K-Digital Training 스마트 팩토리 3기

Pandas

Pandas

Pandas 는 구조화된 데이터나 표 형식의 데이터를 빠르고 쉽게 다룰 수 있도록 하는 라이브러리

Pandas 에서는 Series와 DataFrame, 두 개의 데이터 오브젝트에 익숙해져야 함

쉽게 말하면, Series는 1차원 데이터, DataFrame은 2차원 데이터

Pandas - Series

Series는 List와 다르게 인덱스를 직접 지정할 수 있다.

List

인덱스	값
0	143
1	150
2	157
3	160

Series

인덱스	값
2018	143
2019	150
2020	157
2021	160

Pandas - Series

```
import pandas as pd

이름 = pd.Series(['데이터값1', '데이터값2', '데이터값3'.....], index=['인덱스명1', '인덱스명2', '인덱스명3'....])
```

```
growth = pd.Series([143, 150, 157, 160], index=["2018", "2019", "2020", "2021"])
growth
```

인덱스

```
2018    143
2019    150
2020    157
2021    160
dtype: int64
```

```
growth["2018"]
```

```
143
```

Pandas - Series

인덱스를 지정하지 않는다면? ➡ 리스트처럼 0, 1, 2 … 로 설정된다!

```
growth= pd.Series([143, 150, 157, 160])
growth
```

```
0    143
1    150
2    157
3    160
dtype: int64
```

Pandas - DataFrame

DataFrame은 아래와 같이 Series들을 결합해 놓은 형태이다!

Series

영희	
인덱스	값
2018	143
2019	150
2020	157
2021	160

+

Series

철수	
인덱스	값
2018	165
2019	172
2020	175
2021	180

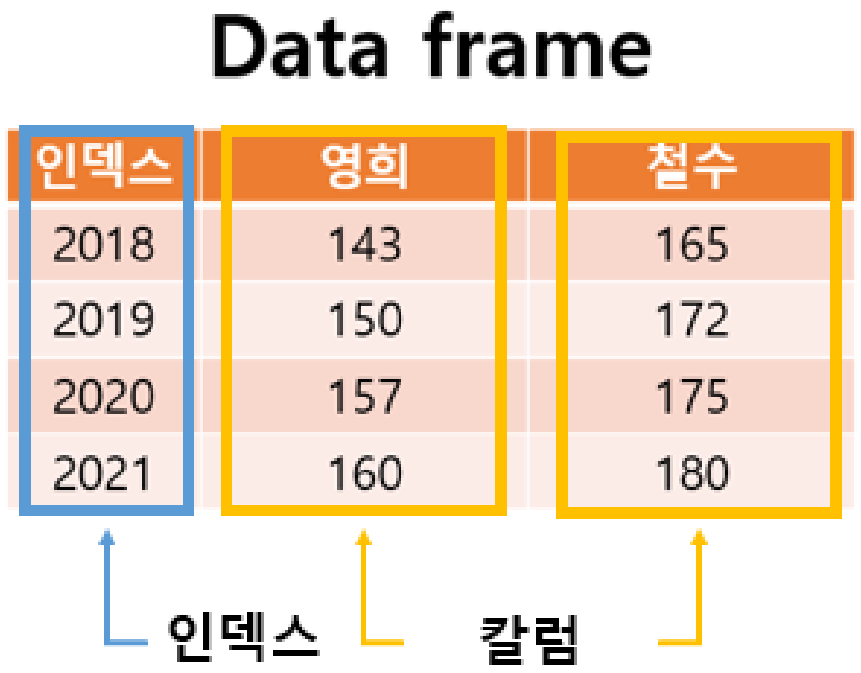
=

Data frame

인덱스	영희	철수
2018	143	165
2019	150	172
2020	157	175
2021	160	180

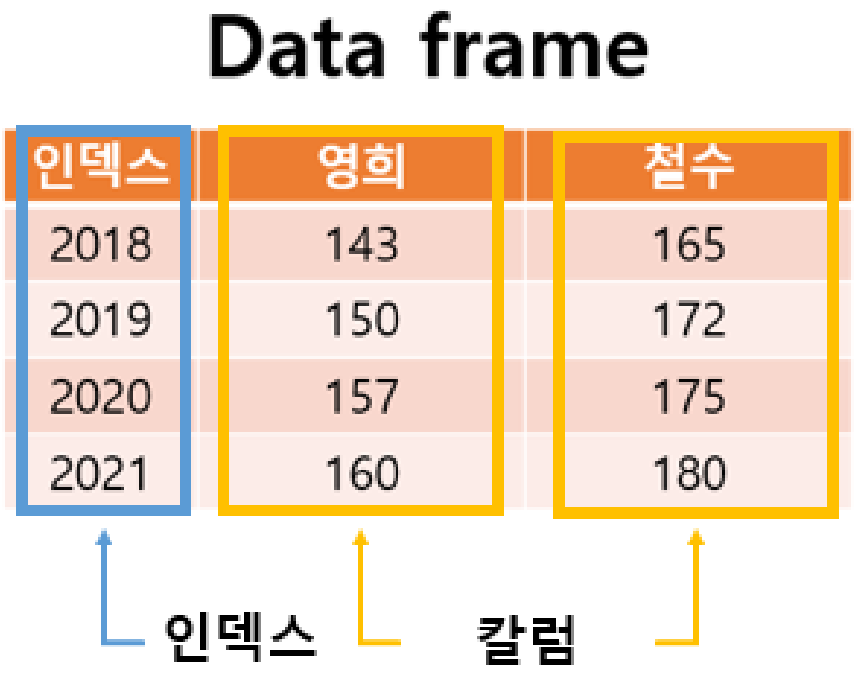
Pandas - DataFrame

DataFrame은 인덱스와 컬럼을 기준으로 표 형태처럼 데이터를 저장!



인덱스와 컬럼 2개를 기준으로
데이터가 형성됨!

Pandas - DataFrame




```
index = ['2018','2019','2020','2021']

Yeonghee = pd.Series([143,150,157,160], index=index)
Cheolsu = pd.Series([165,172,175,180], index=index)

growth = pd.DataFrame({
    '영희': Yeonghee,
    '철수': Cheolsu
})

growth
```

	영희	철수
2018	143	165
2019	150	172
2020	157	175
2021	160	180

Pandas - DataFrame

```
index = ['2018','2019','2020','2021']

data = {
    '영희': [143,150,157,160],
    '철수': [165,172,175,180]
}

growth = pd.DataFrame(data, columns=['영희', '철수'], index=index)

growth
```

	영희	철수
2018	143	165
2019	150	172
2020	157	175
2021	160	180

Pandas - DataFrame

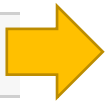
```
print(growth.dtypes)

growth.astype('float')
```

astype : 데이터 타입 변경

```
영희     int64
철수     int64
dtype: object
```

	영희	철수
2018	143.0	165.0
2019	150.0	172.0
2020	157.0	175.0
2021	160.0	180.0

```
growth.astype({'영희':'float'})
```



특정 컬럼에 대해서
데이터 타입 변경

	영희	철수
2018	143.0	165
2019	150.0	172
2020	157.0	175
2021	160.0	180

Pandas

https://www.kaggle.com/learn/pandas > Creating, Reading and Writing

```
wine_reviews = pd.read_csv("../input/wine-reviews/winemag-data-130k-v2.csv")
```

read_csv() : .csv 확장자 파일 읽기

```
wine_reviews.shape
```

.shape : 데이터 프레임 모양 확인 (행, 열)

```
pd.set_option("display.max_rows", 5)
```

None으로 설정할 경우, 제한 없이 출력

Pandas

https://www.kaggle.com/learn/pandas > **Indexing, Selecting & Assigning**

dataframe.컬럼명 : 데이터 프레임에서 해당 컬럼만 select
dataframe['컬럼명'] : 데이터 프레임에서 해당 컬럼만 select

.iloc[] : index를 활용해 location을 지정하는 방법.
ex) dataframe.iloc[row_index, column_index]
(0:10) 이라고 지정했을때 마지막 10이 포함되지 않음

.loc[] : index 및 colum명을 통해 location을 지정하는 방법.
Ex) dataframe.loc[index_name, column_name]
(0:10) 이라고 지정했을때 마지막 10이 포함됨

Pandas

https://www.kaggle.com/learn/pandas > **Summary Functions and Maps**

dataframe.describe() : 데이터프레임 객체의 설명적 통계량을 출력.
제공된 자료에 따라 조금씩 다르게 출력됨.

[numeric 데이터]
Count : 총 데이터 수 / mean : 데이터의 평균 / std : 표준편차 / min : 최소값
max : 최대값 / 25%, 50% , 75% : 백분위수의 각 지점

[object 데이터]
Count : 총 데이터 수 / Unique : 중복없이 나오는 고유한 데이터 값
Top : 가장 값이 많은 데이터 (최빈값인 항목)
Freq : 최빈 데이터의 실제 수 (Top의 개수, 최빈값)

Pandas

series.map() : 데이터 변경
시리즈 자료형(즉, 1차원) 에서 사용

* lamda : 익명함수 생성 키워드

```
def add(x, y):
    return x + y

add(2,3)
```

```
5
```

```
(lambda x,y: x + y)(2, 3)
```

```
review_points_mean = reviews.points.mean()
review_points_mean
```

```
88.44713820775404
```

```
reviews.points
```

```
0         87
1         87
          ..
129969    90
129970    90
Name: points, Length: 129971, dtype: int64
```

```
reviews.points.map(lambda p: p - review_points_mean)
```

```
0        -1.447138
1        -1.447138
             ...
129969    1.552862
129970    1.552862
Name: points, Length: 129971, dtype: float64
```

Pandas

series or dataframe.apply(함수, axis) :
시리즈 혹은 데이터프레임에서 모두 사용 가능

axis = ‘index’ 혹은 axis = 0 : row 방향으로 함수 적용
axis = ‘columns’ 혹은 axis = 1 : column 방향으로 함수 적용

Pandas

```
reviews.apply(lambda p: p.points - review_points_mean, axis='columns')
```

```
0        -1.447138
1        -1.447138
            ...
129969    1.552862
129970    1.552862
Length: 129971, dtype: float64
```

```
def remean_points(row):
    row.points = row.points - review_points_mean
    return row

reviews.apply(remean_points, axis='columns')
```

country	description	designation	points	price	province	region_1	region_2	taster_name	taster_twitter_
Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	-1.447138	NaN	Sicily & Sardinia	Etna	NaN	Kerin O'Keefe	@kerir

Pandas

https://www.kaggle.com/learn/pandas > Grouping and Sorting

dataframe.groupby(컬럼) : 컬럼을 기준으로 group을 지음.

dataframe.reset_index() : 인덱스를 기본 인덱스인 0, 1, 2 … 로 변경.

dataframe.sort_values(by=컬럼명, ascending=bool) :
컬럼명을 기준으로 정렬. ascending이 False일 경우 내림차순 정렬.
(default: True)

Pandas

- dataframe.dtypes : 전체 컬럼의 데이터 타입 확인
- dataframe.column.dtype : 컬럼 하나의 데이터 타입 확인
- dataframe.column.astype() : 데이터 타입 변경

```
reviews.points.dtype
```

```
dtype('int64')
```

```
reviews.points.astype('float64')
```

```
0         87.0
1         87.0
          ...
129969    90.0
129970    90.0
Name: points, Length: 129971, dtype: float64
```

Pandas

- pd.isnull(column) : NaN 데이터 확인
- pd.notnull(column) : NaN이 아닌 데이터 확인
- dataframe.column.fillna(대체 값) : NaN 대체 값으로 변경
- dataframe.column.replace(기존 값, 새로운 값) : 기존 값 새로운 값으로 대치

Pandas

- 컬럼명 변경
  - dataframe.rename(columns={‘기존 컬럼명’: ‘새로운 컬럼명’})
- 인덱스 변경
  - dataframe.rename(index={0: ‘인덱스 명’})

```
reviews.rename(columns={'points': 'score'})
```

	country	description	designation	score	price
0	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87	NaN
1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	87	15.0
...	...	...	...	...	...
129969	France	A dry style of Pinot Gris, this is crisp with ...	NaN	90	32.0
129970	France	Big, rich and off-dry, this is powered by inte...	Lieu-dit Harth Cuvée Caroline	90	21.0

```
reviews.rename(index={0: 'firstEntry', 1: 'secondEntry'})
```

	country	description	designation	points	price	province
firstEntry	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87	NaN	Sicily & Sardinia
secondEntry	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	87	15.0	Douro
...	...	...	...	...	...	...
129969	France	A dry style of Pinot Gris, this is crisp with ...	NaN	90	32.0	Alsace
129970	France	Big, rich and off-dry, this is powered by inte...	Lieu-dit Harth Cuvée Caroline	90	21.0	Alsace

Pandas

- pd.concat([dataframe1, dataframe2])
  - column 이 동일할 때
- join()
  - 공통의 index를 가지고 있을 때

```
left = canadian_youtube.set_index(['title', 'trending_date'])
right = british_youtube.set_index(['title', 'trending_date'])

left.join(right, lsuffix='_CAN', rsuffix='_UK', how='inner')
```