python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

day4

보다 쉽게 데이터 다루기 (pandas, numpy)

2019 경복고등학교

[워밍업 문제]

'data/scientist.csv'파일을 불러와 다음의 문제를 해결해 보세요.

1) 과학자들의 평균 나이(Age)는 얼마일까요?

2) 직업(Occupation)이 화학자(Chemist)인 과학자는 누구일까요?

#1. numpy 라이브러리

#2. pandas 라이브러리

#3. [project] 서울 청소년의 스트레스 데이터 분석하기

#1. numpy 라이브러리

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[numpy 라이브러리]

- 숫자 데이터를 효과적으로 다룰 수 있게 돕는 라이브러리
- 배열을 다루는 데 유용함
- **import numpy as np** 로 np라는 별칭으로 주로 사용
- https://numpy.org/

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[python 기본 코드 Vs. numpy 모듈 사용 코드]

- 0~50까지의 데이터 중 짝수만 발생시켜서 모두 저장하기

```
1 data=[]
2 for i in range(0,51,2) :
3     data.append(i)
4 print(data)
5 print(type(data))
```

```
<class 'list'>
```

```
1 import numpy as np
2 arr=np.arange(0,51,2)
3 print(data)
4 print(type(arr))
```

```
<class 'numpy.ndarray'>
```

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[python 기본 코드 Vs. numpy 모듈 사용 코드]

- 0~50까지의 데이터 중 짝수만 발생시켜서 모두 저장하기

```python
import matplotlib.pyplot as plt
import random
dice=[]
for i in range(10) :
    dice.append(random.randint(1,6))
print(dice)
plt.hist(dice,rwidth=0.9,bins=6)
plt.xlim(1,6)
plt.show()
```

```python
import matplotlib.pyplot as plt
import numpy as np
dice=np.random.choice(range(1,7),10)
print(dice)
plt.hist(dice,rwidth=0.9,bins=6)
plt.xlim(1,6)
plt.show()
```

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[numpy 배열 다루어 보기]

- 1차원 배열

```
import numpy as np
arr1=np.array([10,20,30,40]) #배열생성
arr2=np.array([1,2,3,4])
print(arr1+arr2)
print(arr1-arr2)
print(arr1**arr2)

arr3=np.arange(5)#0~4까지 데이터 발생
print(arr3)
print("합계 : ", arr3.sum(),"평균 : ",arr3.mean())
arr4=np.arange(1,5)#1~4까지 데이터 발생
print(arr4)
print(arr4.cumsum()) #원소들의 누적합
print(arr4.cumprod()) #원소들의 누적곱
```

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[numpy 배열 다루어 보기]

- 다차원 배열

```
import numpy as np
arr1=np.array([[1,2,3],[4,5,6],[7,8,9]]) #3행 3열 배열
print(arr1)
arr2=np.arange(12).reshape(4,3) #0~12까지의 데이터 생성, 4행 3열 배열
print(arr2)
arr3=np.zeros(10) #10개의 원소가 모두 0인 배열
print(arr3)
arr4=np.ones((2,5)) #2행 5열의 원소가 모두 1인 배열
print(arr4)
```

#2. pandas 라이브러리

(Python Data Analysis Library)

[pandas]

- 테이블 형태의 데이터를 쉽게 다룰 수 있게 돕는 라이브러리
- import pandas as pd로 pd라는 별칭으로 주로 사용
- numpy기반으로 만들어졌지만 좀 더 복잡한 연산에 특화
- pandas의 그래프 : matplotlib.pyplot기반
- https://pandas.pydata.org

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[pandas 2가지 데이터 타입]

- Series

index	values

- DataFrame

		columns			
index		values			

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[pandas의 Series]

index	values

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[pandas Series : 인덱스(레이블)을 가지는 1차원 데이터]

```
import pandas as pd
ser=pd.Series([10,20,30])
print(ser)
print('index=',ser.index)
print('values=',ser.values)
```

	index	value
	0	10
	1	20
	2	30

```
0    10
1    20
2    30
dtype: int64
index= RangeIndex(start=0, stop=3, step=1)
values= [10 20 30]
```

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[pandas Series : 인덱스(레이블)을 가지는 1차원 데이터]

```
import pandas as pd
ser=pd.Series([10,20,30,40],['a','b','c','d'])
print(ser)
print('index=',ser.index)
print('values=',ser.values)
```

```
a    10
b    20
c    30
d    40
dtype: int64
index= Index(['a', 'b', 'c', 'd'], dtype='object')
values= [10 20 30 40]
```

index	value
a	10
b	20
c	30
d	40

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[pandas Series의 자주 사용하는 메서드]

Series method(시리즈 메서드)	설명
describe	요약 통계량 계산
isin	시리즈에 포함된 값이 있는지 확인
min	최소값 반환
max	최대값 반환
mean	산술 평균 반환
median	중간값 반환
sorted_values	특정 값을 기준으로 정렬
sorted_index	인덱스를 기준으로 정렬

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[pandas Series 다루어 보기]

'data/scientist.csv'

Name	Born	Died	Age	Occupation
Rosaline Franklin	1920-07-25	1958-04-16	37	Chemist
William Gosset	1876-06-13	1937-10-16	61	Statistician
Florence Nightingale	1820-05-12	1910-08-13	90	Nurse
Marie Curie	1867-11-07	1934-07-04	66	Chemist
Rachel Carson	1907-05-27	1964-04-14	56	Biologist
John Snow	1813-03-15	1858-06-16	45	Physician
Alan Turing	1912-06-23	1954-06-07	41	Computer Scientist
Johann Gauss	1777-04-30	1855-02-23	77	Mathematician

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[pandas Series 다루어 보기]

1) 파일 읽기 : pd.read_csv('파일명')

```
import pandas as pd
scientists=pd.read_csv('./data/scientists.csv',sep=",") #파일 읽기
print(scientists)
print(type(scientists))
```

```
                Name        Born        Died  Age          Occupation
0    Rosaline Franklin  1920-07-25  1958-04-16   37             Chemist
1       William Gosset  1876-06-13  1937-10-16   61        Statistician
2  Florence Nightingale  1820-05-12  1910-08-13   90               Nurse
3         Marie Curie  1867-11-07  1934-07-04   66             Chemist
4       Rachel Carson  1907-05-27  1964-04-14   56           Biologist
5           John Snow  1813-03-15  1858-06-16   45           Physician
6         Alan Turing  1912-06-23  1954-06-07   41  Computer Scientist
7        Johann Gauss  1777-04-30  1855-02-23   77       Mathematician
<class 'pandas.core.frame.DataFrame'>
```

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[pandas Series 다루어 보기]

2) 데이터 프레임에서 행단위 데이터 추출하기 : 데이터프레임명.iloc[행번호]

```
import pandas as pd
scientists=pd.read_csv('./data/scientists.csv',sep=",")
print(scientists)
rowData=scientists.iloc[3] #행번호가 3인 데이터 추출
print(type(rowData))
print(rowData)
```

```
                Name        Born        Died  Age          Occupation
0    Rosaline Franklin  1920-07-25  1958-04-16   37           Chemist
1      William Gosset  1876-06-13  1937-10-16   61       Statistician
2  Florence Nightingale  1820-05-12  1910-08-13   90             Nurse
3         Marie Curie  1867-11-07  1934-07-04   66           Chemist
4       Rachel Carson  1907-05-27  1964-04-14   56          Biologist
5          John Snow  1813-03-15  1858-06-16   45          Physician
6        Alan Turing  1912-06-23  1954-06-07   41  Computer Scientist
7       Johann Gauss  1777-04-30  1855-02-23   77      Mathematician
<class 'pandas.core.series.Series'>
Name           Marie Curie
Born            1867-11-07
Died            1934-07-04
Age                     66
Occupation         Chemist
Name: 3, dtype: object
```

[pandas Series 다루어 보기]

2) 데이터 프레임에서 행단위 데이터 추출하기 : 데이터프레임명. loc['인덱스명']

```
import pandas as pd
scientists=pd.read_csv('./data/scientists.csv',sep=",")
print(scientists)
scientists=scientists.set_index('Name') #index를 'Name'로 변경
print(scientists)
rowData=scientists.loc['John Snow'] #index가 'John'인 데이터 추출
print(type(rowData))
print(rowData)
```

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[pandas Series 다루어 보기]

3) 데이터 프레임에서 열 단위 데이터 추출하기 : 데이터프레임명['컬럼명']

```
import pandas as pd
scientists=pd.read_csv('./data/scientists.csv',sep=",")
colData=scientists['Occupation'] #컬럼이름이 'Occupation'인 열추출
print(type(colData))
print(colData)
```

```
<class 'pandas.core.series.Series'>
0               Chemist
1         Statistician
2                Nurse
3               Chemist
4             Biologist
5             Physician
6    Computer Scientist
7          Mathematician
Name: Occupation, dtype: object
```

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[pandas Series 다루어 보기]

4) Series의 통계 메서드 다뤄보기('Age'컬럼 추출)

: Series에서 나이가 가장 큰 값, 가장 작은 값, 평균 값 출력

```
import pandas as pd
scientists=pd.read_csv('./data/scientists.csv')
ages=scientists['Age']
print(ages)
print(type(ages))
print("나이가 가장 큰 값 : ",ages.max())
print("나이가 가장 작은 값 : ",ages.min())
print("나이의 평균 값 : ",ages.mean())
```

[pandas Series 다루어 보기]

5) Series의 불린(Boolean) 추출 ('Age'컬럼 추출)

: Series에서 평균 나이보다 많은 나이 추출

```
import pandas as pd
scientists=pd.read_csv('./data/scientists.csv')
ages=scientists['Age']
print(ages[ages>ages.mean()])
```

```
1    61
2    90
3    66
7    77
Name: Age, dtype: int64
```

[pandas Series 다루어 보기]

6) Series의 브로드캐스팅(('Age'컬럼 추출)

: 나이를 모두 5살 많게 하기

```
import pandas as pd
scientists=pd.read_csv('./data/scientists.csv')
ages=scientists['Age']
print(ages)
print(ages+5)
print(ages)
```

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[pandas의 DataFrame]

		columns			
index		values			

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[pandas DataFrame구조 : 행과 열에 레이블을 가진 2차원 데이터]

```
import pandas as pd

df=pd.DataFrame(
    [[1,10,100],[2,20,200],[3,30,300]],
    index=['r1','r2','r3'],
    columns=['c1','c2','c3']
)
print(type(df))
print(df)
```

```
<class 'pandas.core.frame.DataFrame'>
    c1  c2   c3
r1   1  10  100
r2   2  20  200
r3   3  30  300
```

	c1	c2	c3
r1	1	10	100
r2	2	20	200
r3	3	30	300

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[pandas DataFrame 다루어 보기]

1) 파일 읽기 : pd.read_csv('파일명')

```
import pandas as pd
df=pd.read_csv('./data/gapminder.tsv',delimiter='\t')
print(type(df))
print(df.head()) #처음 5개의 데이터만 보기
print(df.shape) #데이터의 (행, 열)
```

```
<class 'pandas.core.frame.DataFrame'>
       country continent  year  lifeExp      pop   gdpPercap
0  Afghanistan      Asia  1952   28.801  8425333  779.445314
1  Afghanistan      Asia  1957   30.332  9240934  820.853030
2  Afghanistan      Asia  1962   31.997 10267083  853.100710
3  Afghanistan      Asia  1967   34.020 11537966  836.197138
4  Afghanistan      Asia  1972   36.088 13079460  739.981106
(1704, 6)
```

[pandas DataFrame 다루어 보기]

2) DataFrame의 통계 메서드 다뤄보기

: 연도별 lifeExp의 평균 계산

```
import pandas as pd
df=pd.read_csv('./data/gapminder.tsv',delimiter='\t')
print(df.groupby('year')['lifeExp'].mean())
```

```
year
1952    49.057620
1957    51.507401
1962    53.609249
1967    55.678290
1972    57.647386
1977    59.570157
1982    61.533197
1987    63.212613
1992    64.160338
1997    65.014676
2002    65.694923
2007    67.007423
Name: lifeExp, dtype: float64
```

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[pandas DataFrame 다루어 보기]

3) DataFrame의 통계 메서드 다뤄보기

: 연도별, 대륙별 lifeExp,gdpPercap의 평균 계산

```
import pandas as pd
df=pd.read_csv('./data/gapminder.tsv',delimiter='\t') #tsv(tab separrted value)
print(df.groupby(['year','continent'])['lifeExp','gdpPercap'].mean())
```

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[pandas DataFrame 다루어 보기]

3) DataFrame의 통계 메서드 다뤄보기

: 연도별, 대륙별 lifeExp,gdpPercap의 평균 계산

```
import pandas as pd
df=pd.read_csv('./data/gapminder.tsv',delimiter='\t') #tsv(tab separrted value)
print(df.groupby(['year','continent'])['lifeExp','gdpPercap'].mean())
```

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[pandas DataFrame 다루어 보기]

4) DataFrame의 통계 메서드 다뤄보기

: 대륙별로 그룹화한 데이터 갯수 세기

```
import pandas as pd
df=pd.read_csv('./data/gapminder.tsv',delimiter='\t')
print(df.groupby('continent')['country'].nunique())
```

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[실습1]-워밍업 문제를 pandas를 이용하여 해결해 볼까요?

'data/scientist.csv'파일을 불러와 다음의 문제를 해결해 보세요.

1) 과학자들의 평균 나이(Age)는 얼마일까요?

2) 직업(Occupation)이 화학자(Chemist)인 과학자는 누구일까요?

```
import pandas as pd
df=pd.read_csv('./data/scientists.csv')
Chemist=df.loc[df.Occupation=='Chemist']
print("1.과학자들의 평균 나이 : ",df['Age'].mean())
print("2.직업(Occupation)이 화학자(Chemist)인 과학자 : ",Chemist['Name'].values)
```

```
1.과학자들의 평균 나이 :  59.125
2.직업(Occupation)이 화학자(Chemist)인 과학자 :  ['Rosaline Franklin' 'Marie Curie']
```

```
import csv
f=open('./data/scientists.csv','r')
data=csv.reader(f)
next(data)
sumAge=0
aveAge=0.0
cnt=0
Chemist=[]
data=list(data)
for row in data :
    cnt+=1 #데이터 갯수 세기
    sumAge=sumAge+int(row[3])    #나이의 누적합
    if(row[-1]=='Chemist'):
        Chemist.append(row[0])
aveAge=sumAge/len(data)
print("1.과학자들의 평균 나이 : ",aveAge)
print("2.직업(Occupation)이 화학자(Chemist)인 과학자 : ",Chemist)
f.close()
```

```
import pandas as pd
df=pd.read_csv('./data/scientists.csv')
Chemist=df.loc[df.Occupation=='Chemist']
print("1.과학자들의 평균 나이 : ",df['Age'].mean())
print("2.직업(Occupation)이 화학자(Chemist)인 과학자 : ",Chemist['Name'].values)
```

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[실습2]-2018 월드컵 자료를 분석해 볼까요?

‘data/FIFA2018Statistics.csv’파일을 불러와 다음의 문제를 해결해 보세요

1) 가장 골을 많이 넣은 나라는 어느 나라일까요?

2) 가장 패스 정확도가 높은 나라는 어느 나라일까요?

3) 가장 경고&퇴장을 많이 받은 나라는 어느 나라일까요?

```
1.가장 골을 많이 넣은 나라 :  Belgium
2.가장 패스 정확도가 높은 나라 :  Mexico
3.가장 경고&퇴장을 많이 받은 나라 :  Germany
```

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[pandas로 그래프 그리기]

- matplotlib.pyplot기반
- Series나 DataFrame으로 생성한 데이터가 있을 때 그래프를 그릴 수 있음.

```
Series_data.plot([kind='graph_kind'][,option])
DataFrame_data.plot([x=label 혹은 position, y=label 혹은 position,]
                    [kind='graph_kind'][,option]
```

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[pandas로 그래프 그리기]

https://pandas.pydata.org/pandas-docs/stable/reference/frame.html

Plotting

`DataFrame.plot` is both a callable method and a namespace attribute for specific plotting methods of the form `DataFrame.plot.<kind>`.

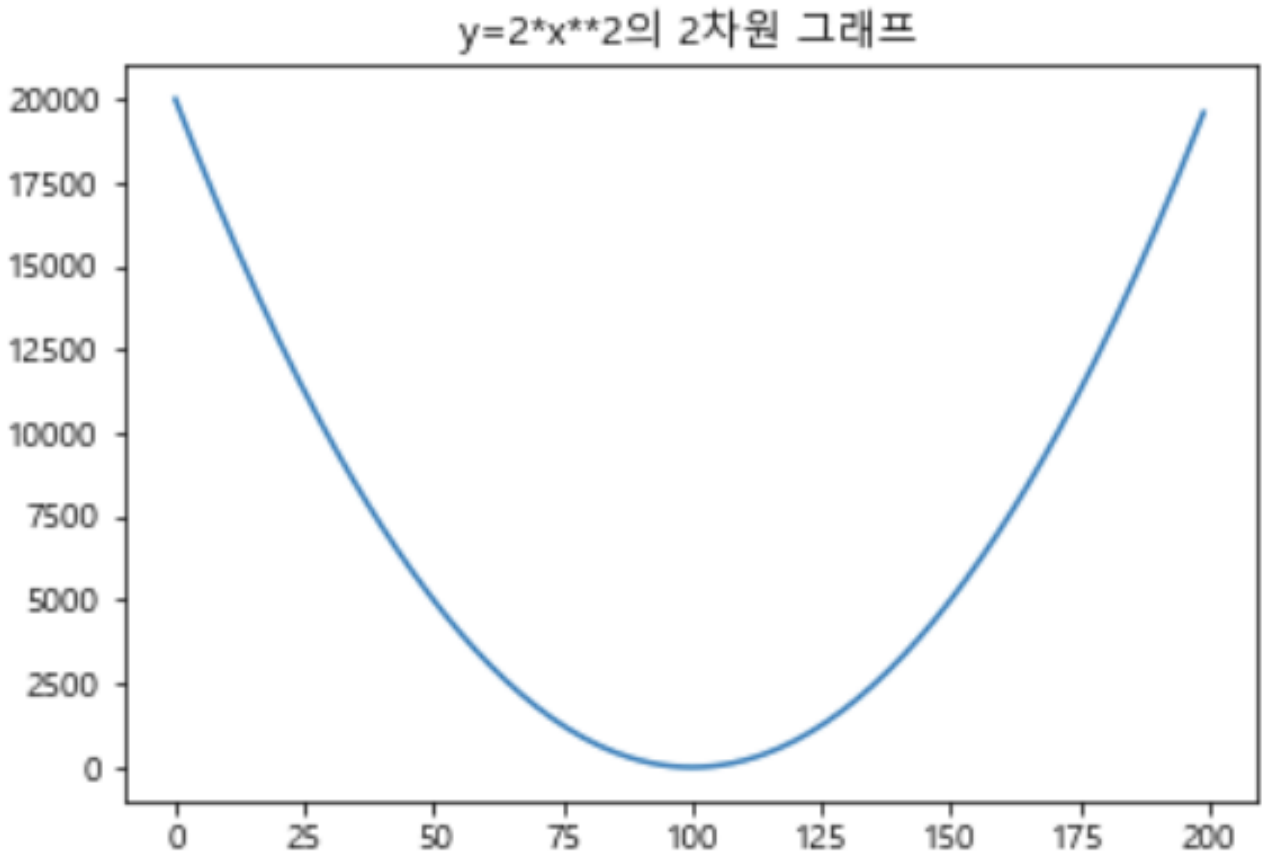
`DataFrame.plot`([x, y, kind, ax, ….])	DataFrame plotting accessor and method
`DataFrame.plot.area`(self[, x, y])	Draw a stacked area plot.
`DataFrame.plot.bar`(self[, x, y])	Vertical bar plot.
`DataFrame.plot.barh`(self[, x, y])	Make a horizontal bar plot.
`DataFrame.plot.box`(self[, by])	Make a box plot of the DataFrame columns.
`DataFrame.plot.density`(self[, bw_method, ind])	Generate Kernel Density Estimate plot using Gaussian kernels.
`DataFrame.plot.hexbin`(self, x, y[, C, …])	Generate a hexagonal binning plot.
`DataFrame.plot.hist`(self[, by, bins])	Draw one histogram of the DataFrame's columns.
`DataFrame.plot.kde`(self[, bw_method, ind])	Generate Kernel Density Estimate plot using Gaussian kernels.
`DataFrame.plot.line`(self[, x, y])	Plot Series or DataFrame as lines.
`DataFrame.plot.pie`(self, **kwargs)	Generate a pie plot.
`DataFrame.plot.scatter`(self, x, y[, s, c])	Create a scatter plot with varying marker point size and color.
`DataFrame.boxplot`(self[, column, by, ax, …])	Make a box plot from DataFrame columns.
`DataFrame.hist`(data[, column, by, grid, …])	Make a histogram of the DataFrame's.

python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[실습3]-Day2의 실습을 pandas를 이용하여 구현해 볼까요?

1) $y = 2x^2$ 그래프

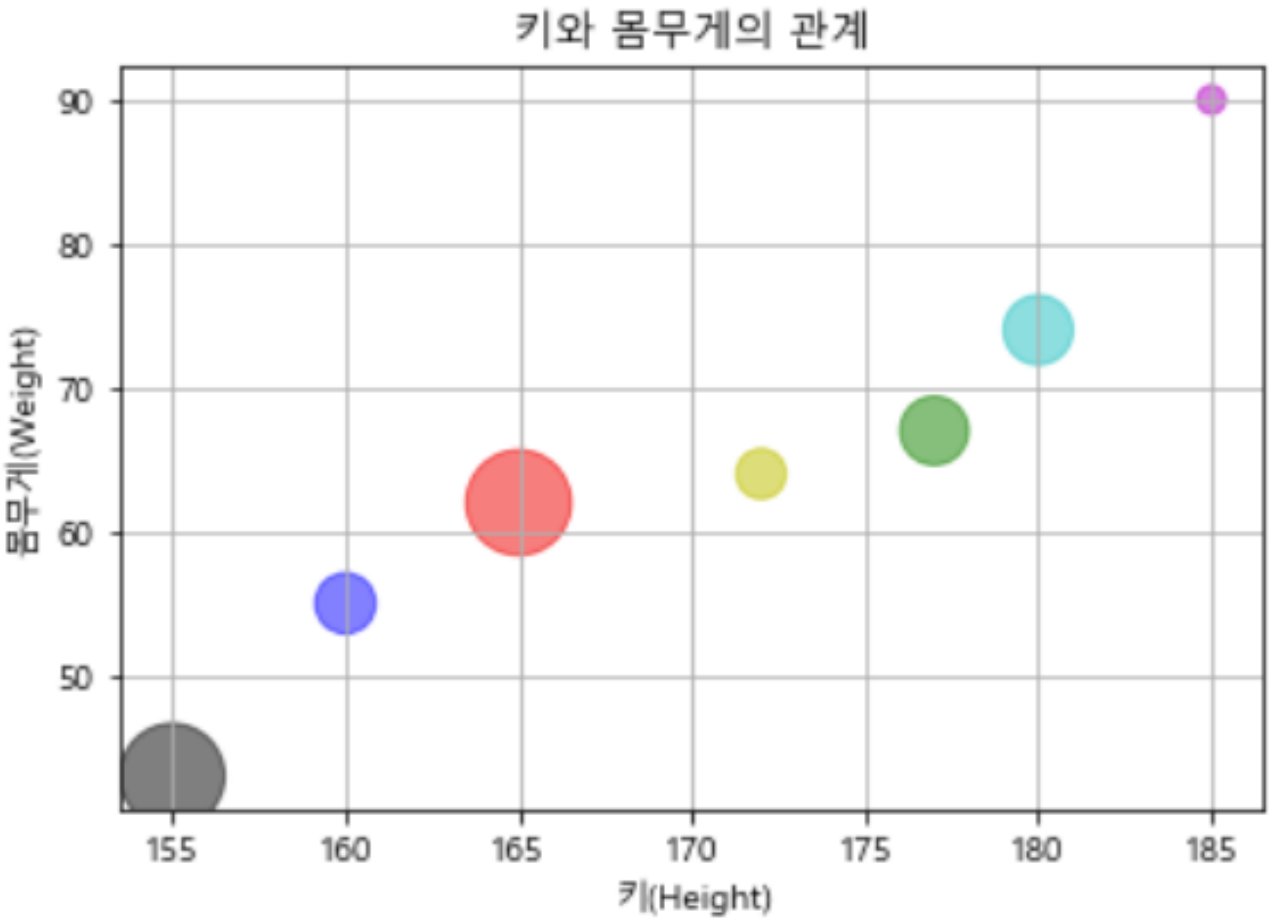


python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[실습3]-Day2의 실습을 pandas를 이용하여 구현해 볼까요?

2) 키와 몸무게의 산점도 그래프

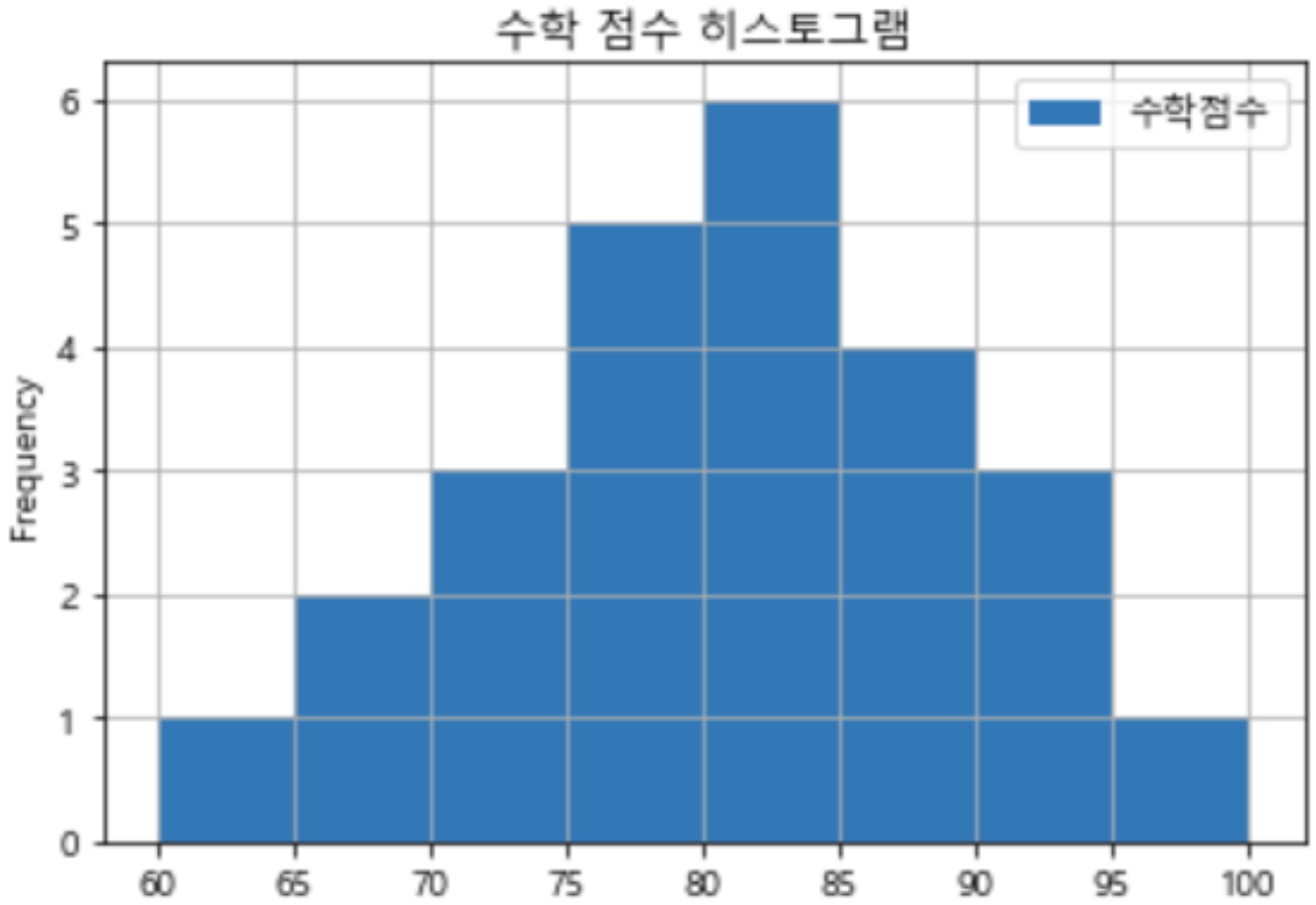


python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[실습3]-Day2의 실습을 pandas를 이용하여 구현해 볼까요?

3) 수학점수 히스토그램

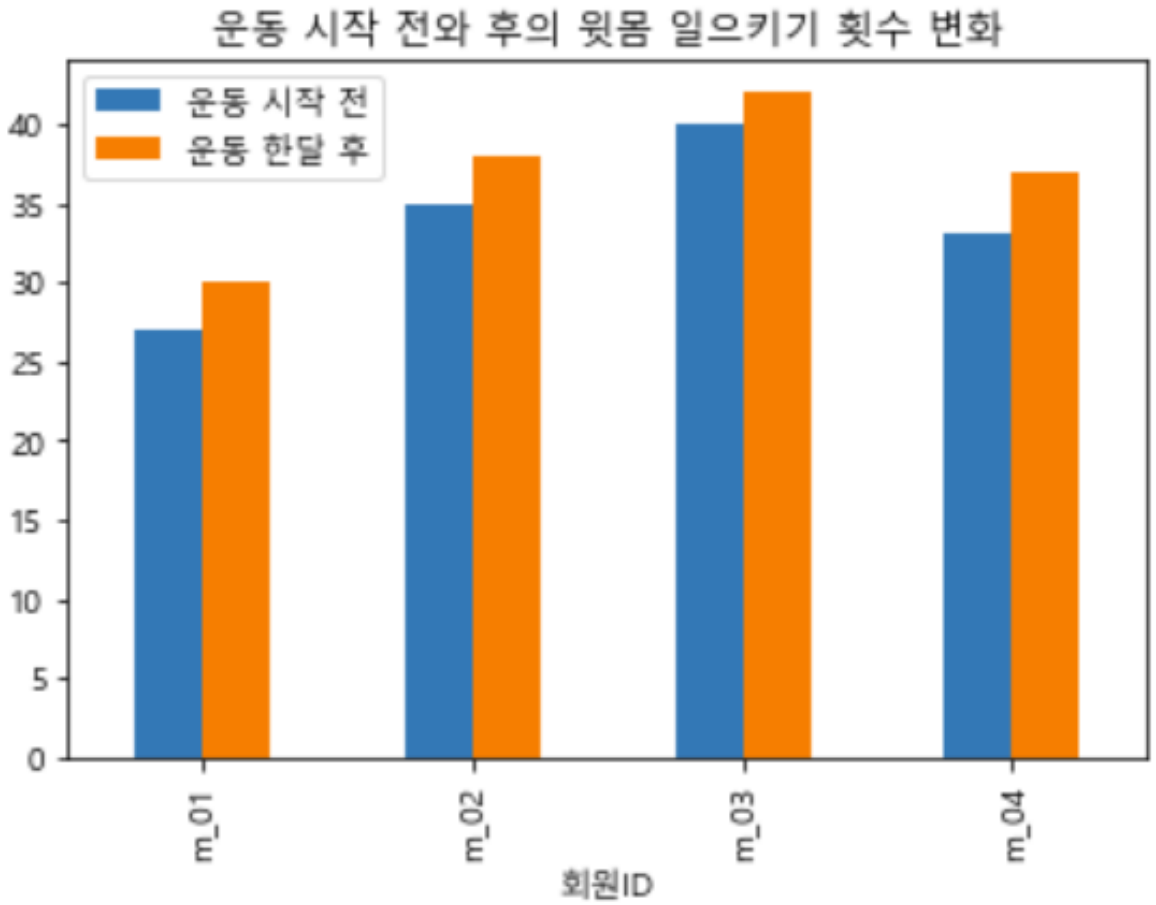


python pandas matplotlib

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

[실습3]-Day2의 실습을 pandas를 이용하여 구현해 볼까요?

4) 운동 시작 전과 후의 윗몸 일으키기 횟수 변화 막대 그래프



#3. [project]

- 서울 청소년의 스트레스 분석

1. 데이터 가져오기 : 서울열린데이터광장 > 보건 > 청소년건강

http://data.seoul.go.kr/dataList/datasetView.do?infId=10956&srvType=S&serviceKind=2

검색어 : *"청소년 정신건강 통계"* (*2* 건이 검색되었습니다.)

통합검색	데이터셋	카탈로그	활용갤러리	이용활용문의

데이터셋 (1건)

결과 더보기 +

통계 SHEET CHART OPEN API

서울시 청소년 정신건강 통계

〈보건 〉 청소년건강〉

제공기관 : 서울특별시

Sheet | Chart | Open Api

자료분석 XLS CSV HWP

언어 한국어 소수점 기간 년 2018 년 ~ 2018 년 자료검색

단위 : %

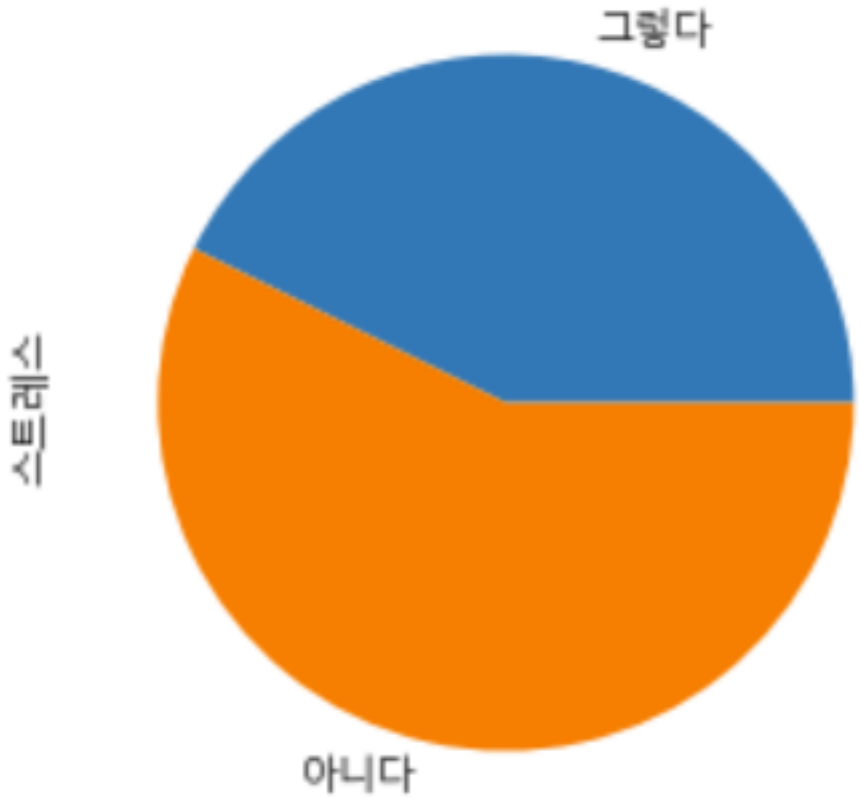
기간	구분	스트레스 인지율			우울감 경험률			자살 생각률		
		전체	남학생	여학생	전체	남학생	여학생	전체	남학생	여학생
2018	구분	42.7	34.5	51.5	29.6	24.2	35.4	15.4	11.8	19.2

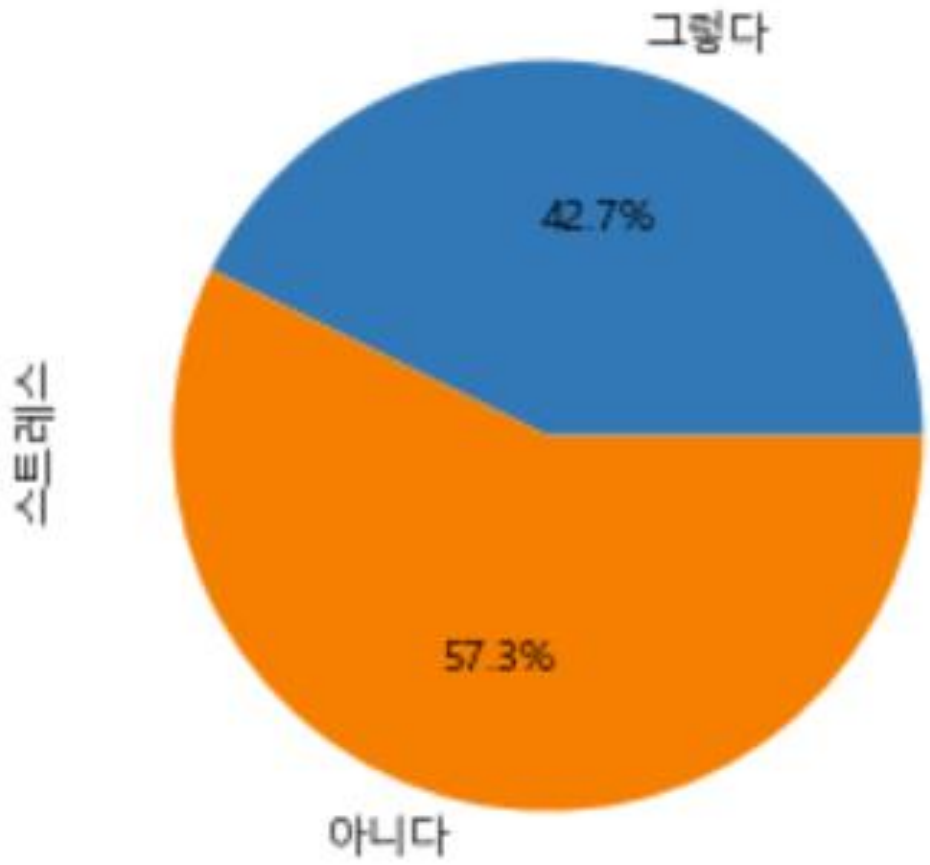
1. 데이터를 불러와서 데이터프레임으로 저장하기

2. 데이터에 필요한 행 추가하기

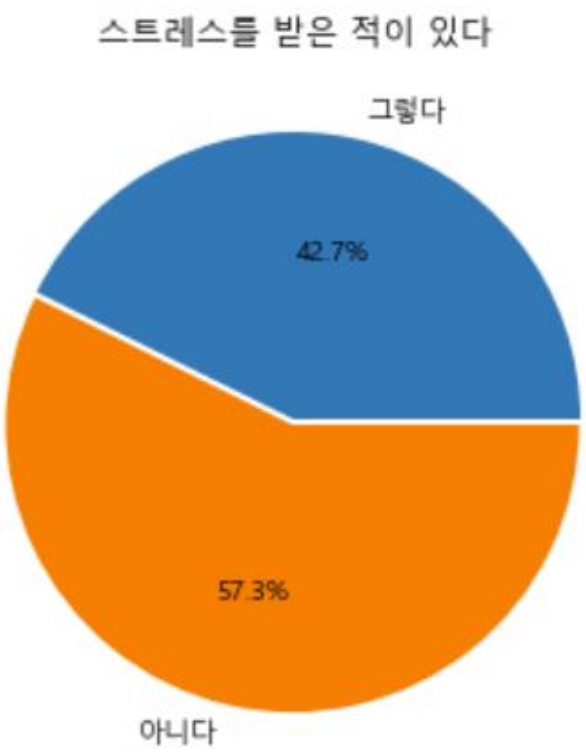
3. 데이터 최종 정리 : 인덱스 설정하기

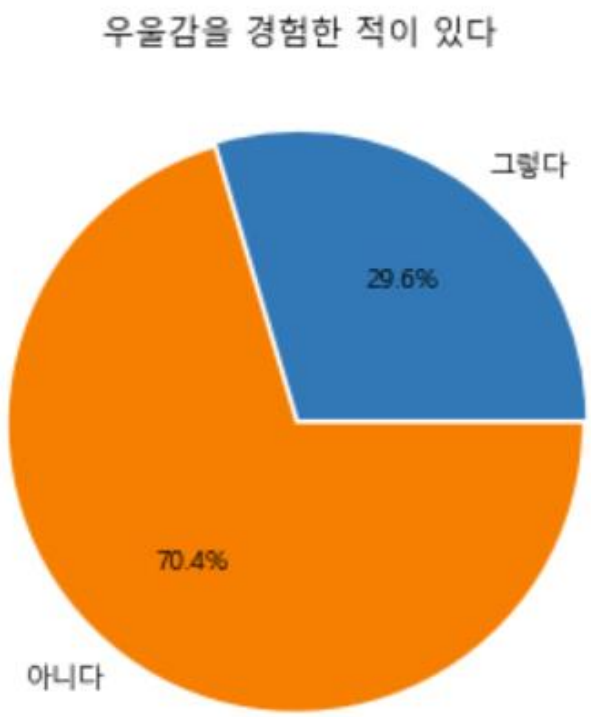
스트레스를 받은 적이 있는지 없는지에 관한 데이터 시각화

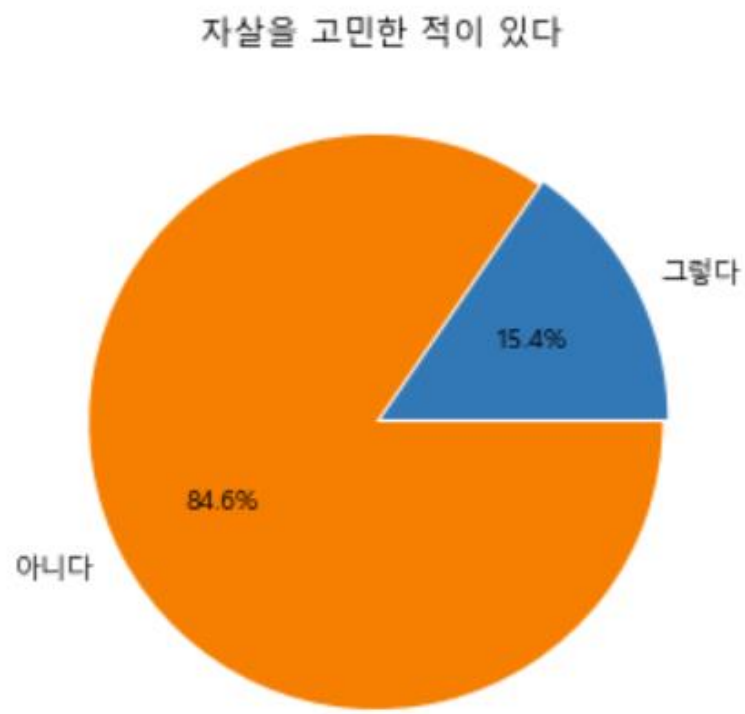




서울 청소년의 스트레스 분석(데이터 시각화)







프로젝트 발표 : 1/13(월)

제출물 : 발표자료(ppt),

파이썬 소스(학번_이름_주제명.ipynb 또는 .py)

리소스(코드를 실행시키는 데 필요한 모든 자료)

교사용컴퓨터의 바탕화면 [최종작품제출]-[자기이름]폴더에

최종 자료를 넣고, 발표 자료 최종 확인하기

추후 일정 안내(1/13~1/18)

1/13 (오전)	팀 프로젝트 완성 및 발표	09:00~10:30 프로젝트 발표 준비 및 완성 10:30~11:30 프로젝트 발표
1/13 (오후)	케라스를 활용한 딥러닝	13:00~13:50 1교시 14:00~14:50 2교시 15:00~15:50 3교시
1/14~18 (오전)	케라스를 활용한 딥러닝	09:00~09:50 1교시 10:00~10:50 2교시 11:00~11:50 3교시