor
6	David	USA	48	Banker

```
df = df.reset_index()

df
```

	index	Name	Country	Age	Job
0	0	John	USA	31	Student
1	2	Kim	Korea	28	Developer
2	4	Lee	Korea	36	Professor
3	6	David	USA	48	Banker

[Self Study]

데이터 프레임 df 생성 후,

df.drop(1, axis=0, inplace=True)

df.drop([3, 5], axis=0, inplace=True)

df.reset_index(inplace=True)

4. 데이터프레임 행과 열 삭제 – df.drop()

4.2 열 삭제 df.drop(column name, axis=1) # axis = 0 행, axis = 1 열

```
df = df.drop('Age', axis=1)
df
```

	Name	Country	Job
0	John	USA	Student
1	Sabre	France	Lawyer
2	Kim	Korea	Developer
3	Sato	Japan	Chef
4	Lee	Korea	Professor
5	Smith	USA	CEO
6	David	USA	Banker

```
df = df.drop(['Name', 'Job'], axis=1)
df
```

	Country
0	USA
1	France
2	Korea
3	Japan
4	Korea
5	USA
6	USA

[Self Study]

데이터 프레임 df 생성 후,

df.drop('Age', axis=1, inplace=True)

df.drop(['Name', 'Job'], axis=1, inplace=True)

5. 데이터프레임 행과 열 추가

5.1 행 추가 df.append(dict_new_data, ignore_index=True)

```
new_data = { 'Name' : 'Park',
             'Country' : 'Korea',
             'Age' : 36,
             'Job' : 'Chef'
           }

df = df.append(new_data, ignore_index=True)

df
```

	Name	Country	Age	Job
0	John	USA	31	Student
1	Sabre	France	33	Lawyer
2	Kim	Korea	28	Developer
3	Sato	Japan	40	Chef
4	Lee	Korea	36	Professor
5	Smith	USA	55	CEO
6	David	USA	48	Banker
7	Park	Korea	36	Chef

```
new_data = { 'Name' : 'Koga',
             'Country' : 'Japan',
             'Age' : 26,
             'Job' : 'Player'
           }

df = df.append(new_data, ignore_index=True)

df
```

	Name	Country	Age	Job
0	John	USA	31	Student
1	Sabre	France	33	Lawyer
2	Kim	Korea	28	Developer
3	Sato	Japan	40	Chef
4	Lee	Korea	36	Professor
5	Smith	USA	55	CEO
6	David	USA	48	Banker
7	Park	Korea	36	Chef
8	Koga	Japan	26	Player

5. 데이터프레임 행과 열 추가

5.2 열 추가 df['column'], df.assign()

```
df['New_Col1'] = df['Age'] / 2.0

df
```

	Name	Country	Age	Job	New_Col1
0	John	USA	31	Student	15.5
1	Sabre	France	33	Lawyer	16.5
2	Kim	Korea	28	Developer	14.0
3	Sato	Japan	40	Chef	20.0
4	Lee	Korea	36	Professor	18.0
5	Smith	USA	55	CEO	27.5
6	David	USA	48	Banker	24.0

```
add_val_1 = df['Age'].values

add_val_2 = df['New_Col1'].values

df = df.assign(ADD_1=add_val_1, ADD_2=add_val_2)

df
```

	Name	Country	Age	Job	New_Col1	ADD_1	ADD_2
0	John	USA	31	Student	15.5	31	15.5
1	Sabre	France	33	Lawyer	16.5	33	16.5
2	Kim	Korea	28	Developer	14.0	28	14.0
3	Sato	Japan	40	Chef	20.0	40	20.0
4	Lee	Korea	36	Professor	18.0	36	18.0
5	Smith	USA	55	CEO	27.5	55	27.5
6	David	USA	48	Banker	24.0	48	24.0

6. 데이터프레임 합치기 pd.concat()

6.1 위 아래 방향으로 합치기 pd.concat([df1, df2], axis=0, ignore_index=True)

```
data1 = { 'Name' : ['John', 'Sabre'],
          'Country' : ['USA', 'France'],
          'Age' : [31, 33],
          'Job' : ['Student', 'Lawyer']
        }

df1 = pd.DataFrame(data1)

df1
```

	Name	Country	Age	Job
0	John	USA	31	Student
1	Sabre	France	33	Lawyer

```
data2 = { 'Name' : ['Lee', 'Smith'],
          'Country' : ['Korea', 'USA'],
          'Age' : [36, 55],
          'Job' : ['Professor', 'CEO']
        }

df2 = pd.DataFrame(data2)

df2
```

	Name	Country	Age	Job
0	Lee	Korea	36	Professor
1	Smith	USA	55	CEO

6. 데이터프레임 합치기 pd.concat()

6.1 위 아래 방향으로 합치기 pd.concat([df1, df2], axis=0, ignore_index=True)

```
df3 = pd.concat([df1, df2], axis=0)

df3
```

	Name	Country	Age	Job
0	John	USA	31	Student
1	Sabre	France	33	Lawyer
0	Lee	Korea	36	Professor
1	Smith	USA	55	CEO

```
df4 = pd.concat([df1, df2], axis=0, ignore_index=True)

df4
```

	Name	Country	Age	Job
0	John	USA	31	Student
1	Sabre	France	33	Lawyer
2	Lee	Korea	36	Professor
3	Smith	USA	55	CEO

6. 데이터프레임 합치기 pd.concat()

6.2 좌우 방향으로 합치기 pd.concat([df1, df2], axis=1)

```
data1 = { 'Name' : ['John', 'Sabre'],
          'Country' : ['USA', 'France'],
          'Age' : [31, 33],
          'Job' : ['Student', 'Lawyer']
        }

df1 = pd.DataFrame(data1)

df1
```

	Name	Country	Age	Job
0	John	USA	31	Student
1	Sabre	France	33	Lawyer

```
data5 = { 'Salary' : 1000,
          'Hobby' : ['Run'],
        }

df5 = pd.DataFrame(data5)

df5
```

	Salary	Hobby
0	1000	Run

```
df6 = pd.concat([df1, df5], axis=1)

df6
```

	Name	Country	Age	Job	Salary	Hobby
0	John	USA	31	Student	1000.0	Run
1	Sabre	France	33	Lawyer	NaN	NaN

7. 데이터프레임 열 순서 변경 및 특정 열 제외

7.1 열 순서 변경

```
df
```

	Name	Country	Age	Job
0	John	USA	31	Student
1	Sabre	France	33	Lawyer
2	Kim	Korea	28	Developer
3	Sato	Japan	40	Chef
4	Lee	Korea	36	Professor
5	Smith	USA	55	CEO
6	David	USA	48	Banker

```
column_name = [ 'Age', 'Job', 'Country', 'Name' ]

df1 = df[column_name]

df1
```

	Age	Job	Country	Name
0	31	Student	USA	John
1	33	Lawyer	France	Sabre
2	28	Developer	Korea	Kim
3	40	Chef	Japan	Sato
4	36	Professor	Korea	Lee
5	55	CEO	USA	Smith
6	48	Banker	USA	David

7. 데이터프레임 열 순서 변경 및 특정 열 제외

7.2 특정 열 제외

```
df
```

	Name	Country	Age	Job
0	John	USA	31	Student
1	Sabre	France	33	Lawyer
2	Kim	Korea	28	Developer
3	Sato	Japan	40	Chef
4	Lee	Korea	36	Professor
5	Smith	USA	55	CEO
6	David	USA	48	Banker

```
df2 = df[df.columns.difference(['Age', 'Job'])]

df2
```

	Country	Name
0	USA	John
1	France	Sabre
2	Korea	Kim
3	Japan	Sato
4	Korea	Lee
5	USA	Smith
6	USA	David