

day3

보다 쉽게 데이터 다루기 (pandas, numpy)

2019 경북고등학교

[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

[numpy 라이브러리]

- 배열 데이터를 효과적으로 다룰 수 있게 돕는 라이브러리
- **import numpy as np** 로 np라는 별칭으로 주로 사용
- <https://numpy.org/>

[numpy 라이브러리 사용 비교]

```
import matplotlib.pyplot as plt
import numpy as np
```

```
t = np.arange(0., 5., 0.2)
```

```
plt.plot(t, t, 'r--', t, t**2, 'bs',
plt.show()
```

```
import matplotlib.pyplot as plt
```

```
t1=[]
```

```
t2=[]
```

```
t3=[]
```

```
for i in range(0,50,2) :
```

```
    t1.append(i/10)
```

```
    t2.append((i/10)**2)
```

```
    t3.append((i/10)**3)
```

```
plt.plot(t1, t1, 'r--', t1, t2, 'bs', t1, t3, 'g^')
```

```
plt.show()
```

[numpy 라이브러리 사용 비교]

```
import matplotlib.pyplot as plt
import random

dice=[]
for i in range(10) :
    dice.append(random.randint(1,6))
print(dice)
plt.hist(dice,bins=6)
plt.show()
```

```
import matplotlib.pyplot as plt
import numpy as np

dice=np.random.choice(range(1,7),10)
print(dice)
plt.hist(dice,bins=6)
plt.show()
```

[numpy 배열 다루어 보기]

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

arr3=np.arange(5)
print(arr3)
print(arr3.sum(),arr3.mean())
arr4=np.arange(1,5)
print(arr4)
print(arr4.cumsum())
print(arr4.cumprod())
```


#2. pandas 라이브러리

[pandas]

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

[pandas 데이터 타입]

- Series

index	values

- DataFrame

		columns		
index		values		

[pandas의 Series]

index	values

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

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

```
0    10
1    20
2    30
dtype: int64
```

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

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

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

	index			value	
	a			10	
	b			20	
	c			30	
	d			40	

[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

[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

[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'>
```

[pandas Series 다루어 보기]

2) 데이터 프레임에서 행단위 데이터 추출하기

- 데이터프레임명.loc['인덱스명'] 또는 데이터프레임명.iloc[행번호]

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

```
<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]

3) 데이터 프레임에서 열 단위 데이터 추출하기

- 데이터프레임명['컬럼명']

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

```
<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
```

[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)
```

[pandas DataFrame]

		columns		
index		values		

[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

[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
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())
```

[실습]-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?**

#3. [project]

- 서울 청소년의 스트레스 분석**

1. 데이터 가져오기 : 서울열린데이터광장 > 보건 > 청소년건강

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



보건



일반행정



문화관광



산업/경제



복지



환경



교통



도시관리



교육



안전



인구/가구



주택/건설

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

자료분석XLSCSVHWP

언어

한국어 ▼

소수점

▼

기간

년 ▼

2018 년 ▼

~

2018 년 ▼

자료검색

단위 : %

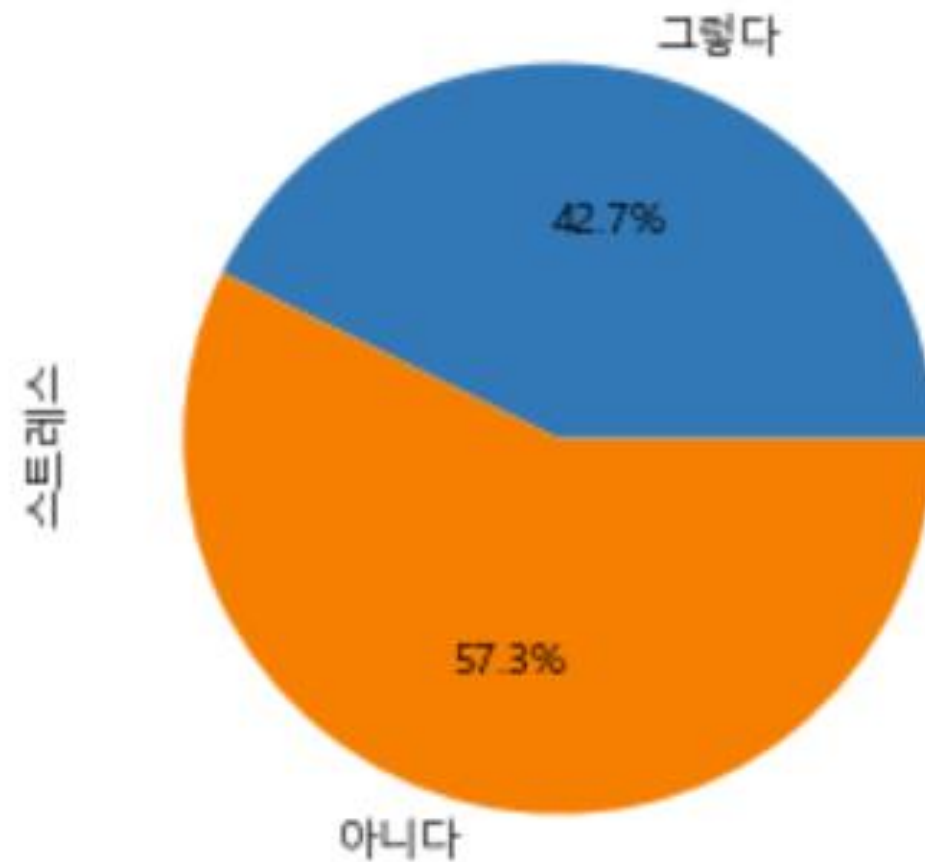
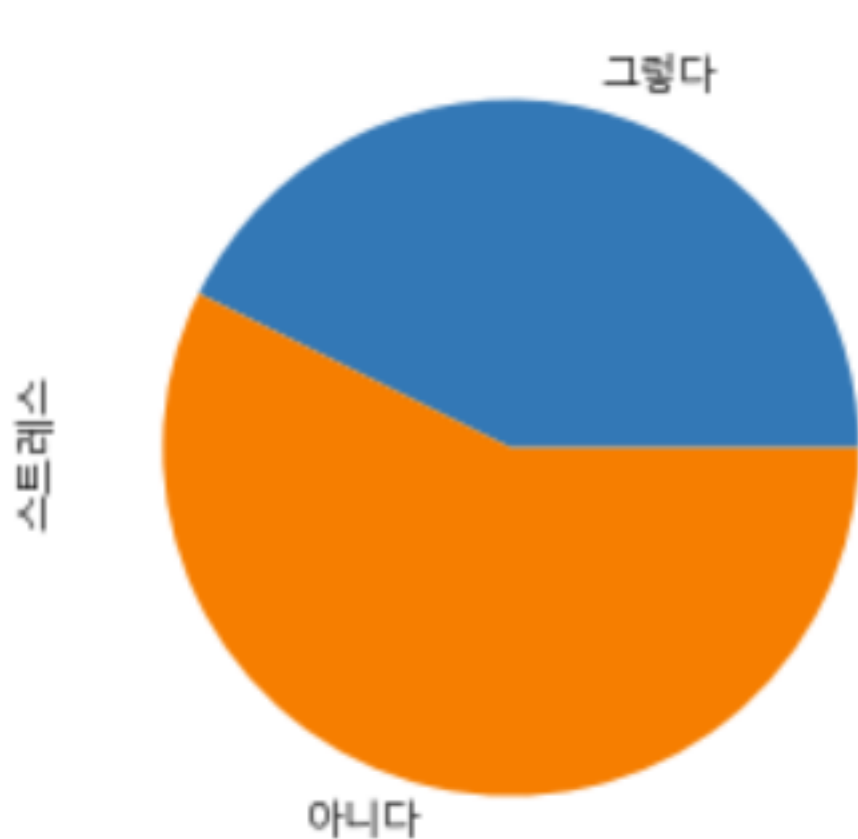
기간	구분	스트레스 인지율			우울감 경험률			자살 생각률		
		전체	남학생	여학생	전체	남학생	여학생	전체	남학생	여학생
2018	구분	42.7	34.5	51.5	29.6	24.2	35.4	15.4	11.8	19.2

1. 데이터를 불러와서 데이터프레임으로 저장하기

2. 데이터에 필요한 행 추가하기

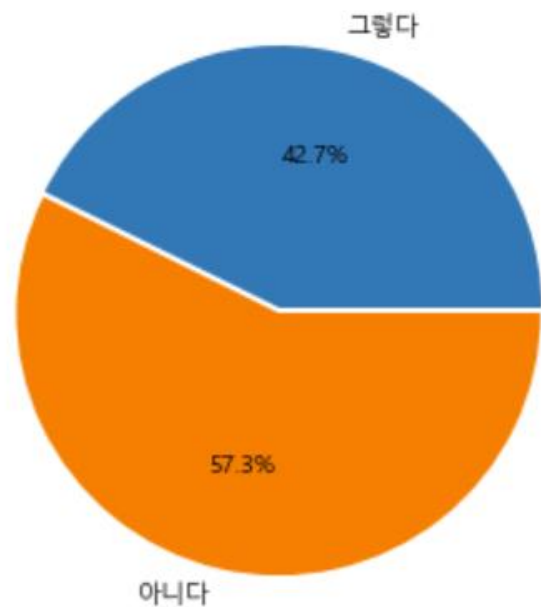
3. 데이터 최종 정리 : 인덱스 설정하기

스트레스를 받은 적이 있는지 없는지에 관한 데이터 시각화

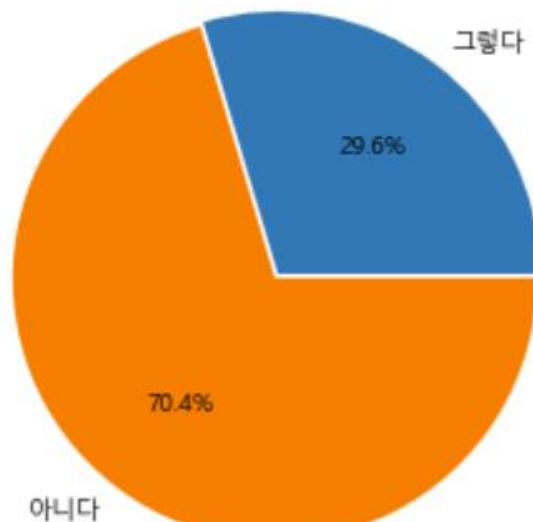


3개 항목을 모두 시각화

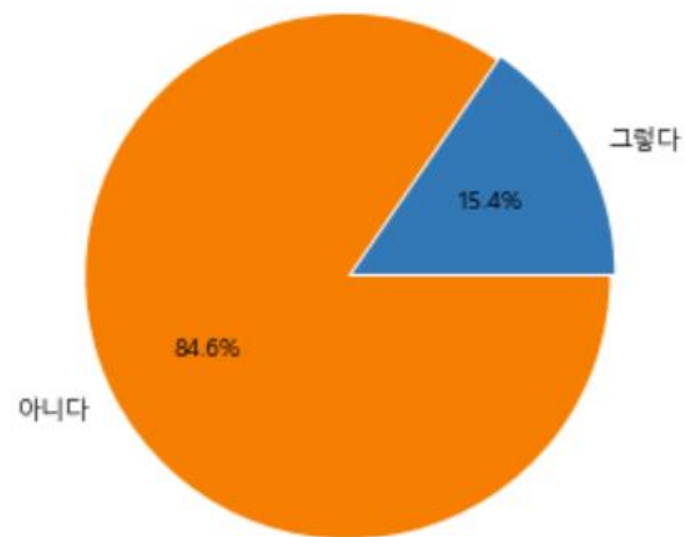
스트레스를 받은 적이 있다



우울감을 경험한 적이 있다



자살을 고민한 적이 있다



내일은 [공공데이터분석]

#1. 우리 동네 인구 구조 분석

#2. 지하철 데이터 분석