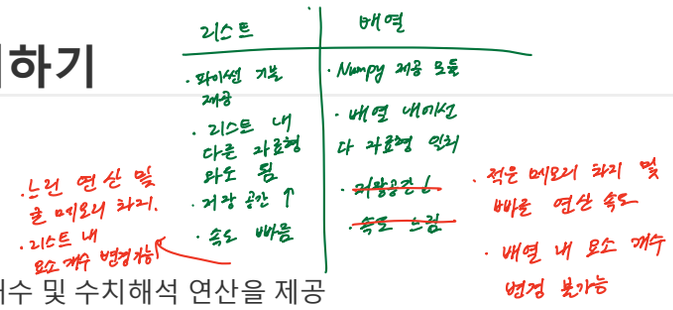
13. 데이터 분석 기초

13.1 Numpy 모듈 이해하기



numpy모듈이란?

파이썬에서 배열을 생성하고, 빠른 선형대수 및 수치해석 연산을 제공

리스트	배열
파이썬에서 원래 제공하는 순서열 자료형 서로 다른 데이터형도 요소로 포함 가능 리스트 내 요소 개수 변경 가능 느린 연산 속도 및 큰 메모리 차지	numpy에서 제공하는 순서열 자료형 모든 요소가 같은 데이터형을 가져야 함 리스트 내 요소 개수 변경 불가능 빠른 연산 속도 및 적은 메모리 차지

배열 만들기

1차원 배열은 벡터, 2차원 배열은 행렬이라 한다.

```
import numpy as np
list1 = [1,2,3]
list2 = [[1,2,3],[4,5,6]]
a = np.array(list1)
b = np.array(list2)

print(a.ndim) # 1
print(a.size) # 3
print(a.dtype) # int
print(a.shape) #(3,)

print(b.ndim) # 2
print(b.size) # 6
print(b.dtype) # int
print(b.shape) #(2,3)
print(type(a)) #numpy.ndarray
```

```
import numpy as np
list1 = [1,2,3]
list2 = [[1,2,3],[4,5,6]]
data1 = np.array(list1)
data2 = np.array(list2)

data1.ndim  # 1
data1.size  # 3
data1.dtype  # int
data1.shape  # (3,)

data2.ndim  # 2
data2.size  # 6
data2.dtype  # int
data2.shape  # (2,3)
```

```
a = np.zeros((2,3))
b = np.ones((2,2))
c = np.full((2,3),4)

print(a)
'''
[[0.,0.,0.],
[0.,0.,0.]]
'''
```

```
a = np.zeros((2,2))
[[0.0],
 [0.0]]
b = np.ones((4,2))
[[1.],
 [1.],
 [1.],
 [1.]]
c = np.full((2,3),4)
[[4.4.4],
 [4.4.4]]
```

```
print(b)
'''
[[1,1],
[1,1]]
'''

print(c)
'''
[[4,4,4],
[4,4,4]]
'''
```

np.ones와 np.zeros는 각 요소들의 값이 float64형이다

```
a = np.eye(2)
b = np.arange(10)
c = np.linspace(0,100,5)
d = np.linspace(10,20,3)
print(a)
'''
[[1.,0.],
[0.,1.]]
'''

print(b)
'''
[0,1,2,3,4,5,6,7,8,9]
'''

print(c)
'''
[0.,25.,50.,75.,100.]
'''

print(d)
'''
[10.,15.,20]
'''
```

a= np.eye(2)
[[1.0],
[0.1]]

b= np.arange(10)
[0.1.2.3.4.5.6.7.8.9]

C= np.linspace ~~(10.5)~~ (0,10,3)
[0. 5. 10]

배열 인덱싱

numpy 배열에서 특정 위치의 요소에 접근하고 싶을 때 각 차원별로 일반적인 파이썬 인덱싱을 사용한다.

```
a = np.array([
[0,1,2,3],
[2,3,4,5]
])

print(a[0][3]) #3
print(a[1][0]) #2
print(a[0][2]) #2
```

배열 슬라이싱

numpy 배열에서 여러 범위의 요소에 접근하고 싶을 때 각 차원별로 일반적인 파이썬 슬라이싱을 사용한다.

```
a = np.array([
[0,1,2,3],
[4,5,6,7]
])

print(a[0,:]) # [0,1,2,3]
print(a[:,1]) #[1, 5]
print(a[1,1:]) # [5,6,7]
print(a[:2,:2]) # [[0,1], [4,5]]
```

벡터화 연산

리스트 연산

```
list1 = [1,2,3,4]
print(2*list1) # [1,2,3,4,1,2,3,4]
```

왜냐면 리스트는 모든 형태의 자료형이 들어갈 수 있으므로..

```
list_matrix = np.array(list1)
print(2*list_matrix) #[2,4,6,8]
```

14.2 pandas 모듈 이해하기

pandas 모듈이란?

numpy모듈을 기반으로 시계열이나 표 형태의 데이터를 불러오고 다루기 위한 모듈

세 가지 자료 구조를 지원

- Series : 1차원 자료구조
- DataFrame : 2차원 자료구조
- Panel : 3차원 자료구조

1차원, Series
2차원, DataFrame
3차원 : Panel

Series

인덱스가 있는 1차원 numpy 배열로 생각

따로 인덱스를 지정하지 않으면 정수값 0부터 인덱스가 된다.

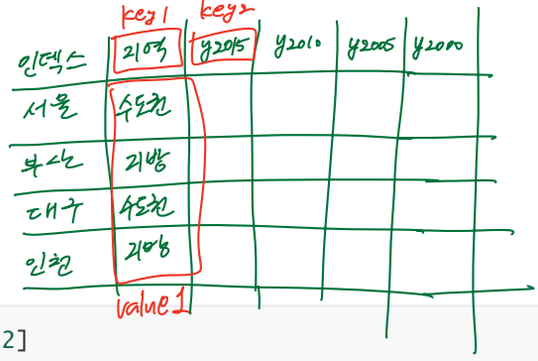
```
# 인덱스를 지정하지 않았을 경우
values = [192, 3924, 5932]
a = pd.Series(values)
print(a)
'''
0     192
1    3924
2    5932
dtype: int64
'''
```

```
# 인덱스를 지정했을 경우
values = [1192, 3924, 5832]
indexes = ['서울','부산','대전']
a = pd.Series(values, index = indexes)
print(a)
'''
서울    1192
부산    3924
대전    5832
dtype: int64
'''
```

DataFrame

2차원 numpy 행렬 데이터에 인덱스를 붙인 것

열마다 각기 다른 데이터형을 가질 수 있지만, 열 안에 있는 데이터들의 자료형은 일치해야 함.



DataFrame 인스턴스 만드는 법

1. 각 열의 데이터를 리스트로 준비
2. 각 열의 이름을 키로 가지는 딕셔너리 만듦
3. 딕셔너리를 DataFrame 함수에 넣음.

```
y2015 = [9904212, 3448737, 2890451, 2466052]
y2010 = [8631482, 3393191, 2632035, 2431774]
y2005 = [9762546, 3512547, 2517680, 2456016]
y2000 = [9853972, 3655437, 2466338, 2473990]
ratio = [0.0283, 0.0163, 0.0982, 0.0141]
region = ["수도권", "경상권", "수도권", "경상권"]

# 각 열의 이름을 딕셔너리로 만들어줌
```

```
table = {
    "지역":region,
    "2015": y2015,
    "2010": y2010,
    "2005": y2005,
    "2000": y2000,
}
index = ["서울","부산","인천","대구"]
df = pd.DataFrame(table, index = index)
df
'''
    지역   2015     2010     2005     2000
서울  수도권 9904212 8631482 9762546 9853972
부산  경상권 3448737 3393191 3512547 3655437
인천  수도권 2890451 2632035 2517680 2466338
대구  경상권 2466052 2431774 2456016 2473990
'''
```

y2015 = [20151, 20152, 20153, 20154]
y2010 = [20101, 20102, 20103, 20104]
y2005 = [20051, 20052, 20053, 20054]
y2000 = [20001, 20002, 20003, 20004]
region = ['수도권','경상권', '수도권', '경상권']
table = {'지역': region
'y2015': y2015
:
'y2000': y2000}
data = pd.DataFrame(table,
index=['서울','부산','인천','대구']

14.3 Matplotlib 모듈 이해하기

Matplotlib 모듈이란?

파이썬에서 chart나 plot으로 데이터를 시각화하는 패키지

다음의 시각화 기능을 지원함.

- line plot, scatter plot, contour plot, surface plot
- bar chart 및 histogram
- box plot

Pyplot 서브모듈이란?

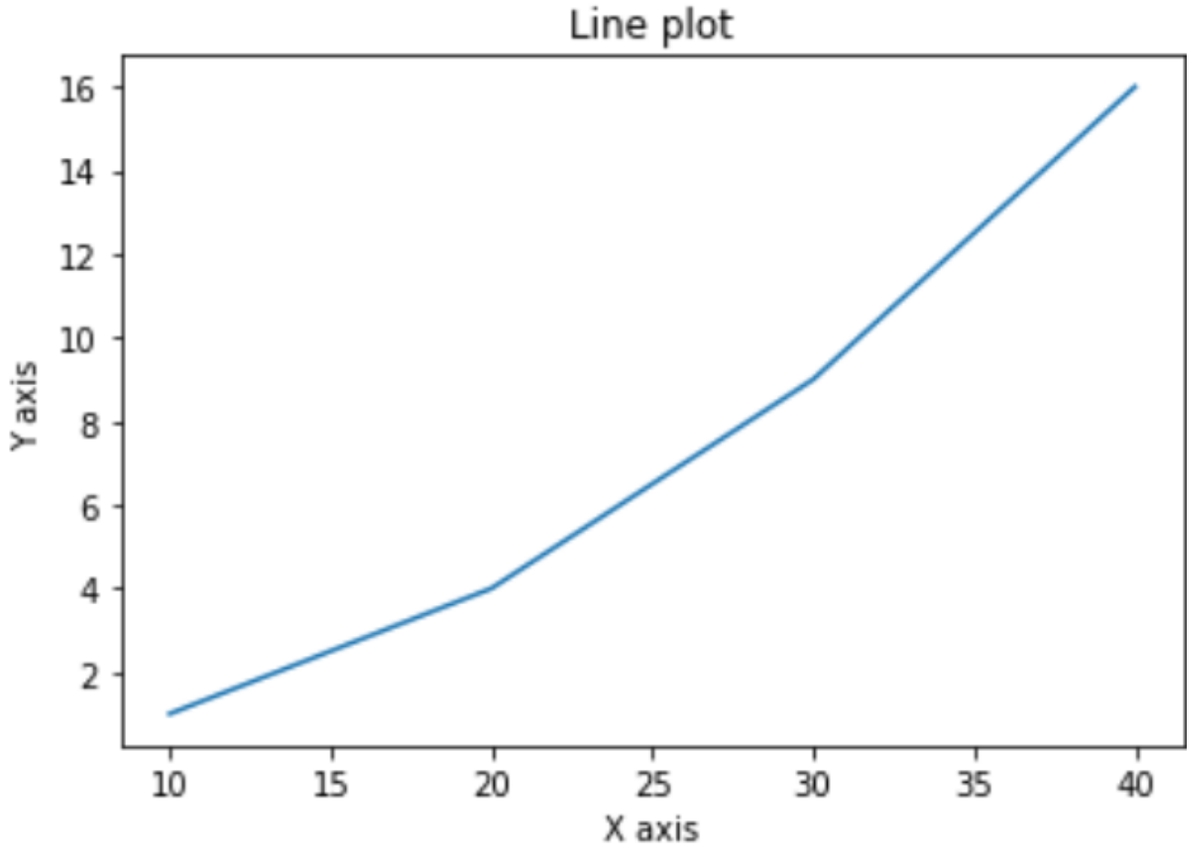
매트랩의 시각화 명령어를 그대로 사용할 수 있도록 지원

```
%matplotlib inline
```

라인 플롯 그리기

```
plt.title('Line plot')
x = [10,20,30,40]
y = [1,4,9,16]
plt.plot(x,y)
plt.xlabel('X axis')
plt.ylabel('Y axis')
plt.show()
```

라인 플롯
Import matplotlib as plt
plt.title("Line Plot")
X = range(10)
y = 10.X + 10
plt.plot(X, y)
plt.



matplotlib의 여러 플롯

bar함수

pie 함수

hist 함수

scatter 함수

contourf 함수

plot_surface함수