동양미래대학교

데이터분석입문

Lecture 12. NumPy로 인구 데이터 분석하기

동양미래대학교
인공지능소프트웨어학과
강 환수

목차

01. numpy를 활용한 나만의 프로젝트 만들기

02. 알고리즘을 코드로 표현하기

❖ ① 관심 있는 데이터 찾기

- 공공데이터포털(https://www.data.go.kr/) 등에서 관심 있는 데이터 찾기

❖ ② 데이터 살펴보며 질문하기 (1/2)

- 데이터를 살펴보는 방법
  - ◆ 엑셀 같은 스프레드시트 프로그램 등을 활용해 데이터 자세히 살펴보기
  - ◆ 데이터가 담고 있는 내용(또는 담고 있지 않은 내용)
  - ◆ 데이터가 기록된 기간은 언제부터 언제까지인지
  - ◆ 어떤 형태로 시각화해보면 어떤 정보들을 알 수 있을지 생각해보기
  - ◆ 데이터를 보며 궁금한 내용들 자유롭게 질문하기

❖ ② 데이터 살펴보며 질문하기 (2/2)

- age.csv의 인구 데이터를 보며 떠오른 질문들 예시
  - ◆ 전국에서 영유아들이 가장 많이 사는 지역은 어디일까?
  - ◆ 보통 학군이 좋다고 알려진 지역에는 청소년들이 많이 살까?
  - ◆ 광역시 데이터를 10년 단위로 살펴보면 청년 비율이 줄고 있다는 사실을 알 수 있을까?
  - ◆ 서울에서 지난 5년간 인구가 가장 많이 증가한 구는 어디일까?
  - ◆ 우리 동네의 인구 구조와 가장 비슷한 동네는 어디일까?

영유아들? 청소년들? 청년비율? 인구 구조와 가장 비슷한 동네?

문제를 명확히 정의할 필요가 있어 보입니다.

❖ ③ 질문을 명확한 문제로 정의하기

- 예를 들어 아래와 같이 문제를 좀 더 명확하게 정의할 수 있습니다.

◆ 전국에서 영유아들이 가장 많이 사는 지역은 어디일까?

➢ 전국에 있는 읍면동 중 만 0세 이상 6세 이하의 인구 비율이 높은 상위 10곳은?

◆ 우리 동네의 인구 구조와 가장 비슷한 동네는 어디일까?

➢ 전국에서 우리 동네의 연령별 인구 구조와 가장 형태가 비슷한 지역은 어디일까?

위 문제를 해결하기 위해서,
알고리즘을 어떻게 설계해야 할까요?

❖ ④ 알고리즘 설계하기

- [문제] 전국에서 우리 동네의 연령별 인구 구조와 가장 형태가 비슷한 지역은 어디일까?
  - ◆ Step 1) 데이터를 읽어온다.
  - ◆ Step 2) 궁금한 지역의 이름을 입력 받는다.
  - ◆ Step 3) 궁금한 지역의 인구 구조를 저장한다.
  - ◆ Step 4) 궁금한 지역의 인구 구조와 가장 비슷한 인구 구조를 가진 지역을 찾는다.
  - ◆ Step 5) 가장 비슷한 곳의 인구 구조와 궁금한 지역의 인구 구조를 시각화한다.

numpy를 활용하여 궁금한 지역의 인구 데이터를 출력하는 코드를 작성해 보겠습니다!

02. 알고리즘을 코드로 표현하기

01. numpy를 활용한 나만의 프로젝트 만들기

❖ Step 1) 데이터 읽어 오기

```
import csv

f = open('age.csv', encoding='cp949')
data = csv.reader(f)

header = next(data)
print(header)

for row in data:
    print(row)
    break

f.close()
```

❖ Step 1) 데이터 읽어 오기(실행결과)

['행정구역', '2021년11월_계_총인구수', '2021년11월_계_연령구간인구수', '2021년11월_계_0세', '2021년11월_계_1세', '2021년11월_계_2세', '2021년11월_계_3세', '2021년11월_계_4세', '2021년11월_계_5세', '2021년11월_계_6세', '2021년11월_계_7세', '2021년11월_계_8세', '2021년11월_계_9세', '2021년11월_계_10세', '2021년11월_계_11세', '2021년11월_계_12세', '2021년11월_계_13세', '2021년11월_계_14세', '2021년11월_계_15세', '2021년11월_계_16세', '2021년11월_계_17세', '2021년11월_계_18세', '2021년11월_계_19세', '2021년11월_계_20세', '2021년11월_계_21세', '2021년11월_계_22세', '2021년11월_계_23세', '2021년11월_계_24세', '2021년11월_계_25세', '2021년11월_계_26세', '2021년11월_계_27세', '2021년11월_계_28세', '2021년11월_계_29세', '2021년11월_계_30세', '2021년11월_계_31세', '2021년11월_계_32세', '2021년11월_계_33세', '2021년11월_계_34세', '2021년11월_계_35세', '2021년11월_계_36세', '2021년11월_계_37세', '2021년11월_계_38세', '2021년11월_계_39세', '2021년11월_계_40세', '2021년11월_계_41세', '2021년11월_계_42세', '2021년11월_계_43세', '2021년11월_계_44세', '2021년11월_계_45세', '2021년11월_계_46세', '2021년11월_계_47세', '2021년11월_계_48세', '2021년11월_계_49세', '2021년11월_계_50세', '2021년11월_계_51세', '2021년11월_계_52세', '2021년11월_계_53세', '2021년11월_계_54세', '2021년11월_계_55세', '2021년11월_계_56세', '2021년11월_계_57세', '2021년11월_계_58세', '2021년11월_계_59세', '2021년11월_계_60세', '2021년11월_계_61세', '2021년11월_계_62세', '2021년11월_계_63세', '2021년11월_계_64세', '2021년11월_계_65세', '2021년11월_계_66세', '2021년11월_계_67세', '2021년11월_계_68세', '2021년11월_계_69세', '2021년11월_계_70세', '2021년11월_계_71세', '2021년11월_계_72세', '2021년11월_계_73세', '2021년11월_계_74세', '2021년11월_계_75세', '2021년11월_계_76세', '2021년11월_계_77세', '2021년11월_계_78세', '2021년11월_계_79세', '2021년11월_계_80세', '2021년11월_계_81세', '2021년11월_계_82세', '2021년11월_계_83세', '2021년11월_계_84세', '2021년11월_계_85세', '2021년11월_계_86세', '2021년11월_계_87세', '2021년11월_계_88세', '2021년11월_계_89세', '2021년11월_계_90세', '2021년11월_계_91세', '2021년11월_계_92세', '2021년11월_계_93세', '2021년11월_계_94세', '2021년11월_계_95세', '2021년11월_계_96세', '2021년11월_계_97세', '2021년11월_계_98세', '2021년11월_계_99세', '2021년11월_계_100세 이상']
['서울특별시 (1100000000)', '9,520,880', '9,520,880', '43,492', '45,054', '49,318', '51,995', '56,210', '63,757', '67,763', '66,950', '67,744', '74,276', '72,335', '72,732', '69,959', '74,746', '78,816', '72,760', '71,164', '79,106', '81,199', '85,628', '99,222', '116,143', '118,167', '130,171', '141,973', '152,695', '160,467', '167,496', '168,785', '171,275', '163,601', '150,119', '143,789', '140,390', '135,790', '134,809', '135,124', '135,240', '146,723', '155,189', '154,986', '155,424', '149,681', '131,846', '139,332', '136,958', '141,983', '153,857', '158,331', '160,766', '170,686', '164,108', '163,523', '158,165', '146,324', '136,933', '143,792', '139,146', '139,431', '145,596', '153,238', '157,332', '143,607', '135,335', '134,397', '123,145', '129,445', '108,604', '96,292', '103,732', '72,522', '81,543', '81,820', '79,906', '80,925', '63,365', '57,788', '57,285', '56,204', '64,534', '49,960', '42,822', '40,456', '34,745', '30,034', '26,040', '23,278', '18,473', '14,917', '12,565', '9,312', '8,147', '6,779', '5,328', '4,048', '2,587', '1,862', '1,368', '1,274', '816', '2,010']

문자열 자료형이고 숫자 사이에 쉼표 (,)가 있습니다.

❖ Step 2) 궁금한 지역의 이름 입력 받기

```
import csv

f = open('age.csv', encoding='cp949')
data = csv.reader(f)

header = next(data)
# print(header)

name = input('인구 구조가 알고 싶은 지역의 이름(읍면동 단위)을 입력해주세요: ')
for row in data:
    print(row)


f.close()
```

❖ Step 3) 궁금한 지역의 인구 구조를 리스트에 저장하기

```
import csv

f = open('age.csv', encoding='cp949')
data = csv.reader(f)

header = next(data)
# print(header)
home = []  # 입력받은 지역의 데이터를 저장할 리스트 생성
name = input('인구 구조가 알고 싶은 지역의 이름(읍면동 단위)을 입력해주세요: ')
for row in data:
    if name in row[0]:
        for i in row[3:]:
            home.append(int(i))

f.close()
print(home)
```

리스트가 아니라 numpy를 활용해 보겠습니다.

```
인구 구조가 알고 싶은 지역의 이름(읍면동 단위)을 입력해주세요: 신도림
[274, 270, 319, 304, 321, 377, 426, 403, 374, 391, 408, 384, 370, 400, 354, 322, 332, 311, 328, 353, 359, 422, 355, 370, 429, 45
1, 456, 441, 454, 486, 501, 472, 561, 619, 636, 612, 619, 664, 701, 687, 760, 691, 712, 637, 628, 602, 626, 636, 663, 600, 616, 5
79, 567, 511, 462, 487, 509, 472, 426, 479, 477, 484, 453, 457, 412, 370, 450, 368, 340, 338, 229, 278, 249, 248, 265, 161, 141,
134, 135, 154, 121, 107, 97, 94, 71, 69, 58, 53, 35, 30, 23, 14, 16, 13, 11, 8, 8, 4, 4, 2, 4]
```

❖ Step 3) 궁금한 지역의 인구 구조를 numpy array에 저장하기

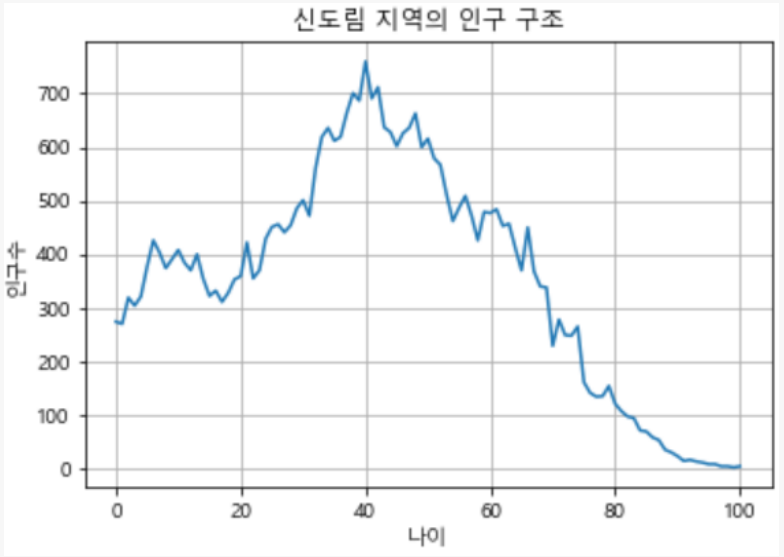
```
import csv
import numpy as np

f = open('age.csv', encoding='cp949')
data = csv.reader(f)

header = next(data)
# print(header)

name = input('인구 구조가 알고 싶은 지역의 이름(읍면동 단위)을 입력해주세요: ')
for row in data:
    if name in row[0]:
        home = np.array(row[3:], dtype=int)

f.close()
print(home)
```

문자열 자료형을 정수형 자료형으로
변환하기 위해서 dtype 속성을 int로 설정해 줍니다.

```
인구 구조가 알고 싶은 지역의 이름(읍면동 단위)을 입력해주세요: 신도림
[274 270 319 304 321 377 426 403 374 391 408 384 370 400 354 322 332 311
 328 353 359 422 355 370 429 451 456 441 454 486 501 472 561 619 636 612
 619 664 701 687 760 691 712 637 628 602 626 636 663 600 616 579 567 511
 462 487 509 472 426 479 477 484 453 457 412 370 450 368 340 338 229 278
 249 248 265 161 141 134 135 154 121 107  97  94  71  69  58  53  35  30
  23  14  16  13  11   8   8   4   4   2   4]
```

❖ Step 3) 궁금한 지역의 인구 구조 시각화하기

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

f = open('age.csv', encoding='cp949')
data = csv.reader(f)

header = next(data)
# print(header)

name = input('인구 구조가 알고 싶은 지역의 이름(읍면동 단위)을 입력해주세요: ')
for row in data:
    if name in row[0]:
        home = np.array(row[3:], dtype=int)

f.close()

plt.rc('font', family='Malgun Gothic')
plt.plot(home)
plt.title(name + ' 지역의 인구 구조')
plt.grid(True)
plt.xlabel('나이')
plt.ylabel('인구수')
plt.show()
```




```
인구 구조가 알고 싶은 지역의 이름(읍면동 단위)을 입력해주세요: 신도림
```

❖ Step 4) 궁금한 지역의 인구 구조와 가장 비슷한 인구 구조를 가진 지역 찾기

- 내가 알고자 하는 궁금한 지역: A
- 비교할 지역(= A지역을 제외한 나머지 지역): B

A와 B의 인구 구조가 비슷하다는 것을
어떻게 알 수 있을까요?

❖ Step 4) 궁금한 지역의 인구 구조와 가장 비슷한 인구 구조를 가진 지역 찾기

알고리즘 버전 1

ⓐ 전국의 모든 지역 중 한 곳(B)을 선택한다.
ⓑ 궁금한 지역 A의 0세 **인구수**에서 B의 0세 **인구수**를 뺀다.
ⓒ ⓑ를 "100세 이상 인구수"에 해당하는 값까지 반복한 후 각각의 차이를 모두 더한다.
ⓓ 전국의 모든 지역에 대해 반복하며 그 차이가 가장 작은 지역을 찾는다.

비슷한 인구 구조를 찾기 위한 간단한 방법으로,
우선 두 지역의 연령별 인구수 차이를 구해 모두 더해보면 어떨까요?

❖ Step 4) 궁금한 지역의 인구 구조와 가장 비슷한 인구 구조를 가진 지역 찾기

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

f = open('age.csv', encoding='cp949')
data = csv.reader(f)

header = next(data)
# print(header)

name = input('인구 구조가 알고 싶은 지역의 이름(읍면동 단위)을 입력해주세요: ')
for row in data:
    if name in row[0]:
        home = np.array(row[3:], dtype=int)
    if '화촌면' in row[0]:
        homeB = np.array(row[3:], dtype=int)

f.close()

plt.rc('font', family='Malgun Gothic')
plt.plot(home, label=name)
plt.plot(homeB, label='화촌면')
plt.title(name + ' 지역의 인구 구조')
plt.grid(True)
plt.xlabel('나이')
plt.ylabel('인구수')
plt.legend()
plt.show()
```

인구 구조가 알고 싶은 지역의 이름(읍면동 단위)을 입력해주세요: 신도림

알고리즘 버전 1

✓ 전체 인구수가 다른 두 지역에서 연령별 인구수 차이를 구하는 형태의 접근 방법은 문제가 있을 것 같습니다.

인구수 차이가 아니라 인구 비율을 고려해 보겠습니다.

❖ Step 4) 궁금한 지역의 인구 구조와 가장 비슷한 인구 구조를 가진 지역 찾기

알고리즘 버전 2 ("인구수" → "인구 비율"로 변경)

ⓐ 전국의 모든 지역 중 한 곳(B)을 선택한다.
ⓑ 궁금한 지역 A의 0세 **인구 비율**에서 B의 0세 **인구 비율**을 뺀다.
ⓒ ⓑ를 "100세 이상 인구수"에 해당하는 값까지 반복한 후 각각의 차이를 모두 더한다.
ⓓ 전국의 모든 지역에 대해 반복하며 그 차이가 가장 작은 지역을 찾는다.

❖ Step 4) 궁금한 지역의 인구 구조와 가장 비슷한 인구 구조를 가진 지역 찾기

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

f = open('age.csv', encoding='cp949')
data = csv.reader(f)

header = next(data)
# print(header)

name = input('인구 구조가 알고 싶은 지역의 이름(읍면동 단위)을 입력해주세요: ')
for row in data:
    for i in range(102):
        row[i + 2] = int(row[i + 2].replace(',', ""))
    if name in row[0]:
        home = np.array(row[3:], dtype=int) / row[2]
    if '화촌면' in row[0]:
        away = np.array(row[3:], dtype=int) / row[2]

f.close()

plt.rc('font', family='Malgun Gothic')
plt.plot(home, label=name)
plt.plot(away, label='화촌면')
plt.title(name+' 지역과 화촌면 지역의 인구 구조 비교')
plt.grid(True)
plt.xlabel('나이')
plt.ylabel('인구 비율')
plt.legend()
plt.show()
```

인구 구조가 알고 싶은 지역의 이름(읍면동 단위)을 입력해주세요: 신도림

전체 인구수가 다르더라도
인구 구조를 비교할 수 있을 것 같습니다.

❖ Step 4) 궁금한 지역의 인구 구조와 가장 비슷한 인구 구조를 가진 지역 찾기

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

f = open('age.csv', encoding='cp949')
data = csv.reader(f)

header = next(data)
# print(header)

name = input('인구 구조가 알고 싶은 지역의 이름(읍면동 단위)을 입력해주세요: ')
for row in data:
    for i in range(102):
        row[i + 2] = int(row[i + 2].replace(',', ""))
    if name in row[0]:
        home = np.array(row[3:], dtype=int) / row[2]

for row in data:
    print(row)

f.close()
```

아무것도 출력되지 않습니다!

```
인구 구조가 알고 싶은 지역의 이름(읍면동 단위)을 입력해주세요: 신도림
```

❖ Step 4) 궁금한 지역의 인구 구조와 가장 비슷한 인구 구조를 가진 지역 찾기

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

f = open('age.csv', encoding='cp949')
data = csv.reader(f)

header = next(data)
# print(header)
print(type(data))
data = list(data)
print(type(data))
name = input('인구 구조가 알고 싶은 지역의 이름(읍면동 단위)을 입력해주세요: ')
for row in data:
    for i in range(102):
        row[i + 2] = int(row[i + 2].replace(',', ""))
    if name in row[0]:
        home = np.array(row[3:], dtype=int) / row[2]

for row in data:
    print(row)

f.close()
```

data의 자료형을 확인하고,
리스트로 저장하겠습니다.

❖ Step 4) 궁금한 지역의 인구 구조와 가장 비슷한 인구 구조를 가진 지역 찾기

```
<class '_csv.reader'>
<class 'list'>
인구 구조가 알고 싶은 지역의 이름(읍면동 단위)을 입력해주세요: 신도림
['서울특별시  (1100000000)', '9,520,880', 9520880, 43492, 45054, 49318, 51995, 56210, 63757, 67763, 66950, 67744, 74276, 72335,
72732, 69959, 74746, 78816, 72760, 71164, 79106, 81199, 85628, 99222, 116143, 118167, 130171, 141973, 152695, 160467, 167496, 1
68785, 171275, 163601, 150119, 143789, 140390, 135790, 134809, 135124, 135240, 146723, 155189, 154986, 155424, 149681, 131846,
139332, 136958, 141983, 153857, 158331, 160766, 170686, 164108, 163523, 158165, 146324, 136933, 143792, 139146, 139431, 145596,
153238, 157332, 143607, 135335, 134397, 123145, 129445, 108604, 96292, 103732, 72522, 81543, 81820, 79906, 80925, 63365, 57788,
57285, 56204, 64534, 49960, 42822, 40456, 34745, 30034, 26040, 23278, 18473, 14917, 12565, 9312, 8147, 6779, 5328, 4048, 2587,
1862, 1368, 1274, 816, 2010]
['서울특별시 종로구 (1111000000)', '145,073', 145073, 486, 475, 557, 576, 664, 773, 848, 859, 832, 969, 909, 994, 956, 1011, 11
06, 1002, 951, 1098, 1109, 1329, 1607, 1914, 1962, 2164, 2382, 2574, 2595, 2548, 2597, 2553, 2369, 2210, 1929, 1921, 1856, 171
7, 1813, 1735, 1944, 1979, 2129, 2081, 2057, 1800, 1870, 1928, 2063, 2226, 2414, 2497, 2779, 2705, 2603, 2578, 2466, 2253, 240
1, 2381, 2397, 2419, 2545, 2515, 2221, 2104, 2127, 1962, 2052, 1699, 1522, 1615, 1101, 1231, 1295, 1286, 1345, 1038, 1034, 106
8, 1058, 1203, 1000, 857, 866, 740, 617, 553, 502, 421, 324, 266, 208, 183, 139, 114, 104, 71, 45, 28, 25, 27, 42]
['서울특별시 종로구 청운효자동(1111051500)', '12,006', 12006, 43, 48, 58, 54, 80, 74, 96, 93, 83, 120, 93, 106, 124, 112, 137,
122, 97, 131, 112, 117, 111, 160, 147, 160, 133, 189, 155, 158, 168, 194, 158, 145, 147, 153, 144, 150, 171, 150, 170, 195, 21
9, 207, 223, 181, 175, 213, 216, 199, 245, 212, 218, 227, 214, 210, 208, 178, 169, 165, 163, 171, 184, 170, 160, 143, 139, 125,
141, 103, 114, 122, 77, 105, 99, 86, 111, 91, 89, 92, 91, 92, 81, 69, 77, 66, 57, 50, 42, 32, 31, 20, 18, 10, 10, 10, 7, 7, 3,
```

❖ Step 4) 궁금한 지역의 인구 구조와 가장 비슷한 인구 구조를 가진 지역 찾기

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

f = open('age.csv', encoding='cp949')
data = csv.reader(f)

header = next(data)
# print(header)

data = list(data)

name = input('인구 구조가 알고 싶은 지역의 이름(읍면동 단위)을 입력해주세요: ')
for row in data:
    for i in range(102):
        row[i + 2] = int(row[i + 2].replace(',', ""))
    if name in row[0]:
        home = np.array(row[3:], dtype=int) / row[2]

for row in data:
    if name not in row[0]:
        away = np.array(row[3:], dtype=int) / row[2]
        print(home - away)

f.close()
```

❖ Step 4) 궁금한 지역의 인구 구조와 가장 비슷한 인구 구조를 가진 지역 찾기

```
인구 구조가 알고 싶은 지역의 이름(읍면동 단위)을 입력해주세요: 신도림
[ 2.72476195e-03  2.79721999e-03  2.68719454e-03  3.48993570e-03
  2.58605815e-03  2.69475574e-03  4.00021626e-03  4.35974502e-03
  3.73453696e-03  2.91696281e-03  3.30451059e-03  3.41820368e-03
  3.13772438e-03  2.90957339e-03  2.65058498e-03  9.83168793e-04
  1.44689028e-03  1.20471624e-03  3.90665219e-04  3.05165843e-04
 -2.73040621e-04 -4.18974202e-04 -1.98001130e-03 -3.61947163e-03
 -3.63373244e-03 -3.41814396e-03 -3.63786054e-03 -5.37212786e-03
 -4.09171436e-03 -4.62433765e-03 -3.06964799e-03 -2
 -2.09031653e-03  2.41421118e-03  2.79440253e-03
  3.04069586e-03  3.10826067e-03  3.73352599e-03
  2.78017492e-03  4.20100266e-03  2.85688771e-03  3.81628830e-03
  3.81986548e-03  3.08491807e-03  1.06902116e-03  3.09158188e-03
  1.28396837e-03  1.14399074e-03  8.26797251e-05 -1.53594923e-03
 -4.90573453e-04 -1.19717965e-03 -3.23720824e-03 -1.67419879e-03
 -9.77390972e-04 -6.36317720e-04 -2.36676399e-03 -2.94418749e-03
 -1.71271483e-03 -2.95085110e-03 -2.75996348e-03 -2.37675199e-03
 -1.98043277e-03 -2.51026174e-03 -1.18425915e-03 -1.80279738e-03
 -8.90148488e-04 -6.66029540e-04 -1.50497706e-03 -1.23503117e-03
                                 ⋮
```

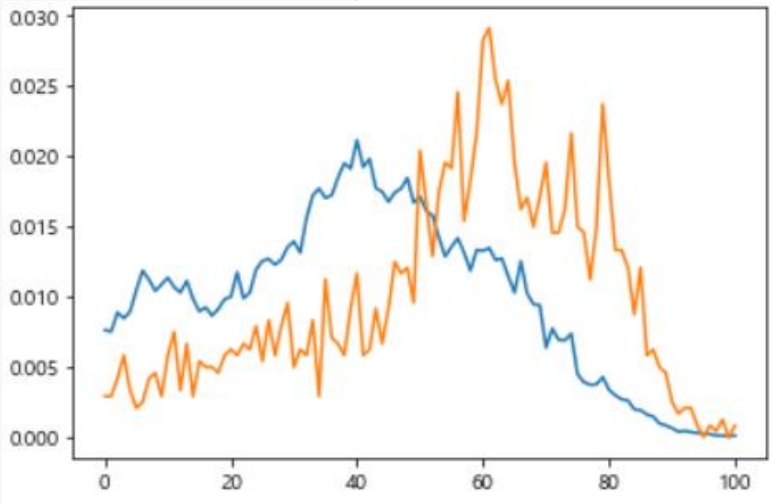
3.05066470e−03은 $3.05066470 \times 10^{-3}$이라는 의미로 0.00305066470을 의미합니다.

❖ Step 4) 궁금한 지역의 인구 구조와 가장 비슷한 인구 구조를 가진 지역 찾기

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

f = open('age.csv', encoding='cp949')
data = csv.reader(f)

header = next(data)
# print(header)

data = list(data)

min_val = 1          # 최소값을 저장할 변수 생성 및 초기화
result_name = ''     # 최소값을 갖는 지역의 이름을 저장할 변수 생성 및 초기화
result = 0           # 최소값을 갖는 지역의 연령대별 인구 비율을 저장할 배열 생성 및 초기화

name = input('인구 구조가 알고 싶은 지역의 이름(읍면동 단위)을 입력해주세요: ')
for row in data:
    for i in range(102):
        row[i + 2] = int(row[i + 2].replace(',', ""))
    if name in row[0]:
        home = np.array(row[3:], dtype=int) / row[2]

for row in data:
    if name not in row[0]:
        away = np.array(row[3:], dtype=int) / row[2]
        s = np.sum(home - away)
        if s < min_val:
            min_val = s
            result_name = row[0]
            result = away

f.close()
```

❖ Step 5) 가장 비슷한 곳의 인구 구조와 궁금한 지역의 인구 구조 시각화하기

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

f = open('age.csv', encoding='cp949')
data = csv.reader(f)

header = next(data)
# print(header)

data = list(data)

min_val = 1          # 최소값을 저장할 변수 생성 및 초기화
result_name = ''     # 최소값을 갖는 지역의 이름을 저장할 변수 생성 및 초기화
result = 0           # 최소값을 갖는 지역의 연령대별 인구 비율을 저장할 배열 생성 및 초기화

name = input('인구 구조가 알고 싶은 지역의 이름(읍면동 단위)을 입력해주세요: ')
for row in data:
    for i in range(102):
        row[i + 2] = int(row[i + 2].replace(',', ""))
    if name in row[0]:
        home = np.array(row[3:], dtype=int) / row[2]

for row in data:
    if name not in row[0]:
        away = np.array(row[3:], dtype=int) / row[2]
        s = np.sum(home - away)
        if s < min_val:
            min_val = s
            result_name = row[0]
            result = away

f.close()

plt.plot(home)
plt.plot(result)
plt.show()
```



인구 구조가 비슷해 보이지 않는데
알고리즘에 문제가 있었던 걸까요?

❖ Step 5) 가장 비슷한 곳의 인구 구조와 궁금한 지역의 인구 구조 시각화하기

알고리즘 버전 2 ("인구수" → "인구 비율"로 변경)

ⓐ 전국의 모든 지역 중 한 곳(B)을 선택한다.
ⓑ 궁금한 지역 A의 0세 **인구 비율**에서 B의 0세 **인구 비율**을 뺀다.
ⓒ ⓑ를 "100세 이상 인구수"에 해당하는 값까지 반복한 후 각각의 차이를 모두 더한다.
ⓓ 전국의 모든 지역에 대해 반복하며 그 차이가 가장 작은 지역을 찾는다.

문제 원인을 발견했습니다!

예를 들어

✓ ⓑ에서 0세에 대해 계산한 결과가 음수이고, 1세에 대해 계산한 결과가 양수였다고 가정해 보겠습니다.
✓ 그러면 ⓒ에서 차이를 더하는 과정에서 **음수와 양수가 상쇄되는 현상이 발생**합니다.

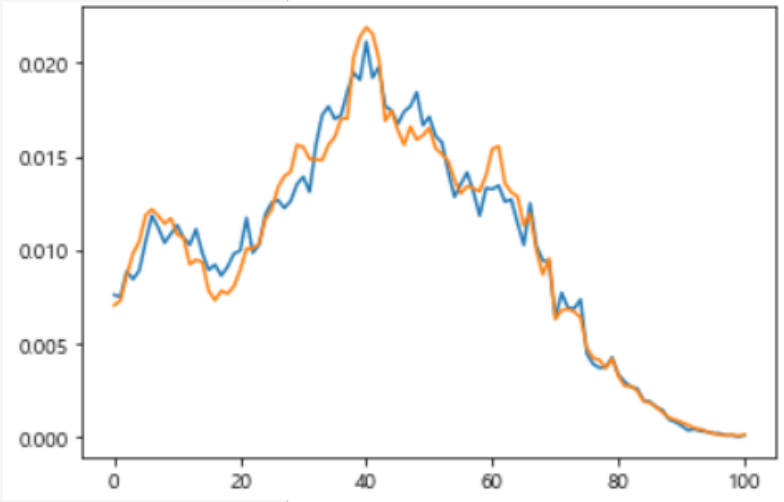
❖ Step 5) 가장 비슷한 곳의 인구 구조와 궁금한 지역의 인구 구조 시각화하기

알고리즘 버전 3 ("차이의 합" → "차이의 제곱의 합"으로 변경)

ⓐ 전국의 모든 지역 중 한 곳 (B)을 선택한다.
ⓑ 궁금한 지역 A의 0세 인구 비율에서 B의 0세 인구 비율을 뺀다.
ⓒ ⓑ를 "100세 이상 인구수"에 해당하는 값까지 반복한 후 각각의 **차이의 제곱**의 합을 구한다.
ⓓ 전국의 모든 지역에 대해 반복하며 그 차이가 가장 작은 지역을 찾는다.

❖ Step 5) 가장 비슷한 곳의 인구 구조와 궁금한 지역의 인구 구조 시각화하기

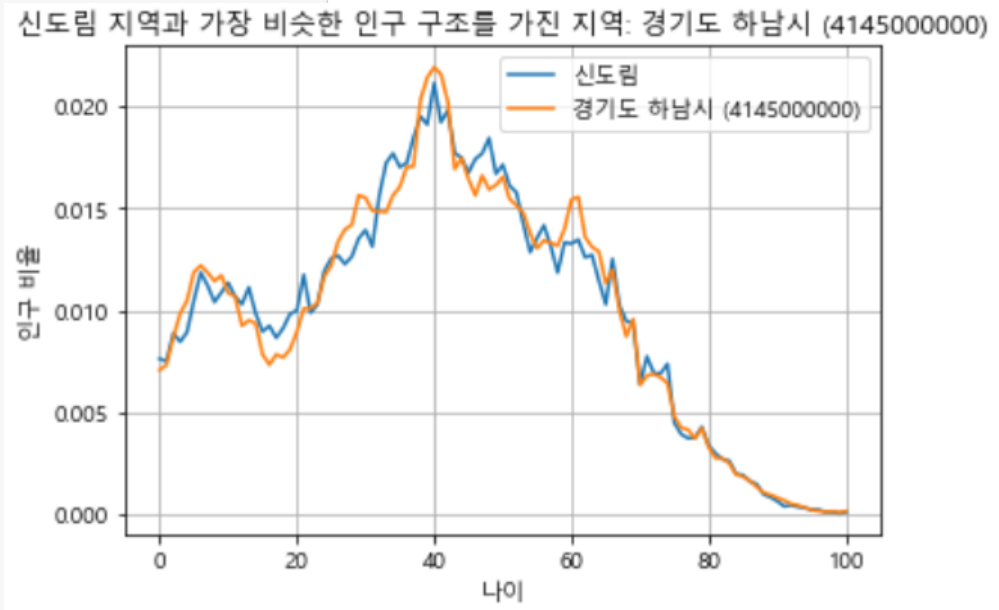
```
import csv
import numpy as np
import matplotlib.pyplot as plt

f = open('age.csv', encoding='cp949')
data = csv.reader(f)

header = next(data)
# print(header)

data = list(data)

min_val = 1          # 최소값을 저장할 변수 생성 및 초기화
result_name = ''     # 최소값을 갖는 지역의 이름을 저장할 변수 생성 및 초기화
result = 0           # 최소값을 갖는 지역의 연령대별 인구 비율을 저장할 배열 생성 및 초기화

name = input('인구 구조가 알고 싶은 지역의 이름(읍면동 단위)을 입력해주세요: ')
for row in data:
    for i in range(102):
        row[i + 2] = int(row[i + 2].replace(',', ""))
    if name in row[0]:
        home = np.array(row[3:], dtype=int) / row[2]

for row in data:
    if name not in row[0]:
        away = np.array(row[3:], dtype=int) / row[2]
        s = np.sum((home - away)**2)   # (home-away) * (home-away)와 같습니다.
        if s < min_val:
            min_val = s
            result_name = row[0]
            result = away

f.close()

plt.plot(home)
plt.plot(result)
plt.show()
```



인구 구조가 비슷해 보이네요!

❖ Step 5) 가장 비슷한 곳의 인구 구조와 궁금한 지역의 인구 구조 시각화하기

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

f = open('age.csv', encoding='cp949')
data = csv.reader(f)

header = next(data)
# print(header)

data = list(data)

min_val = 1          # 최소값을 저장할 변수 생성 및 초기화
result_name = ''     # 최소값을 갖는 지역의 이름을 저장할 변수 생성 및 초기화
result = 0           # 최소값을 갖는 지역의 연령대별 인구 비율을 저장할 배열 생성 및 초기화

name = input('인구 구조가 알고 싶은 지역의 이름(읍면동 단위)을 입력해주세요: ')
for row in data:
    for i in range(102):
        row[i + 2] = int(row[i + 2].replace(',', ""))
    if name in row[0]:
        home = np.array(row[3:], dtype=int) / row[2]

for row in data:
    if name not in row[0]:
        away = np.array(row[3:], dtype=int) / row[2]
        s = np.sum((home - away)**2)   # (home-away) * (home-away)와 같습니다.
        if s < min_val:
            min_val = s
            result_name = row[0]
            result = away

f.close()

plt.rc('font', family='Malgun Gothic')
plt.title(name + ' 지역과 가장 비슷한 인구 구조를 가진 지역: ' + result_name)
plt.plot(home, label = name)
plt.plot(result, label = result_name)
plt.xlabel('나이')
plt.ylabel('인구 비율')
plt.grid(True)
plt.legend()
plt.show()
```



끝맺음

- 01. numpy를 활용한 나만의 프로젝트 만들기
- 02. 알고리즘을 코드로 표현하기

THANK YOU!
Q & A

- Name: 강환수
- Office: 동양미래대학교 2호관 706호 (02-2610-1941)
- E-mail: hsknag@dongyang.ac.kr
- Homepage: https://github.com/ai7dnn/2023-DA