데이터 분석을 위해 사용하는 라이브러리

Numpy : 행렬 / 고차원 배열을 쉽게 처리, 수치 계산등을 해주는 라이브러리
Pandas : 데이터를 쉽게 다루고, 관리(Series, DataFrame, Panel)해주는 라이브러리
Matplotlib : 시각화를 위한 라이브러리

```
#분석을 위한 라이브러리 임포트
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

인덱스 = 행
컬럼 = 열

인덱싱 : 1) 표준 인덱싱 ~> [ ] 표기법으로 수행 https://bearwoong.tistory.com/65
2) loc https://ichi.pro/ko/python-pandas-dataframe-indegsing-mich-seullaiseu-261108134871414
3) iloc

데이터를 불러온 후 해당 열을 인덱스로 사용하기

옵션값으로 "index_col =" 을 사용

```
#CSV데이터 프레임 가져오기
#오토인덱싱 현상 생김
practice2 = pd.read_csv('practice.csv',encoding='utf-8')
practice2
```

	Unnamed: 0	날짜	운동	양
0	0	19-3-1	달리기	1.0
1	1	19-3-2	걷기	1.0
2	2	19-3-3	달리기	1.0
3	3	19-3-4	계단오르기	1.0
4	4	19-3-5	걷기	1.5
5	5	19-3-6	달리기	1.0

```
#오토 인덱싱 생긴거 제거 -> index_col = 0 0번째 열을 인덱스로 사용하기!!
pd.read_csv("practice.csv", index_col = 0, encoding = 'utf-8')
```

	날짜	운동	양
0	19-3-1	달리기	1.0
1	19-3-2	걷기	1.0
2	19-3-3	달리기	1.0
3	19-3-4	계단오르기	1.0
4	19-3-5	걷기	1.5
5	19-3-6	달리기	1.0

특정 인덱스를 요약 하거나 분석할 때

.pivot_table 함수사용 => pivot : 축

```
#pivot_table -> 요약!!!
practice3.pivot_table(index = '운동') #avg 기준! 데이터타입이 숫자인 것만 index기준으로 평균구해줌
```

	양
운동	
걷기	1.25
계단오르기	1.00
달리기	1.00

```
# aggfunc : 계산방식 / 디폴트 : 평균
practice3.pivot_table(index = '운동', aggfunc = np.sum) #np.sum : 합계를 구해줌
```

	양
운동	
걷기	5.0
계단오르기	1.0
달리기	5.0

```
#여러 arrfunc 사용하기 -> 리스트 사용(or 튜플)
practice3.pivot_table(index= '운동', aggfunc=[np.sum, np.average, len])
```

	sum	average	len	
	양	양	날짜	양
운동				
걷기	5.0	1.25	4	4.0
계단오르기	1.0	1.00	1	1.0
달리기	5.0	1.00	5	5.0

.pivot함수

```
prac_pivot = practice3.pivot('날짜','운동','양')
                    #날짜 : 인덱스(행), 운동: 종류(열), 양: 테이블 가운데 채워지는 값(양)
prac_pivot
#NaN : 행과 열에 적합한 값이 없다() => 삭제, 수치로 변환해야함
#수치변환 : 평균 / 0 / 기본값으로 변환
```

운동	걷기	계단오르기	달리기
날짜			
19-3-1	NaN	NaN	1.0
19-3-10	NaN	NaN	1.0
19-3-2	1.0	NaN	NaN
19-3-3	NaN	NaN	1.0
19-3-4	NaN	1.0	NaN
19-3-5	1.5	NaN	NaN
19-3-6	NaN	NaN	1.0
19-3-7	NaN	NaN	1.0
19-3-8	1.0	NaN	NaN
19-3-9	1.5	NaN	NaN

불린 인덱싱 : 조건을 주어서 나온 true/false로 인덱싱 하기

```
#대분류를 통해 연령, 성별, 등등으로 나눠서 분석 조건 주기!!
#not_exercise[] : 조건을 넣기위한 [] 역할
not_exercise['대분류'] == '성별' #대분류 안에 있는 것 중 성별에 해당하는 것만 가져오기 : 불린값 추출
                              #브로드캐스팅(백터(열)를 다 돌아 다니면서 확인)
                              #불린 인덱싱 -> 인덱싱 방법중 1개 DataFrame에 적용시 DataFrame 반환, Series에 적용시 Series반환
```

```
0     False
1      True
2      True
3     False
4     False
5     False
6     False
7     False
8     False
9     False
10    False
11    False
12    False
13    False
14    False
15    False
16    False
17    False
18    False
19    False
20    False
21    False
Name: 대분류, dtype: bool
```

불린 값으로 인덱싱하기

```
not_ex_sex = not_exercise[not_exercise['대분류'] == '성별'].copy()
not_ex_sex
```

	대분류	분류	운동을 할 충분한 시간이 없어서	함께 운동을 할 사람이 없어서	운동을 할 만한 장소가 없어서	운동을 싫어해서	기타
1	성별	남자	55.2	7.9	5.5	29.8	1.6
2	성별	여자	45.0	8.3	6.0	38.8	1.8

drop함수 : 특정 행 혹은

```
df
```

	c0	c1	c2	c3
0	0	1	4	7
1	1	2	5	8
2	2	3	6	9

```
df.drop(index= 0, inplace=True)
df
```

	c0	c1	c2	c3
1	1	2	5	8
2	2	3	6	9

```
df.drop(index= range(1,3), inplace=True)
df
```

	c0	c1	c2	c3
0	0	1	4	7

```
df.drop(columns= 'c2', inplace=True)
df
```

	c0	c1	c3
0	0	1	7
1	1	2	8
2	2	3	9

```
df.drop(columns= ['c2', 'c3'], inplace=True)
df
```

	c0	c1
0	0	1
1	1	2
2	2	3

원하는 컬럼을 인덱스(행)으로 변환 : set_index 함수 사용

	분류	운동을 할 충분한 시간이 없어서	함께 운동을 할 사람이 없어서	운동을 할 만한 장소가 없어서	운동을 싫어해서	기타
23	도심권	50.5	6.9	4.5	36.4	1.7
24	동북권	47.4	7.9	5.9	36.1	2.7
25	서북권	49.4	13.9	7.4	28.5	0.8
26	서남권	50.7	6.7	5.7	35.0	1.9
27	동남권	52.2	6.9	4.9	35.4	0.6

```
not_ex_place.set_index('분류', inplace=True)
```

```
not_ex_place
```

분류	운동을 할 충분한 시간이 없어서	함께 운동을 할 사람이 없어서	운동을 할 만한 장소가 없어서	운동을 싫어해서	기타
도심권	50.5	6.9	4.5	36.4	1.7
동북권	47.4	7.9	5.9	36.1	2.7
서북권	49.4	13.9	7.4	28.5	0.8
서남권	50.7	6.7	5.7	35.0	1.9
동남권	52.2	6.9	4.9	35.4	0.6

loc VS iloc

- loc의 경우

```
df1.loc[:2, ['Survived', 'Pclass','Name']]
```

라벨을 이용

	Survived	Pclass	Name
0	0	3	Braund, Mr. Owen Harris
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...
2	1	3	Heikkinen, Miss. Laina

- iloc의 경우

```
df1.iloc[:2,1:4]
```

인덱스를 이용

	Survived	Pclass	Name
0	0	3	Braund, Mr. Owen Harris
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...

비파괴함수인 경우 함수사용하고 변경된 값이 기존 변수에 변경이 안되서

따로 변수에 넣어줘야하는데

파괴함수로 사용할 경우

변수에 넣어주지 않아도 기존 변수 내용이 바뀌는?

이런 느낌 인거 같아여

오후 3:46

*0504.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

데이터분석

- 전처리, 인사이트

- [일반적인 분석]:
  기술통계: 간단한 통계(평균, 합, 그룹별 요약)
  추론통계: 평균차이낸 결과값이 확률적으로 올바른지 검증
  => 주장하는 바가 있어야 함. 주장하는 바가 있어서 증명.
  => 내가 주장하는 가설: 대립가설(연구가설), alternative hypothesis(대안, 선택가설), H1
  => 내 가설을 무력화(0으로 만들어버리는)시키는 가설: 영가설(귀무가설)
  null hypothesis, Ho
  => 영가설이 발생할 확률이 희박하다면 나의 영가설과 대립되는 대립가설이 옳다.

  => 대륙별로 차이가 있는 것 같다.(대립가설), 나는 95%이상!
  <----> 대륙별로 차이가 없는 것 같다.(영가설), 5%미만! **인 것만 확인하면 됨**
  정말로 희박하다면! 내 주장이 옳다.

Ln 11, Col 28 100% Windows (CRLF) UTF-8