A23 Pandas

텍스트 기반의 데이터 형태 읽어서 분석하기
(정형화된 데이터 분석)

~~Series~~ (1차원) → DataFrame (2차원)

pip install pandas.

import pandas. as pd.

- Series (데이터 타입 다를 수 있음).

pd. Series ([1, 2, 3] , index=['A', ...])

↑ 축가 가능.

(dict 1)

↑ 디셔너리.

- DataFrame

x = ['x', 5 , True]

y

df = pd.DataFrame ([x, y])

pd.DataFrae({'x': x, ...})

{디셔너리}: 열 이름으로 지정

행이 아니라.

pd ([x, y], index = [ ]

, columns = [ ])

• df 접근

df['NAME'] ⟶ 'NAME'열

df.loc['x'] ⟶ 'x' 행

여러행 (df[['NAME', 'VALUE']]
df.loc[['x', 'x']]

df.loc[0:1] 다르게! 1행도 포함

df.loc[행, 열]

↑ 1개일수도, 여러개일수도

번호로.

df.iloc[행번호, 열번호]

↑ 여러행 슬라이싱과 동일

0:1

↑ 1개의.

조건부선택

df[(df['VALUE']>6) 여러, or !
df['COND']]

A24. Pandas 2

- CSV.

read_csv (파일이름, encoding = '인코딩')

r
w

to_csv

여러 파일 형식 지원

- 통계 함수

count, min, max, sum, mean, median, ..., var

- 정렬 함수

df.sort_values(by=["","",""], ascending=[True, False])

(1개 이상 List)

- groupby

data.groupby("type").sum()

(2개 이상 List)

· 결측값

dropna() → 행 지움.

fillna(3) ← 위의값: method=ffill, bfill ← 아래값.
(.isna() / .notna()).

평균 또는 다른 값으로 채움

· 데이터 추가

df.loc[index]

df.append(df)

~~df~~ pd.concat([df1, df2])

len(df) → 3 , 3개행 존재.

df.loc[3] = ['d', 4, False]

└ 마지막에 추가.

df.loc[len(df)] = ~~데이터프레임~~

df = df.append(df1, ignore_index=True)

└ 동일한 열 가지는 df.

pd.concat([df, df2], axis=0)

⇒ 열 이름 다르면 행 추가됨

1 → 달라도 행 추가 없이 열 추가.