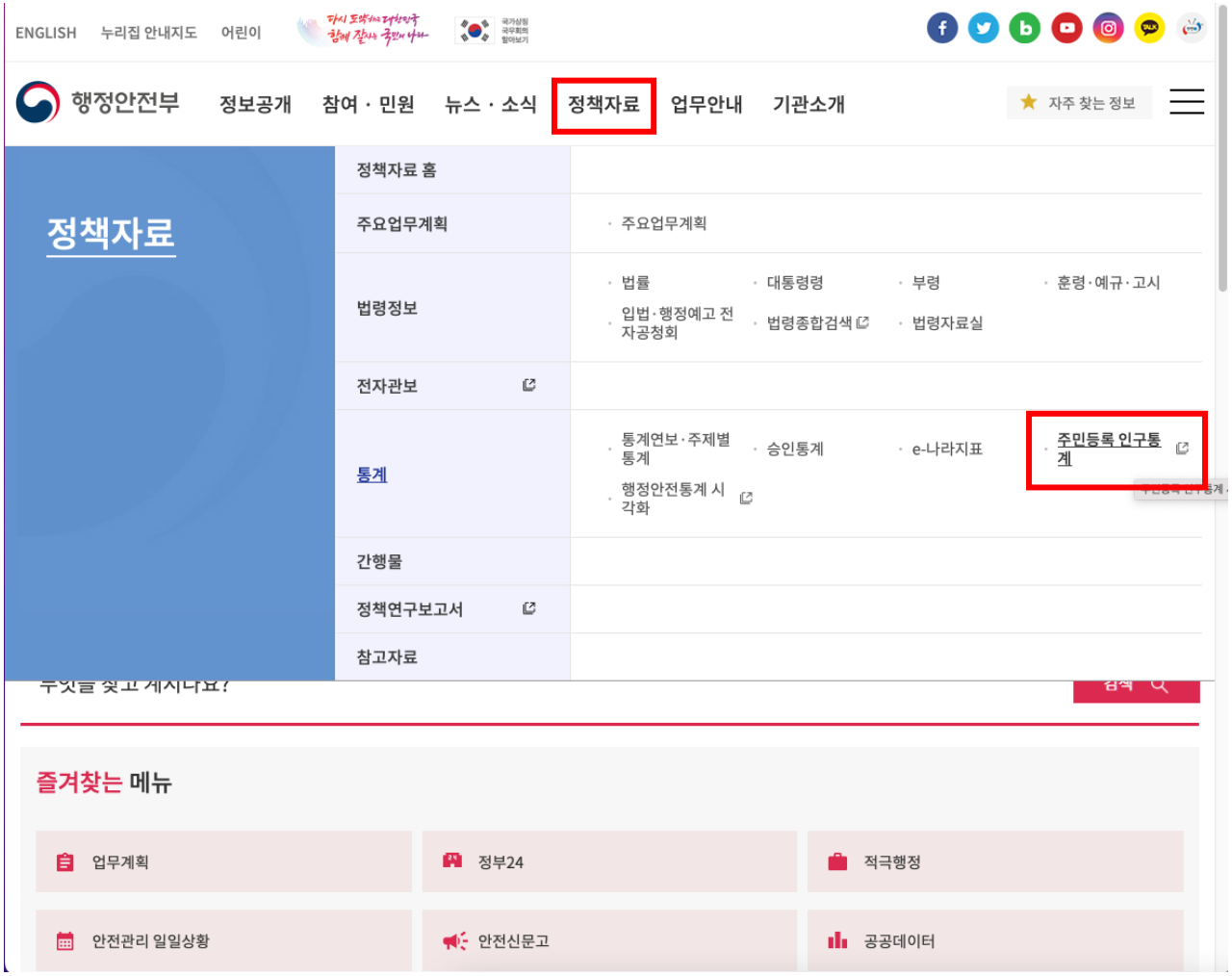
인구 데이터 분석

공공데이터 분서

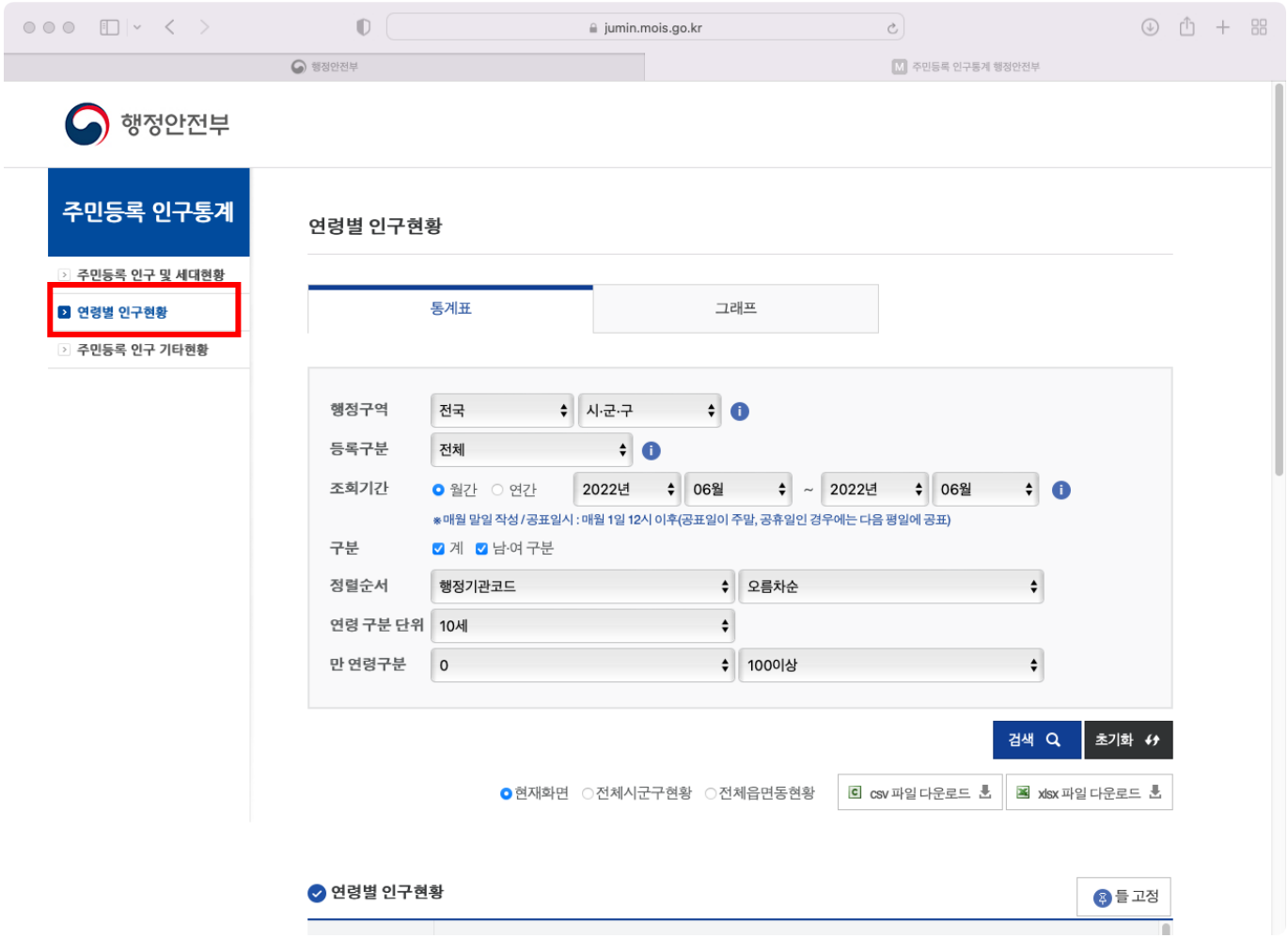
인구 공공데이터 내려 받기 #1

- 행정안전부 홈페이지: https://www.mois.go.kr
  - 정책자료 > 주민등록 인구 통계 메뉴 선택



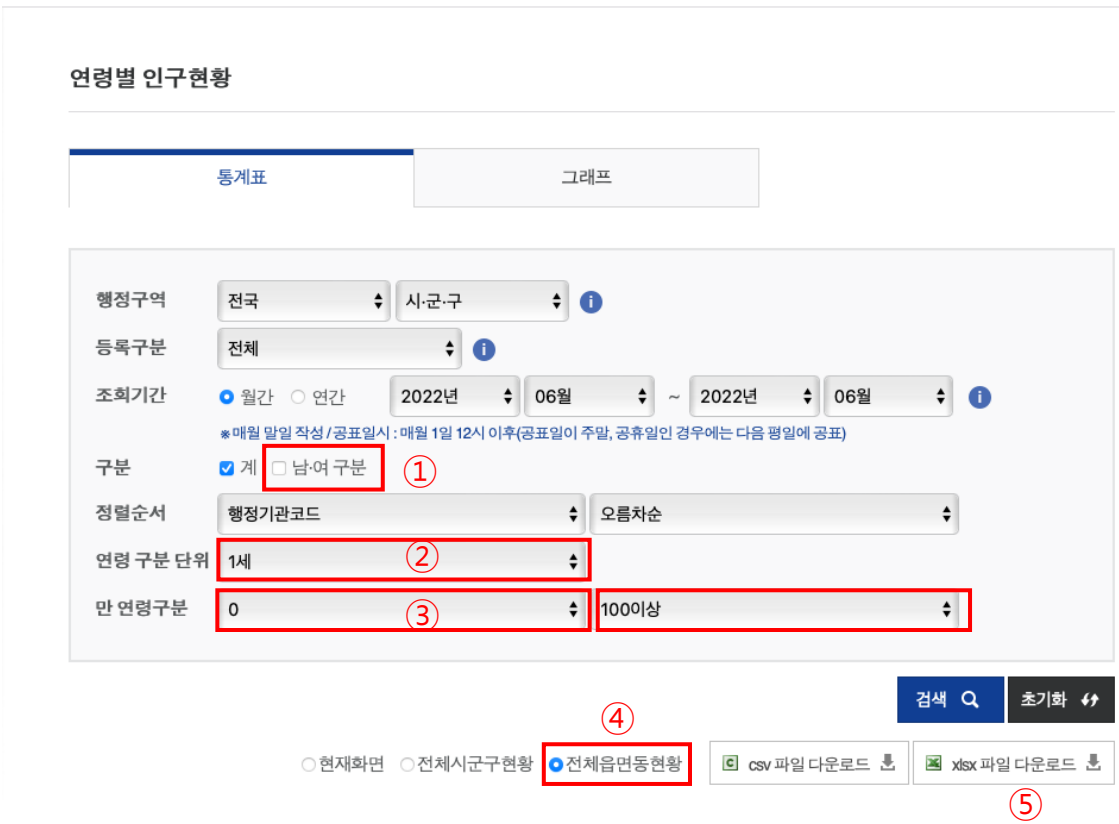
인구 공공데이터 내려 받기 #2

- 주민등록 인구통계
  - 연령별 인구현황 선택



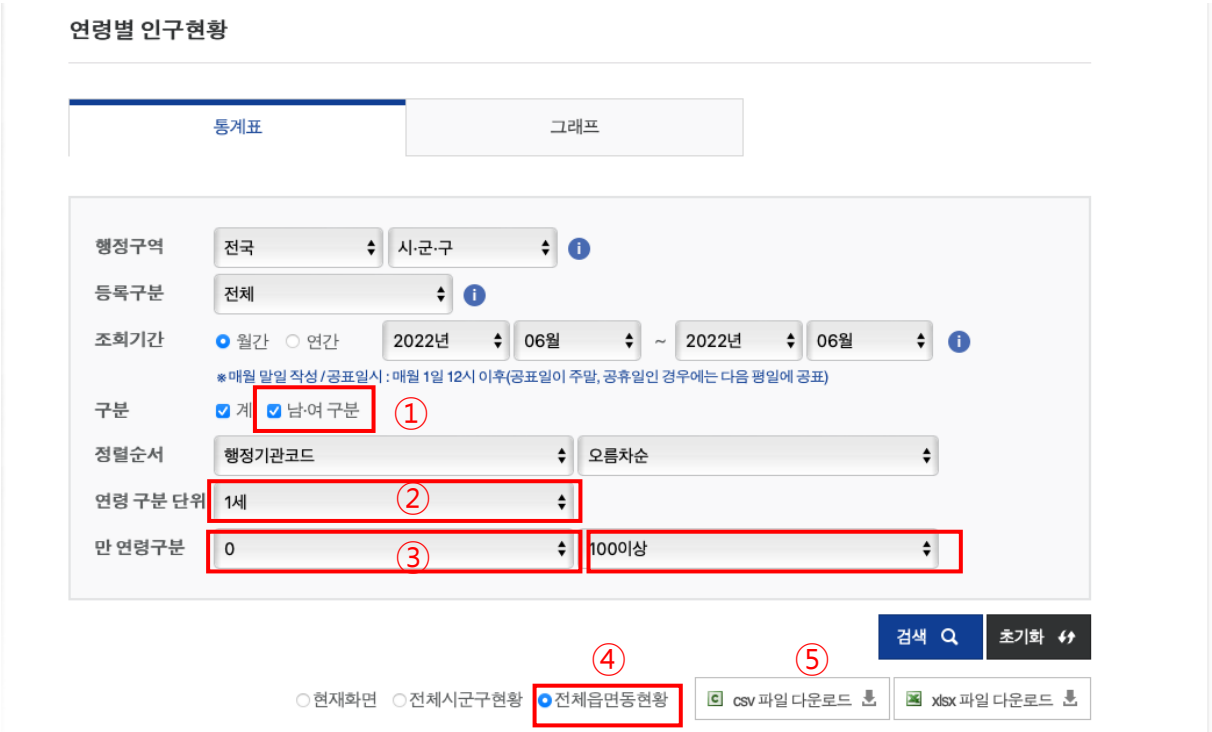
연령별 인구 현황

- 통계표
  - 조회기간: 월간 2022년 6월 ~ 2022년 6월
  - 구분: **남·여구분 해제**
  - 연령 구분 단위: **1세**
  - 만 연령구분: **0 ~ 100이상**
  - **전체 읍면동 현황** 선택
  - xlsx 파일 다운로드
    - **age.xlsx** 로 저장



남녀 성별이 포함된 데이터

- 연령별 성별 인구현황
  - 조회기간: 2022년 6월
  - 구분: **남·여구분 선택**
  - 연령 구분 단위: **1세**
  - 만 연령구분: **0 ~ 100이상**
  - **전체 읍면동 현황** 선택
  - csv 파일 다운로드
    - **gender.csv** 로 저장



전국 읍면동, 연령별 인구 현황

- age.csv 파일:
  - 데이터에 남·여 구분 없음
- 헤더 정보
  - **A열**: 행정구역(행정 구역 이름과 10자리 행정구역 코드)
  - **B~C열**: 총인구수 및 연령구간 인구수
  - **D열 ~ CZ열**: 연령별 인구 수(0세 … 99세, 100세 이상)

	A	B	C	D	E	F	G	H
1	행정구역	2022년06월_계_총인구수	2022년06월_계_연령구간인구수	2022년06월_계_0세	2022년06월_계_1세	2022년06월_계_2세	2022년06월_계_3세	2022년06월_계_4세
2	서울특별시 (1100000000)	9,494,807	9,494,807	41,427	44,825	46,252	49,895	53,002
3	서울특별시 종로구 (1111000000)	143,624	143,624	449	479	542	545	594
4	서울특별시 종로구 청운효자동(11	11,803	11,803	43	46	51	50	54
5	서울특별시 종로구 사직동(11110	9,310	9,310	31	31	36	48	48
6	서울특별시 종로구 삼청동(11110	2,386	2,386	3	7	4	4	12
7	서울특별시 종로구 부암동(11110	9,432	9,432	26	22	33	26	45
8	서울특별시 종로구 평창동(11110	17,963	17,963	65	79	102	91	115
9	서울특별시 종로구 무악동(11110	8,150	8,150	35	41	41	45	60
10	서울특별시 종로구 교남동(11110	9,947	9,947	51	71	79	81	68
11	서울특별시 종로구 가회동(11110	4,026	4,026	16	11	8	16	12
12	서울특별시 종로구 종로1.2.3.4가	7,199	7,199	11	17	18	22	18
13	서울특별시 종로구 종로5.6가동(1	5,056	5,056	8	8	4	6	11
14	서울특별시 종로구 이화동(11110	7,231	7,231	19	14	22	20	14
15	서울특별시 종로구 혜화동(11110	16,369	16,369	39	32	46	50	40
16	서울특별시 종로구 창신제1동(111	4,857	4,857	7	14	11	9	13
17	서울특별시 종로구 창신제2동(111	7,837	7,837	15	14	19	18	21
18	서울특별시 종로구 창신제3동(111	6,496	6,496	33	30	36	25	27
19	서울특별시 종로구 숭인제1동(111	5,835	5,835	18	16	16	19	17
20	서울특별시 종로구 숭인제2동(111	9,727	9,727	29	26	16	15	19
21	서울특별시 중구 (1114000000)	122,088	122,088	556	565	570	543	564
22	서울특별시 중구 소공동(1114052	2,237	2,237	17	12	8	11	20
23	서울특별시 중구 회현동(1114054	4,733	4,733	17	13	17	11	15
24	서울특별시 중구 명동(111405500	2,889	2,889	6	5	7	7	6
25	서울특별시 중구 필동(111405700	4,036	4,036	15	13	7	11	15

대구 산격동 인구 현황

```
import csv
f = open('age.csv', encoding='euc_kr')
data = csv.reader(f)

header = next(data)
print(header)
# row[0]: 행정구역
for row in data:
    if '산격3' in row[0]: # '산격3'이 포함된 자료만 출력
        print(row)
f.close()
```

모두 문자열 형태로 저장되어 있음

```
['행정구역', '2022년06월_계_총인구수', '2022년06월_계_연령구간인구수', '2022년06월_계_0세', '2022년06월_계_1세', '2022년06월_계_2세', '2022년06월_계_3세', '2022년06월_계_4세', '2022년06월_계_5세', '2022년06월_계_6세', '2022년06월_계_7세', '2022년06월_계_8세', '2022년06월_계_9세', '2022년06월_계_10세', '2022년06월_계_11세', '2022년06월_계_12세', '2022년06월_계_13세', '2022년06월_계_14세', '2022년06월_계_15세', '2022년06월_계_16세', '2022년06월_계_17세', '2022년06월_계_18세', '2022년06월_계_19세', '2022년06월_계_20세', '2022년06월_계_21세', '2022년06월_계_22세', '2022년06월_계_23세', '2022년06월_계_24세', '2022년06월_계_25세', '2022년06월_계_26세', '2022년06월_계_27세', '2022년06월_계_28세', '2022년06월_계_29세', '2022년06월_계_30세', '2022년06월_계_31세', '2022년06월_계_32세', '2022년06월_계_33세', '2022년06월_계_34세', '2022년06월_계_35세', '2022년06월_계_36세', '2022년06월_계_37세', '2022년06월_계_38세', '2022년06월_계_39세', '2022년06월_계_40세', '2022년06월_계_41세', '2022년06월_계_42세', '2022년06월_계_43세', '2022년06월_계_44세', '2022년06월_계_45세', '2022년06월_계_46세', '2022년06월_계_47세', '2022년06월_계_48세', '2022년06월_계_49세', '2022년06월_계_50세', '2022년06월_계_51세', '2022년06월_계_52세', '2022년06월_계_53세', '2022년06월_계_54세', '2022년06월_계_55세', '2022년06월_계_56세', '2022년06월_계_57세', '2022년06월_계_58세', '2022년06월_계_59세', '2022년06월_계_60세', '2022년06월_계_61세', '2022년06월_계_62세', '2022년06월_계_63세', '2022년06월_계_64세', '2022년06월_계_65세', '2022년06월_계_66세', '2022년06월_계_67세', '2022년06월_계_68세', '2022년06월_계_69세', '2022년06월_계_70세', '2022년06월_계_71세', '2022년06월_계_72세', '2022년06월_계_73세', '2022년06월_계_74세', '2022년06월_계_75세', '2022년06월_계_76세', '2022년06월_계_77세', '2022년06월_계_78세', '2022년06월_계_79세', '2022년06월_계_80세', '2022년06월_계_81세', '2022년06월_계_82세', '2022년06월_계_83세', '2022년06월_계_84세', '2022년06월_계_85세', '2022년06월_계_86세', '2022년06월_계_87세', '2022년06월_계_88세', '2022년06월_계_89세', '2022년06월_계_90세', '2022년06월_계_91세', '2022년06월_계_92세', '2022년06월_계_93세', '2022년06월_계_94세', '2022년06월_계_95세', '2022년06월_계_96세', '2022년06월_계_97세', '2022년06월_계_98세', '2022년06월_계_99세', '2022년06월_계_100세 이상']
['대구광역시 북구 산격3동(2723063000)', '9,499', '9,499', '13', '9', '19', '26', '22', '26', '24', '35', '16', '33', '30', '38', '37', '31', '38', '40', '35', '49', '127', '179', '193', '250', '345', '371', '351', '339', '298', '268', '231', '226', '205', '161', '158', '123', '90', '87', '81', '85', '75', '83', '106', '87', '112', '100', '68', '98', '92', '105', '117', '115', '109', '104', '131', '117', '114', '137', '124', '116', '108', '133', '120', '171', '158', '127', '135', '109', '129', '118', '102', '107', '103', '79', '98', '95', '84', '82', '44', '67', '55', '74', '46', '53', '61', '35', '39', '32', '39', '19', '18', '12', '10', '12', '8', '9', '2', '3', '0', '3', '0', '0', '1']
```

인구수 출력

- age.csv 데이터 헤더 (연령 구분: 1세)

[0]	[1]	[2]	[3]	[4]	...		[103]
행정구역	2022년 6월 총 인구수	2022년 06월 연령 구간 인구수	0세	1세	...	99세	100세 이상

- 산격3동 (경북대 인근)의 인구 데이터 출력

```
import csv
f = open('age.csv', encoding='euc_kr')
data = csv.reader(f)
header = next(data)
# row[0]: 행정구역
result = []
for row in data:
    if '산격3' in row[0]: # '산격3'이 포함된 자료 중에서
        for data in row[3:]: # '0세 ~ 100세 이상까지 자료만 리스트에 추가
            result.append(data)
print(result)
f.close()
```

슬라이싱: [start : end : step]

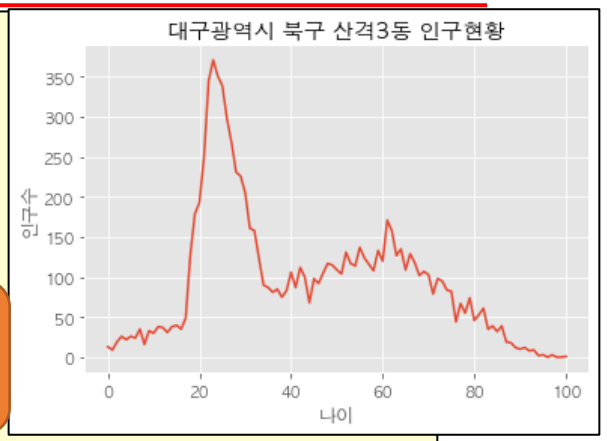
```
['13', '9', '19', '26', '22', '26', '24', '35', '16', '33', '30', '38', '37', '31', '38', '40', '35',
'49', '127', '179', '193', '250', '345', '371', '351', '339', '298', '268', '231', '226', '205',
'161', '158', '123', '90', '87', '81', '85', '75', '83', '106', '87', '112', '100', '68', '98', '92',
'105', '117', '115', '109', '104', '131', '117', '114', '137', '124', '116', '108', '133', '120',
'171', '158', '127', '135', '109', '129', '118', '102', '107', '103', '79', '98', '95', '84', '82',
'44', '67', '55', '74', '46', '53', '61', '35', '39', '32', '39', '19', '18', '12', '10', '12', '8',
'9', '2', '3', '0', '3', '0', '0', '1']
```

대구시 산격3동의 인구 분포 그래프 그리기

```
import csv
import matplotlib.pyplot as plt
import re
import platform

f = open('age.csv', encoding='euc_kr')
data = csv.reader(f)
result = []
city = ''
for row in data:
    if '산격3' in row[0]:
        str_list = re.split('[()]', row[0]) # [0]: '대구광역시 북구 산격3동(2723063000)'
        city = str_list[0]
        for data in row[3:]: # 0세부터 100세 이상까지 데이터
            result.append(int(data)) # 숫자로 변환
f.close()

if platform.system() == 'Windows':
    plt.rc('font', family='Malgun Gothic')
else:
    plt.rc('font', family='AppleGothic')

plt.title('{0} 인구현황'.format(city))
plt.xlabel('나이')
plt.ylabel('인구수')
plt.style.use('ggplot') # R에서 사용하는 ggplot 패키지 형태
plt.plot(result)
plt.show()
```

re.split('[()]', row[0])
- 괄호 문자('(', ')')를 기준으로 문자열 분리

그래프를 그리기 위해 문자열 자료를 숫자로 변환함



정규식을 사용한 문자열 분리

- 문자열 분리

```
import re
re.split(패턴, 문자열, [최대분할개수])
```

```
import re

city = '대구광역시 북구 산격3동(2723063000)'
str_list = re.split('[()]', city)

print(len(str_list))
print(str_list[0])
for i in range(len(str_list)):
    print('[{}]: {}'.format(i, str_list[i]))
```

```
3
대구광역시 북구 산격3동
[0]: 대구광역시 북구 산격3동
[1]: 2723063000
[2]:
```

```
import re

s = 'apple orange:banana,tomato;melon'

fruits = re.split('[ ,:;]', s)
print(fruits)
```

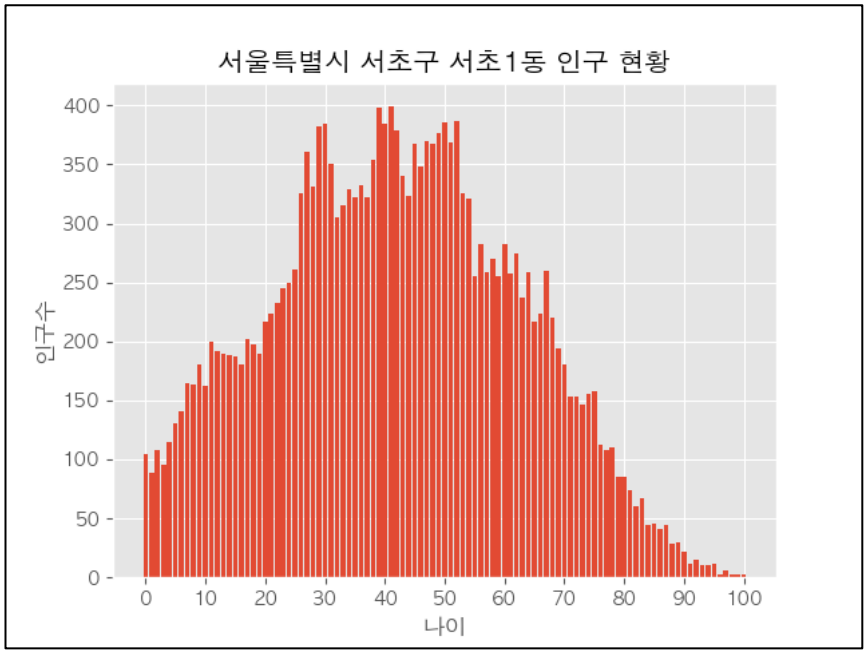
```
['apple', 'orange', 'banana', 'tomato', 'melon']
```

실습: 인구 구조 그래프 함수 구현

- 동 이름을 입력하면 해당 동의 인구 분포를 그리는 함수 구현

```
인구 구조를 알고 싶은 지역의 이름(읍면동 단위)을 입력하세요: 서초1
```

```
  0세:  104명   1세:   89명   2세:  108명   3세:   96명   4세:  115명   5세:  131명   6세:  141명   7세:  165명   8세:  164명   9세:  180명
 10세:  162명  11세:  200명  12세:  192명  13세:  190명  14세:  188명  15세:  187명  16세:  180명  17세:  202명  18세:  197명  19세:  189명
 20세:  217명  21세:  223명  22세:  233명  23세:  245명  24세:  250명  25세:  261명  26세:  326명  27세:  361명  28세:  331명  29세:  382명
 30세:  385명  31세:  351명  32세:  305명  33세:  315명  34세:  329명  35세:  322명  36세:  332명  37세:  322명  38세:  354명  39세:  398명
 40세:  385명  41세:  399명  42세:  379명  43세:  340명  44세:  323명  45세:  368명  46세:  348명  47세:  370명  48세:  367명  49세:  376명
 50세:  386명  51세:  369명  52세:  387명  53세:  325명  54세:  321명  55세:  255명  56세:  282명  57세:  259명  58세:  270명  59세:  255명
 60세:  282명  61세:  258명  62세:  274명  63세:  237명  64세:  259명  65세:  217명  66세:  223명  67세:  260명  68세:  220명  69세:  194명
 70세:  180명  71세:  153명  72세:  153명  73세:  147명  74세:  156명  75세:  158명  76세:  112명  77세:  108명  78세:  110명  79세:   85명
 80세:   85명  81세:   74명  82세:   60명  83세:   67명  84세:   45명  85세:   46명  86세:   41명  87세:   44명  88세:   28명  89세:   30명
 90세:   22명  91세:   11명  92세:   15명  93세:   10명  94세:   10명  95세:   12명  96세:    3명  97세:    6명  98세:    3명  99세:    2명
100세:    2명
```



인구 구조 그래프 함수 구현 #1

```
import csv
import matplotlib.pyplot as plt
import platform
import matplotlib.font_manager as fm
import re

def parse_district_name(district):
    '''
    '행정구역' 명칭에서 숫자 부분을 제거함
    - 서울특별시 종로구 (1111000000)
    '''
    district_name = re.split('[()]', district)
    # [0]: 행정구역 이름, [1]: 코드 번호
    return district_name[0]


def print_population(population):
    '''
    특정 지역의 인구 현황을 화면에 출력함
    '''
    for i in range(len(population)):
        print('{0:3d}세: {1:4d}명'.format(i, population[i]), end=' ')
        if (i + 1) % 10 == 0:
            print()
```

인구 구조 그래프 함수 구현 #2

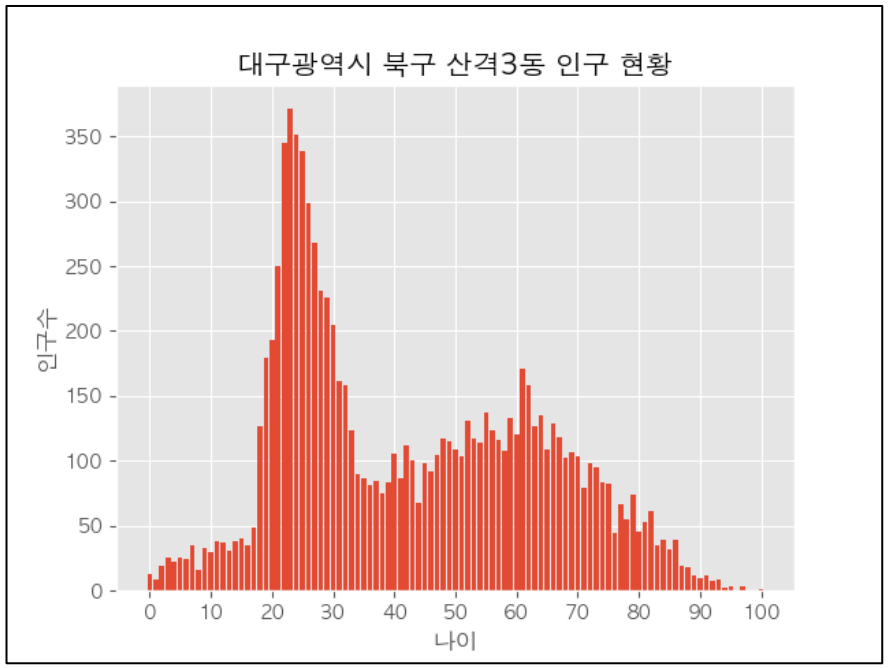
```
def draw_population(district_name, population_list):
    '''
    특정 지역에 대한 인구 분포를 그래프로 나타냄(plot)
    - district_name: 지역 이름
    - population_list: 0~100세 이상까지 인구수 리스트
    '''
    # 그래프 출력
    if platform.system() == 'Windows':
        font_name = fm.FontProperties(fname="c:\Windows\Fonts\malgun.ttf").get_name()
        plt.rc('font', family=font_name)
    else:
        plt.rc('font', family='AppleGothic')

    plt.style.use('ggplot')
    plt.title('{} 인구 현황'.format(district_name))
    plt.xlabel('나이')
    plt.ylabel('인구수')

    plt.bar(range(101), population_list)
    plt.xticks(range(0, 101, 10)) # 0세 ~ 100세 이상

    plt.plot(population_list)
    plt.show()
```

인구 구조 그래프 함수 구현 #3

```
def get_population(district):
    f = open('age.csv', encoding='euc_kr')
    data = csv.reader(f)
    header = next(data) # 헤더 정보 건너뜀

    population_list = []
    full_district_name = ''
    for row in data:
        if district in row[0]:
            full_district_name = parse_district_name(row[0]) # (시 구 동) 이름만 분리
            for data in row[3:]:
                if ',' in data:
                    data = data.replace(',','') # 천단위 콤마 제거
                population_list.append(int(data))

    f.close()
    print_population(population_list)
    draw_population(full_district_name, population_list)


district = input('인구 구조를 알고 싶은 지역의 이름(읍면동 단위)을 입력하세요: ')
get_population(district)
```

막대 그래프: 읍면동 입력 #2 (실행 결과)

```
인구 구조를 알고 싶은 지역의 이름(읍면동 단위)을 입력하세요: 산격3
 0세:   13명  1세:    9명  2세:   19명  3세:   26명  4세:   22명  5세:   26명  6세:   24명  7세:   35명  8세:   16명  9세:   33명
10세:   30명 11세:   38명 12세:   37명 13세:   31명 14세:   38명 15세:   40명 16세:   35명 17세:   49명 18세:  127명 19세:  179명
20세:  193명 21세:  250명 22세:  345명 23세:  371명 24세:  351명 25세:  339명 26세:  298명 27세:  268명 28세:  231명 29세:  226명
30세:  205명 31세:  161명 32세:  158명 33세:  123명 34세:   90명 35세:   87명 36세:   81명 37세:   85명 38세:   75명 39세:   83명
40세:  106명 41세:   87명 42세:  112명 43세:  100명 44세:   68명 45세:   98명 46세:   92명 47세:  105명 48세:  117명 49세:  115명
50세:  109명 51세:  104명 52세:  131명 53세:  117명 54세:  114명 55세:  137명 56세:  124명 57세:  116명 58세:  108명 59세:  133명
60세:  120명 61세:  171명 62세:  158명 63세:  127명 64세:  135명 65세:  109명 66세:  129명 67세:  118명 68세:  102명 69세:  107명
70세:  103명 71세:   79명 72세:   98명 73세:   95명 74세:   84명 75세:   82명 76세:   44명 77세:   67명 78세:   55명 79세:   74명
80세:   46명 81세:   53명 82세:   61명 83세:   35명 84세:   39명 85세:   32명 86세:   39명 87세:   19명 88세:   18명 89세:   12명
90세:   10명 91세:   12명 92세:    8명 93세:    9명 94세:    2명 95세:    3명 96세:    0명 97세:    3명 98세:    0명 99세:    0명
100세:    1명
```



투표 가능 인구수 분석 #1

- 대구 광역시 투표 가능 인구 분석
  - 만 18세 이상 ~
  - csv 파일에서 처음으로 '대구광역시'가 나오는 인구수 활용
    - 전체 대구 인구수

676	대구광역시 (2700000000)	2,375,306	2,375,306	10,208	11,038	12,632	13,901	15,680	17,812	19,828	20,349	20,004	21,238	21,819
677	대구광역시 중구 (2711000000)	78,984	78,984	448	411	450	535	603	633	632	640	580	587	556
678	대구광역시 중구 동인동(27110517	7,825	7,825	24	19	16	24	14	28	24	27	27	36	38
679	대구광역시 중구 삼덕동(27110545	6,473	6,473	26	20	23	38	32	44	37	46	46	44	41
680	대구광역시 중구 성내1동(2711056	4,870	4,870	7	11	13	14	9	11	12	13	20	21	16
681	대구광역시 중구 성내2동(2711057	4,471	4,471	12	4	14	13	12	14	11	12	12	12	15
682	대구광역시 중구 성내3동(2711058	4,677	4,677	45	38	50	49	67	59	66	59	57	57	38
683	대구광역시 중구 대신동(27110595	7,671	7,671	68	70	71	72	87	80	102	81	67	80	68
684	대구광역시 중구 남산1동(2711064	4,592	4,592	20	15	15	21	16	22	34	25	29	27	27
685	대구광역시 중구 남산2동(2711065	6,685	6,685	57	43	45	73	93	81	84	93	67	66	61
686	대구광역시 중구 남산3동(2711066	6,553	6,553	36	36	32	30	63	51	64	55	49	54	64
687	대구광역시 중구 남산4동(2711067	13,979	13,979	88	80	104	123	128	140	115	158	121	116	106

  - 그래프 출력
    - 전체 인구대비 투표 가능 인구 비율: pie chart

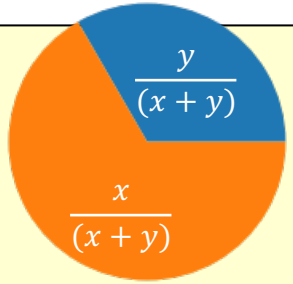
투표 가능 인구수 분석 #2

```
import csv
import matplotlib.pyplot as plt
import platform
import matplotlib.font_manager as fm
import re

def parse_city_name(city):
    '''
    행정구역명에서 도시 이름 파싱 (코드 번호 제거)
    :param city:
    :return:
    '''
    city_name = re.split('[()]', city)
    # [0]: 행정구역 이름, [1]: 코드 번호
    return city_name[0]
```

행정구역에서 도시 이름만 분리
대구광역시 (2700000000)

투표 가능 인구수 분석 #3

```
def draw_piechart(city_name, city_population, voting_population):
    '''
    전체 인구수 대비 투표 가능 인구의 파이차트 작성
    '''
    non_voting_population = city_population - voting_population
    population = [non_voting_population, voting_population]

    if platform.system() == 'Windows':
        plt.rc('font', family='Malgun Gothic')
    else:
        plt.rc('font', family='AppleGothic')

    color = ['tomato', 'royalblue']
    plt.pie(population, labels=['18세 미만', '투표가능인구'], autopct='%.1f%%',
            colors=color, startangle=90)

    plt.legend()
    plt.title(city_name + " 투표 가능 인구 비율")
    plt.show()
```



[18세 미만 인구, 투표가능 인구]
를 pie chart에 전달

투표 가능 인구수 분석 #4

```
def get_voting_population(city):
    '''
     전체 인구수 : row[1], 투표 가능 인구수 분석 row[21:]
    '''
    f = open('age.csv', encoding='euc_kr')
    data = csv.reader(f)
    header = next(data) # 헤더 정보 건너뜀
    city_name = ''
    city_population = 0 # 도시 전체 인구수
    voting_population = 0 # 투표 가능 인구수
    for row in data:
        if city in row[0]:
            city_population = row[1]
            if ',' in city_population:
                # 도시 전체 인구수에서 천단위 콤마 제거
                city_population = city_population.replace(',', '')

            city_population = int(city_population)
            city_name = parse_city_name(row[0]) # (시 구 동) 이름만 분리: 지역 번호 제거
            for data in row[21:]:
                if ',' in data:
                    data = data.replace(',','') # 천단위 콤마 제거
                voting_num = int(data)
                voting_population += voting_num     # 누적된 투표 가능 인구수
            break
    f.close()
    print('{} 전체 인구수:{:,}명, 투표 가능 인구수: {:,}명'.
          format(city_name, city_population, voting_population))
    draw_piechart(city_name, city_population, voting_population)

city = input('투표 가능 인구수를 확인할 도시이름을 입력하시오: ')
get_voting_population(city)
```

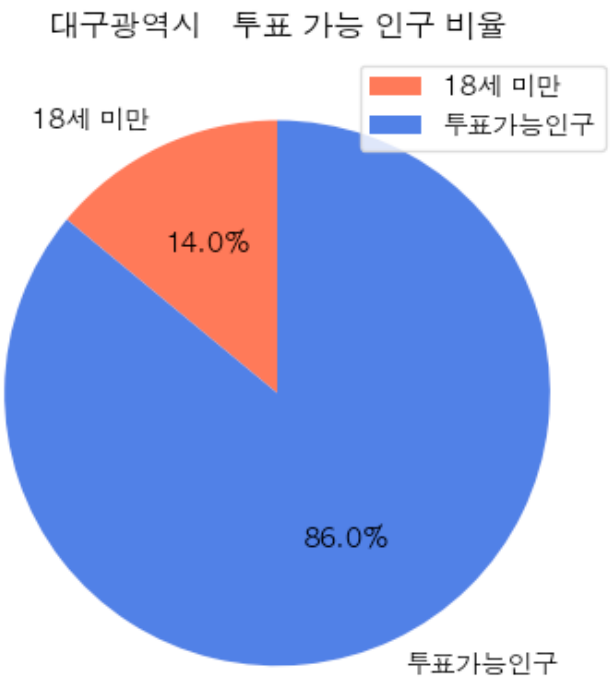
도시 전체 인구수에서 천단위 콤마 제거

특정 도시의 데이터 중에서 제일 먼저 나오는 데이터만 분석하기 위함

18세 이상(투표가능 인구수) 누적

투표 가능 인구수 분석: 실행 결과

```
투표 가능 인구수를 확인할 도시이름을 입력하시오: 대구광역시

대구광역시 전체 인구수:2,375,306명, 투표 가능 인구수: 2,042,207명
```



gender.csv 데이터 분석

- 성별 인구 자료 내부 구조

	연령별 총 인구수						남성 연령별 인구수					여성 연령별 인구수				
열 이름	행정 구역	총인구 수	연령 구간 인구수	계_0세	...	계100 세	남자 총인구 수	남자 연령 구간 총인구 수	남자 0세	...	남자 100세 이상	여자 총인구 수	여자 연령 구간 총인구 수	여자 0세	...	여자 100세 이상
인덱스	[0]	[1]	[2]	[3]	...	[103]	[104]	[105]	[106]		[206]	[207]	[208]	[209]		[309]

- 남성 데이터
  - 총 인구수: 인덱스[104]
  - 연령별 인구수 :[106] ~ [206]까지 저장

- 여성 데이터
  - 총 인구수: 인덱스[207]
  - 연령별 인구수: [209] ~ [309] 까지 저장

gender.csv 헤더 정보

```
import csv

f = open('gender.csv', encoding='euc_kr')
data = csv.reader(f)
header = next(data)

for i in range(len(header)):
    print('[{0:4d}]: {1}'.format(i, header[i]), end=', ')

    if (i+1) % 5 == 0:
        print()

f.close()
```

```
[   0]: 행정구역, [   1]: 2022년06월_계_총인구수, [   2]: 2022년06월_계_연령구간인구수, [   3]: 2022년06월_계_0세,
[   4]: 2022년06월_계_1세, [   5]: 2022년06월_계_2세, [   6]: 2022년06월_계_3세, [   7]: 2022년06월_계_4세,
. . .
[ 100]: 2022년06월_계_97세, [ 101]: 2022년06월_계_98세, [ 102]: 2022년06월_계_99세, [ 103]: 2022년06월_계_100세 이상,

[ 104]: 2022년06월_남_총인구수, [ 105]: 2022년06월_남_연령구간인구수, [ 106]: 2022년06월_남_0세, [ 107]: 2022년06월_남_1세,
[ 108]: 2022년06월_남_2세, [ 109]: 2022년06월_남_3세, [ 110]: 2022년06월_남_4세, [ 111]: 2022년06월_남_5세,
. . .
[ 204]: 2022년06월_남_98세, [ 205]: 2022년06월_남_99세, [ 206]: 2022년06월_남_100세 이상,

[ 207]: 2022년06월_여_총인구수, [ 208]: 2022년06월_여_연령구간인구수, [ 209]: 2022년06월_여_0세, [ 210]: 2022년06월_여_1세,
. . .
[ 306]: 2022년06월_여_97세, [ 307]: 2022년06월_여_98세, [ 308]: 2022년06월_여_99세, [ 309]: 2022년06월_여_100세 이상,
```

1000단위 자리수 제거 및 추가

문자열에서 천 단위 ‘,’ 제거 후 숫자(int)로 변환

```
jeju_male='336,994'
seoul_male = '4,762,711'
# 자릿수 문자를 없앰(replace 사용)
seoul_male = int(seoul_male.replace(',', ''))
print("서울 남자수: ", seoul_male, type(seoul_male))

jeju_male = int(jeju_male.replace(',', ''))
print("제주 남자수: ", jeju_male, type(jeju_male))

print(seoul_male + male)
```

```
서울 남자수: 4762711 <class 'int’>
제주 남자수: 336994 <class 'int’>
5099705
```

크기가 큰 숫자의 경우 1000 단위 자리수 추가

```
seoul_male = 4762711
print("서울 남자수: ", format(seoul_male, ','))
```

```
서울 남자수: 4,762,711
```

연령별 성별 데이터 시각화 #1

- barh(y, data) 그래프: 수평 막대 그래프

```
import csv
import matplotlib.pyplot as plt
import platform

def print_population(population):
    '''
    특정 지역의 인구 현황을 화면에 출력함
    '''
    for i in range(len(population)):
        print('{0:3d}세: {1:4d}명'.format(i, population[i]), end=' ')
        if (i + 1) % 10 == 0:
            print()
    print()
```

```
def draw_geneder_population(male_num_list, female_num_list):
    if platform.system() == 'Windows':
        plt.rc('font', family='Malgun Gothic')
    else:
        plt.rc('font', family='AppleGothic')

    # barh(y축 범위, data)
    plt.barh(range(len(male_num_list)), male_num_list, label='남성')
    plt.barh(range(len(female_num_list)), female_num_list, label='여성')
    plt.rcParams['axes.unicode_minus'] = False
    plt.legend()
    plt.show()
```

연령별 성별 데이터 시각화 #2

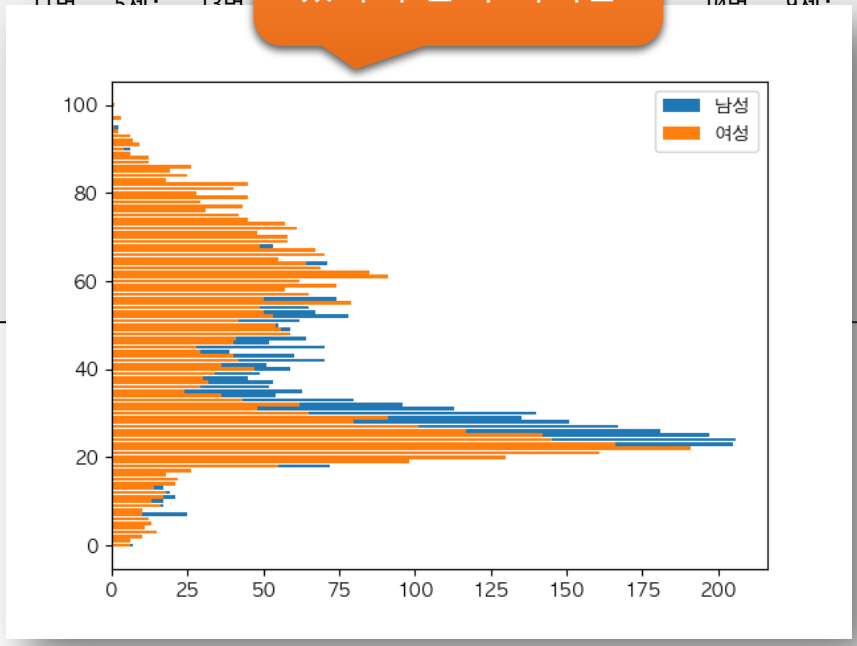
```
def calculate_population():
    f = open('gender.csv', encoding='euc_kr')
    data = csv.reader(f)
    male_num_list = []
    female_num_list = []

    district = input('지역(동) 이름을 입력하세요: ')
    for row in data:
        if district in row[0]:
            for male in row[106:207]:  # 남성 연령별 인구수 구간[106:206]
                if ',' in male:
                    male = male.replace(',', '')  # 천단위 콤마 제거
                male_num_list.append(int(male))

            for female in row[209:310]: # 여성 연령별 인구수 구간[209:309]
                if ',' in female:
                    female = female.replace(',', '')
                female_num_list.append(int(female))

    f.close()
    print('남성 총인구수: ', sum(male_num_list))
    print_population(male_num_list)
    print('------------------------------')
    print('여성 총인구수: ', sum(female_num_list))
    print_population(female_num_list)
    draw_geneder_population(male_num_list, female_num_list)

calculate_population()
```

연령별 성별 데이터 시각화 #3: 실행 결과

지역(동) 이름을 입력하세요: 산격3

```
남성 총인구수:  4977
  0세:    7명   1세:    3명   2세:    9명   3세:   11명   4세:   11명   5세:   13명   6세:   12명   7세:   25명   8세:    6명   9세:   17명
 10세:   17명  11세:   21명  12세:   19명  13세:   17명  14세:   17명  15세:   18명  16세:   17명  17세:   23명  18세:   72명  19세:   81명
 20세:   63명  21세:   89명  22세:  154명  23세:  205명  24세:  206명  25세:  197명  26세:  181명  27세:  167명  28세:  151명  29세:  135명
 30세:  140명  31세:  113명  32세:   96명  33세:   80명  34세:   54명  35세:   63명  36세:   52명  37세:   53명  38세:   45명  39세:   49명
 40세:   59명  41세:   51명  42세:   70명  43세:   60명  44세:   39명  45세:   70명  46세:   52명  47세:   64명  48세:   58명  49세:   59명
 50세:   55명  51세:   62명  52세:   78명  53세:   67명  54세:   65명  55세:   58명  56세:   74명  57세:   51명  58세:   51명  59세:   59명
 60세:   58명  61세:   80명  62세:   73명  63세:   58명  64세:   71명  65세:   54명  66세:   59명  67세:   51명  68세:   53명  69세:   49명
 70세:   45명  71세:   31명  72세:   37명  73세:   38명  74세:   39명  75세:   40명  76세:   13명  77세:   24명  78세:   26명  79세:   29명
 80세:   18명  81세:   13명  82세:   16명  83세:   17명  84세:   14명  85세:   13명  86세:   13명  87세:    7명  88세:    6명  89세:    6명
 90세:    6명  91세:    3명  92세:    1명  93세:    3명  94세:    0명  95세:    2명                                  0명  99세:    0명
100세:    0명
--------------------------------
여성 총인구수:  4522
  0세:    6명   1세:    6명   2세:   10명   3세:   15명   4세:                                                          16명
 10세:   13명  11세:   17명  12세:   18명  13세:   14명  14세:                                                          98명
 20세:  130명  21세:  161명  22세:  191명  23세:  166명  24세:                                                          91명
 30세:   65명  31세:   48명  32세:   62명  33세:   43명  34세:                                                          34명
 40세:   47명  41세:   36명  42세:   42명  43세:   40명  44세:                                                          56명
 50세:   54명  51세:   42명  52세:   53명  53세:   50명  54세:                                                          74명
 60세:   62명  61세:   91명  62세:   85명  63세:   69명  64세:                                                          58명
 70세:   58명  71세:   48명  72세:   61명  73세:   57명  74세:                                                          45명
 80세:   28명  81세:   40명  82세:   45명  83세:   18명  84세:                                                           6명
 90세:    4명  91세:    9명  92세:    7명  93세:    6명  94세:                                                           0명
100세:    1명
```

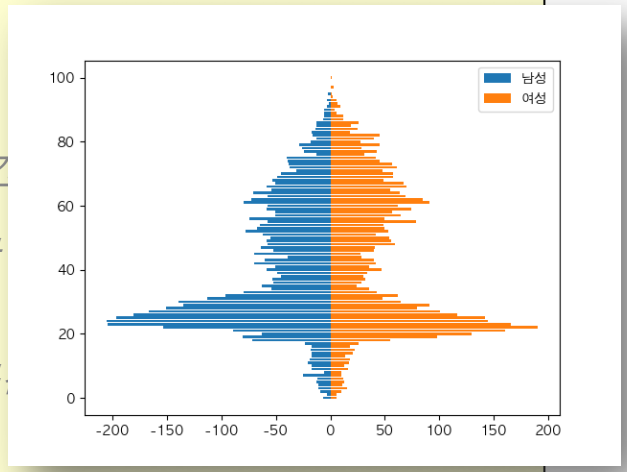
2개의 데이터가 겹쳐 있어 구분이 어려움



연령별 성별 데이터 시각화: 항아리 모양

- 여성 데이터: 오른쪽, 남성 데이터: 왼쪽 표시
  - male_num_list.append(-int(m)): 음수로 변환

```
def calculate_population():
    f = open('gender.csv', encoding='euc_kr')
    data = csv.reader(f)
    male_num_list = []
    female_num_list = []

    district = input('지역(동) 이름을 입력하세요: ')
    for row in data:
        if district in row[0]:
            for male in row[106:207]:  # 남성 연령별 인구수 구간
                if ',' in male:
                    male = male.replace(',', '')  # 천단위 콤마
                male_num_list.append(-int(male))

            for female in row[209:310]: # 여성 연령별 인구수 구
                if ',' in female:
                    female = female.replace(',', '')
                female_num_list.append(int(female))

    . . .
```

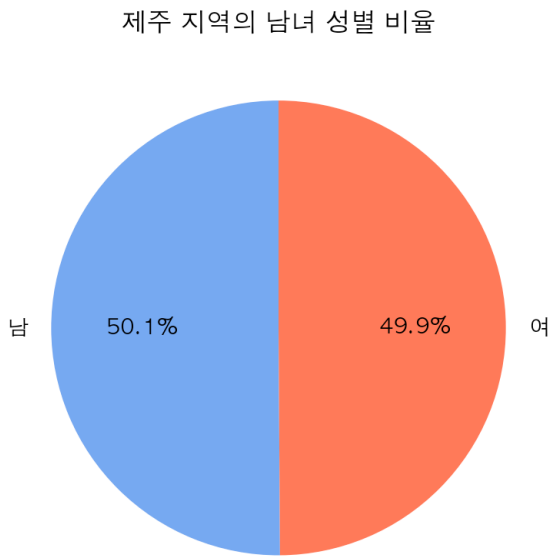


제주도의 성별 인구 비율 표현

▪ 제주특별자치도 인구 현황

행정구역	2022년 6월 남 총인구수	2022년 6월 남 연령구간 인구수	…	2022년 6월 여 총인구수	2022년 6월 여 연령구간 인구수	…
제주특별자치도	**336,697**	336,697		**338,315**	338,315	…
[0]	[104]	[105]	…	[207]	[208]	…

제주 제주 남자 인구수: 336,697명, 여자 인구수: 338,315명



제주도 남녀 인구 비율 예제

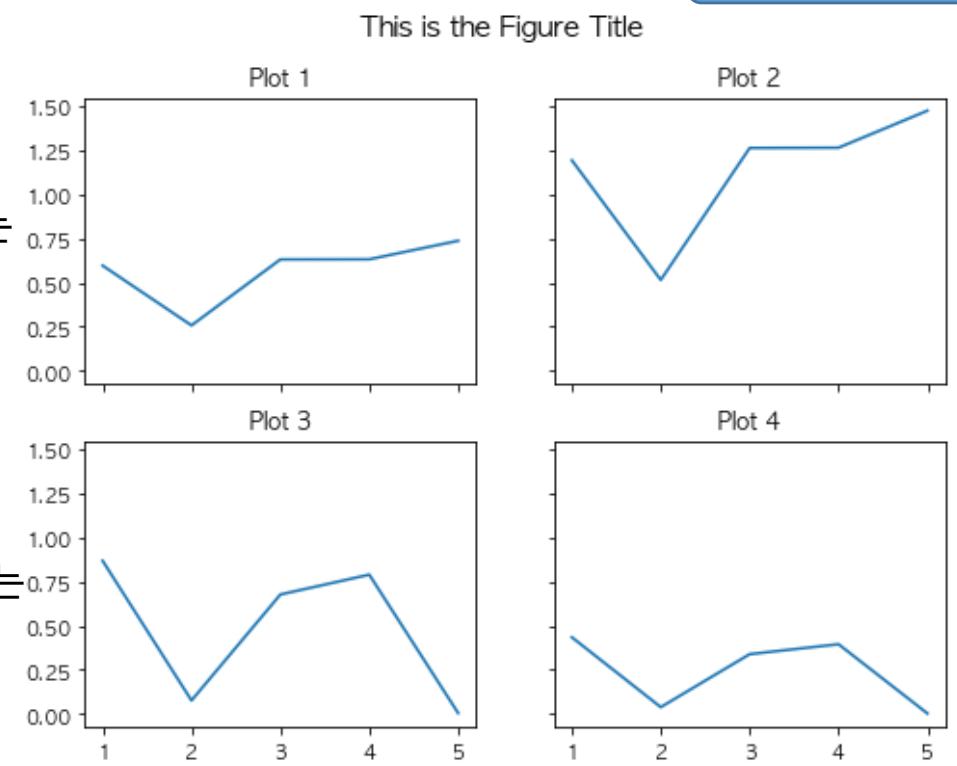
```
import csv
import matplotlib.pyplot as plt
import platform

f = open('gender.csv', encoding='euc_kr')
data = csv.reader(f)
population=[]   # Pie chart에 넣을 데이터 (남, 여 인구수)
city = input('찾고 싶은 지역의 이름을 입력하세요: ')
male_count = 0
female_count = 0

for row in data:
    if city in row[0]:
        male_count = int(row[104].replace(',', ''))  # 자리수 문자열 제거 및 숫자로 변환
        female_count = int(row[207].replace(',', ''))
        break  # 도시별 하위 목록이 많음. 처음에 나오는 데이터가 전체 총합

print("{0} 남자 인구수: {1:}명, 여자 인구수: {2}명".format(city, format(male_count, ','),
                                                    format(female_count, ',')))
population = [male_count, female_count]
if platform.system() == 'Windows':
    plt.rc('font', family='Malgun Gothic')
else:
    plt.rc('font', family='AppleGothic')
color = ['cornflowerblue', 'tomato']
plt.pie(population, labels=['남', '여'], autopct='%.1f%%', colors=color, startangle=90)
plt.title(city + " 지역의 남녀 성별 비율")
plt.show()
```

파이 차트의 시작 각도 설정(90도)

여러 그래프를 한번에 그리기

- subplots(행의 수, 열의 수, figsize=(x, y))
  - 전체 subplot의 개수를 설정함
  - figsize: 각 subplot들의 크기
- subplot(행, 열, index) , axes[행, 열].plot으로 접근
  - index는 1부터 시작함

suptitle('타이틀명')



axes[0, 0].plot 또는 subplot(2,2,1)

axed[0, 1].plot 또는 subplot(2,2,2)

axes[1, 0].plot 또는 subplot(2,2,3)

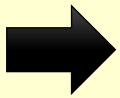
axes[1, 1].plot 또는 subplot(2,2,4)

여러 그래프를 한번에 그리기 예제

```
import matplotlib.pyplot as plt

x = [1,2,3,4,5]
y1 = [0.59705847, 0.25786401, 0.63213726, 0.63287317, 0.73791151]
y2 = [1.19411694, 0.51572803, 1.26427451, 1.26574635, 1.47582302]
y3 = [0.86793828, 0.07563408, 0.67670068, 0.78932712, 0.0043694]
y4 = [0.43396914, 0.03781704, 0.33835034, 0.39466356, 0.0021847]
# 전체 subplot의 개수 설정 (2 x 2= 총 4개)
fig, axes = plt.subplots(2, 2, figsize=(8, 6), sharex=True, sharey=True)
# 전체 그래프의 타이틀 설정
fig.suptitle('This is the Figure Title', fontsize=15)

# Top Left Subplot
axes[0,0].plot(x, y1)
axes[0,0].set_title("Plot 1")

# Top Right Subplot
axes[0,1].plot(x, y2)
axes[0,1].set_title("Plot 2")

# Bottom Left Subplot
axes[1,0].plot(x, y3)
axes[1,0].set_title("Plot 3")

# Bottom Right Subplot
axes[1,1].plot(x, y4)
axes[1,1].set_title("Plot 4")
plt.show()
```

x, y축 tick 공유
sharex=True,
sharey=True

방법 #1

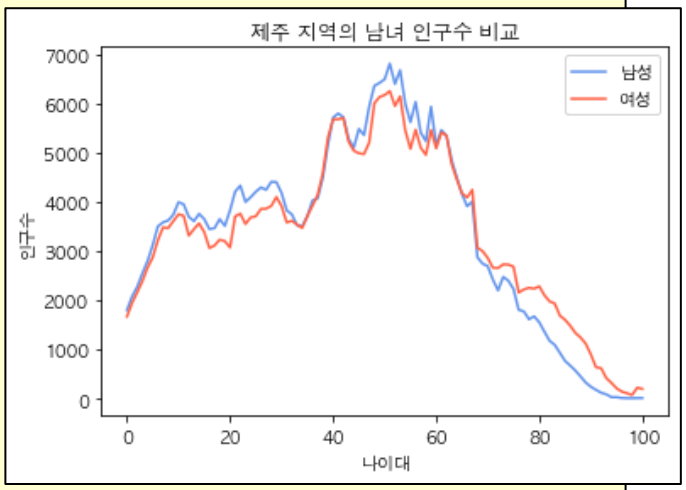


방법 #2

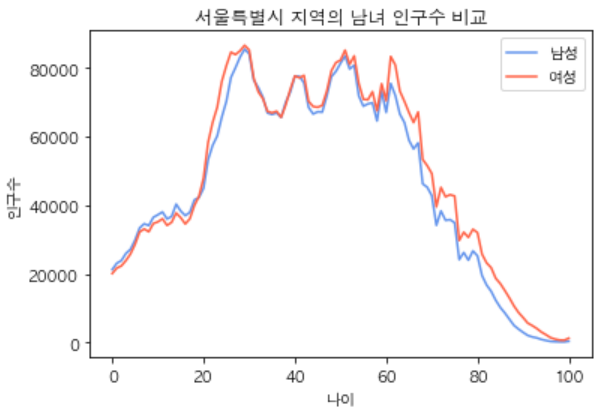
```
# Top Left Subplot
plt.subplot(2, 2, 1)
plt.plot(x, y1)
plt.title('Plot 1')
# Top Right Subplot
plt.subplot(2, 2, 2)
plt.plot(x, y2)
plt.title("Plot 2")
# Bottom Left Subplot
plt.subplot(2, 2, 3)
plt.plot(x, y3)
plt.title("Plot 3")
# Bottom Right Subplot
plt.subplot(2, 2, 4)
plt.plot(x, y4)
plt.title("Plot 4")

plt.show()
```

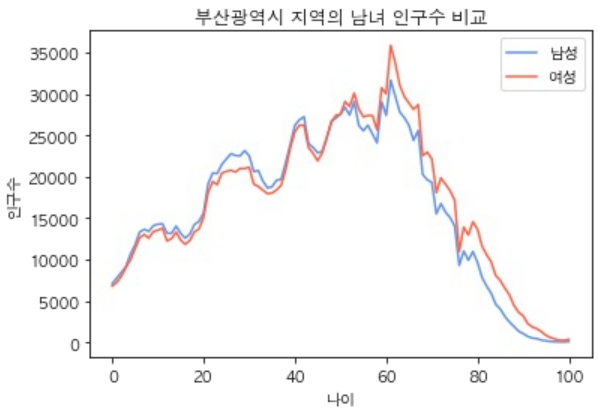
제주도 나이별 인구 현황(꺾은 선 그래프)

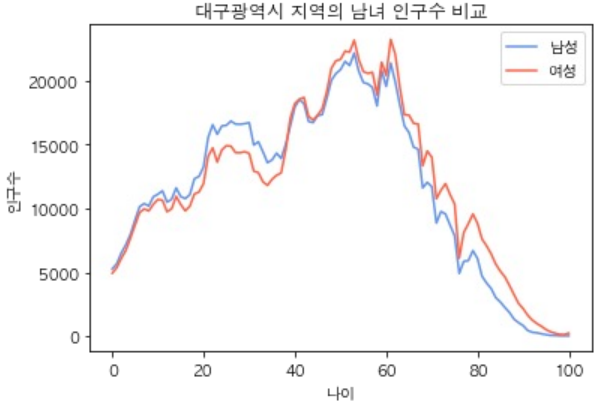
```
import csv
import matplotlib.pyplot as plt
import platform

f = open('gender.csv', encoding='euc_kr')
data = csv.reader(f)
male_list = []
female_list = []
city = input('찾고 싶은 지역의 이름을 입력하세요: ')
for row in data:
    if city in row[0]:
        for i in range(106, 207):
            male_list.append(int(row[i].replace(',','')))
            female_list.append(int(row[i+103].replace(',','')))
        break  # 도시 하위 목록이 많음. 처음에 나오는 데이터가 전체 총합

if platform.system() == 'Windows':
    plt.rc('font', family='Malgun Gothic')
else:
    plt.rc('font', family='AppleGothic')

color = ['cornflowerblue', 'tomato']
plt.plot(male_list, label='남성', color=color[0])
plt.plot(female_list, label='여성', color=color[1])
plt.title(city + " 지역의 남녀 인구수 비교")
plt.xlabel('나이')
plt.ylabel('인구수')
plt.legend()
plt.show()
```

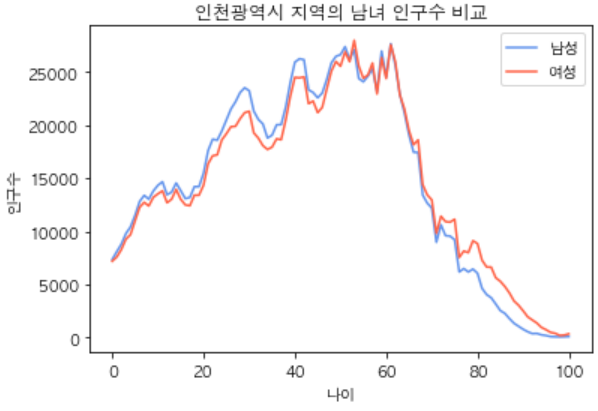


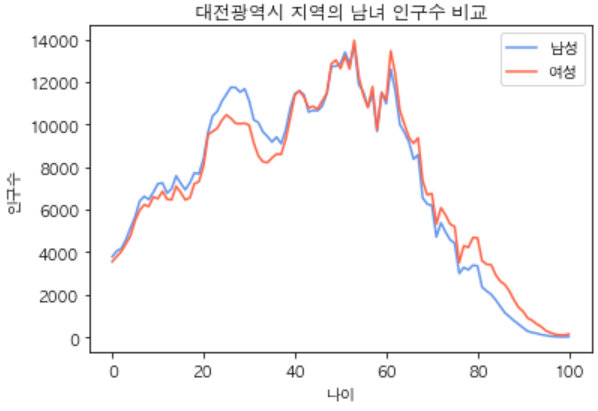
지역별 남녀 인구수 비교

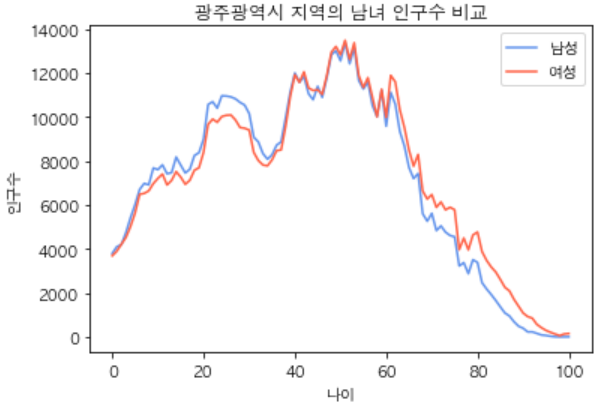

서울특별시 지역의 남녀 인구수 비교
남성
여성
인구수
나이


부산광역시 지역의 남녀 인구수 비교
남성
여성
인구수
나이


대구광역시 지역의 남녀 인구수 비교
남성
여성
인구수
나이

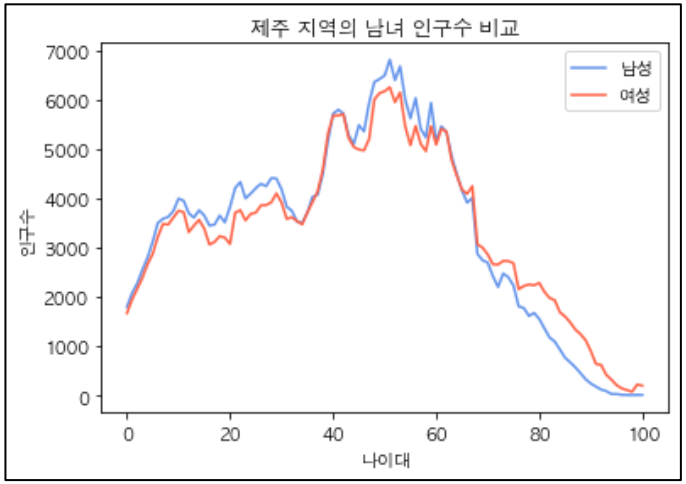

인천광역시 지역의 남녀 인구수 비교
남성
여성
인구수
나이

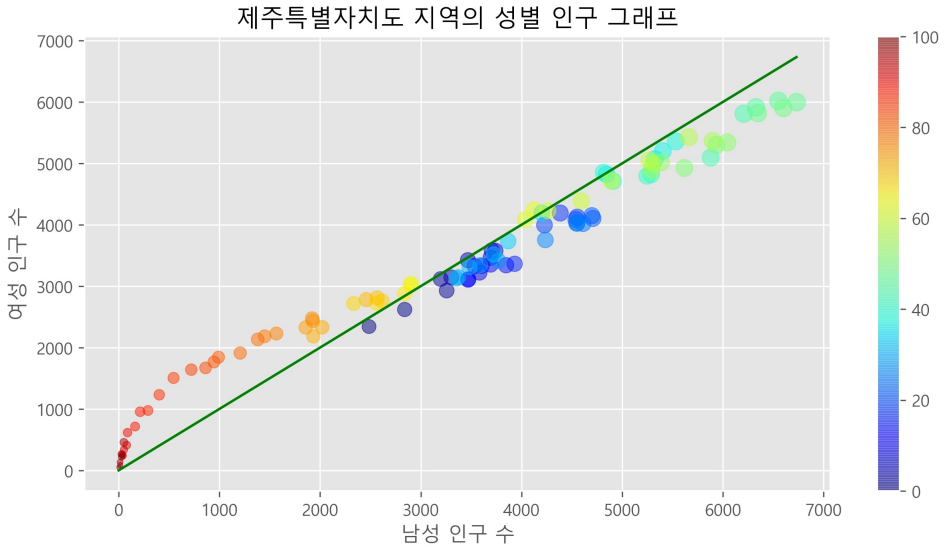

대전광역시 지역의 남녀 인구수 비교
남성
여성
인구수
나이


광주광역시 지역의 남녀 인구수 비교
남성
여성
인구수
나이

산점도(scatter)로 표현하기

- 산점도
  - 가로축과 세로축을 기준으로 두 요소가 서로 어떤 관계를 맺고 있는지를 파악하기 쉽게 나타낸 그래프
  - x축, y축 상관 관계 표시
  - 각 점들은 오른쪽 컬러바를 참고하여 색깔별로 나이를 표시함
    - 버블의 위치: 남녀 성비
    - 버블의 크기: 연령대별 인구수를 표현
  - 제주도 성별 비율
    - 60대 초중반을 기점으로 남녀 성비가 바뀜을 알 수 있음



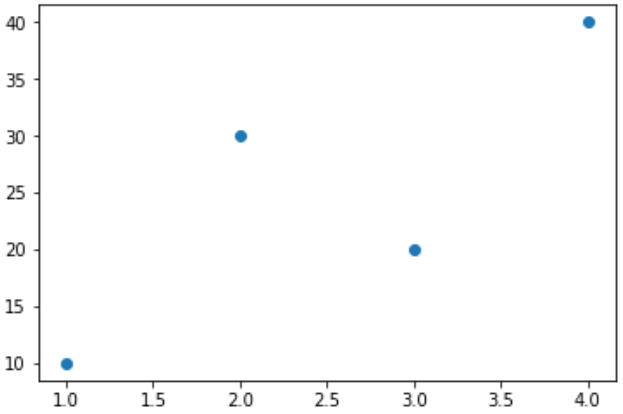


산점도(산포도) 표현

- scatter([x축 데이터], [y축 데이터])함수 사용
  - x축 데이터와 y축 데이터를 넣으면 산점도가 완성됨

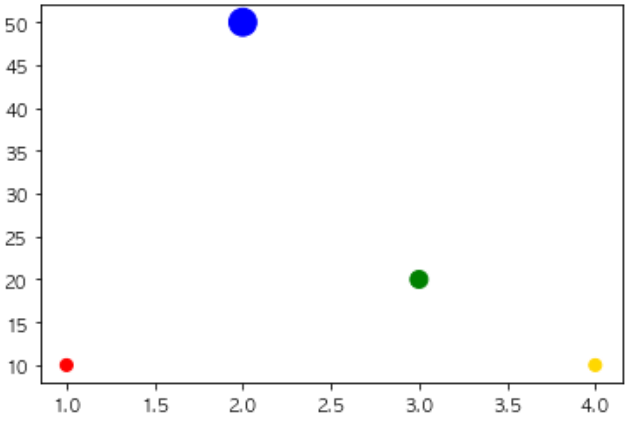
간단한 산점도

```
import matplotlib.pyplot as plt
plt.scatter([1, 2, 3, 4], [10, 30, 20, 40])
plt.show()
```



버블의 크기 표시: s 속성
버블의 색상 변경: c 속성

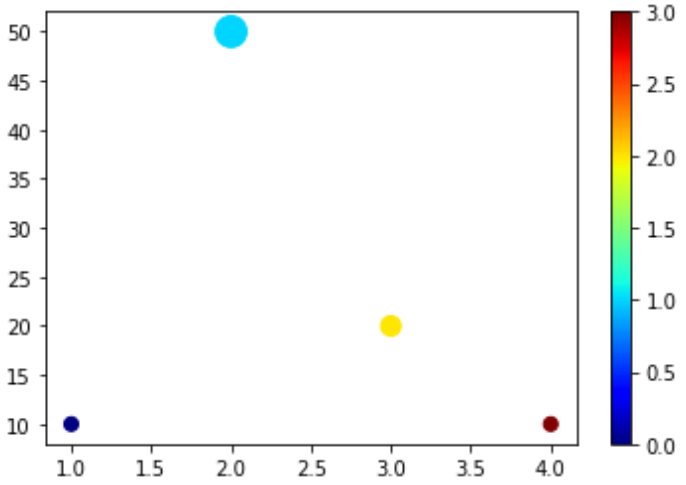
```
import matplotlib.pyplot as plt
y_value = [10, 50, 20, 10]
x_value = [1, 2, 3, 4]
size = []
for value in y_value:
    size.append(value * 5)
# s(size): 버블의 크기
plt.scatter(x_value, y_value, s=size,
            c=['red','blue', 'green', 'gold'])
plt.show()
```



산점도에 color bar 추가

- colorbar() 함수
  - 그래프 우측에 color bar를 추가함
- scatter() 함수 속성 추가
  - c=range(색상 개수)
    - 각 데이터에 해당하는 color bar의 색으로 정해짐
  - cmap: 컬러맵 속성 사용 (cmap='jet') – 무지개색
    - https://matplotlib.org/tutorials/colors/colormaps.html?highlight=colormap

```
import matplotlib.pyplot as plt
y_value = [10, 50, 20, 10]
x_value = [1, 2, 3, 4]
size = []
for y in y_value:
    size.append(y * 5)
plt.scatter(x_value, y_value, s=size, c=range(4),
cmap='jet')
plt.colorbar()
plt.show()
```

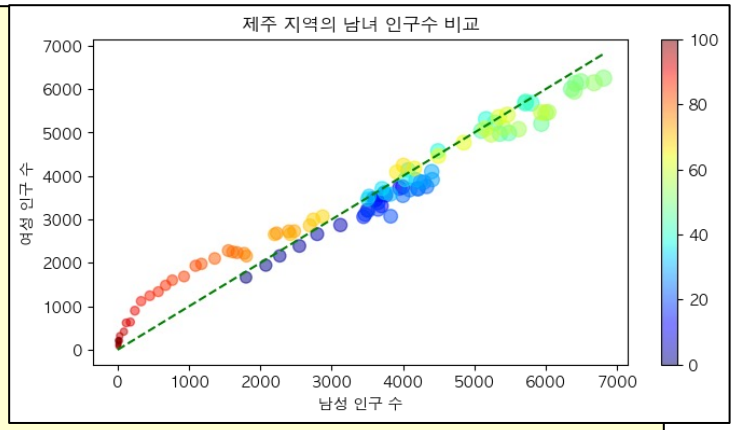


제주도의 연령대별 성별 비율 산점도 #1

```
import csv
import matplotlib.pyplot as plt
import platform
import math

def draw_scatter(city, male_list, female_list, bubble_size_list):
    if platform.system() == 'Windows':
        plt.rc('font', family='Malgun Gothic')
    else:
        plt.rc('font', family='AppleGothic')

    plt.figure(figsize = (8, 4), dpi=100)

    plt.scatter(male_list, female_list, s=bubble_size_list, c=range(101),
                alpha=0.5, cmap='jet')
    plt.colorbar()
    plt.plot(range(max(male_list)), range(max(male_list)), 'g--') # 추세선 추가

    plt.title(city + " 지역의 남녀 인구수 비교")
    plt.xlabel('남성 인구 수')
    plt.ylabel('여성 인구 수')
    plt.show()
```

제주도의 연령대별 성별 비율 산점도 #2

```
def calculate_population():
    f = open('gender.csv', encoding='euc_kr')
    data = csv.reader(f)
    male_list = []
    female_list = []
    bubble_size_list = []
    city = input('찾고 싶은 지역의 이름을 입력하세요: ')

    for row in data:
        if city in row[0]:
            for i in range(106, 207):
                male_num = int(row[i].replace(',',''))
                female_num = int(row[i + 103].replace(',',''))
                # 버블의 사이즈 조절
                bubble_size_list.append(math.sqrt(male_num + female_num))
                #bubble_size_list.append(male_num + female_num)
                male_list.append(male_num)
                female_list.append(female_num)
            break
    f.close()
    draw_scatter(city, male_list, female_list, bubble_size_list)

calculate_population()
```



Questions?