

미니 프로젝트

4교시 :

우리나라 인구 소멸 위기 지역 분석



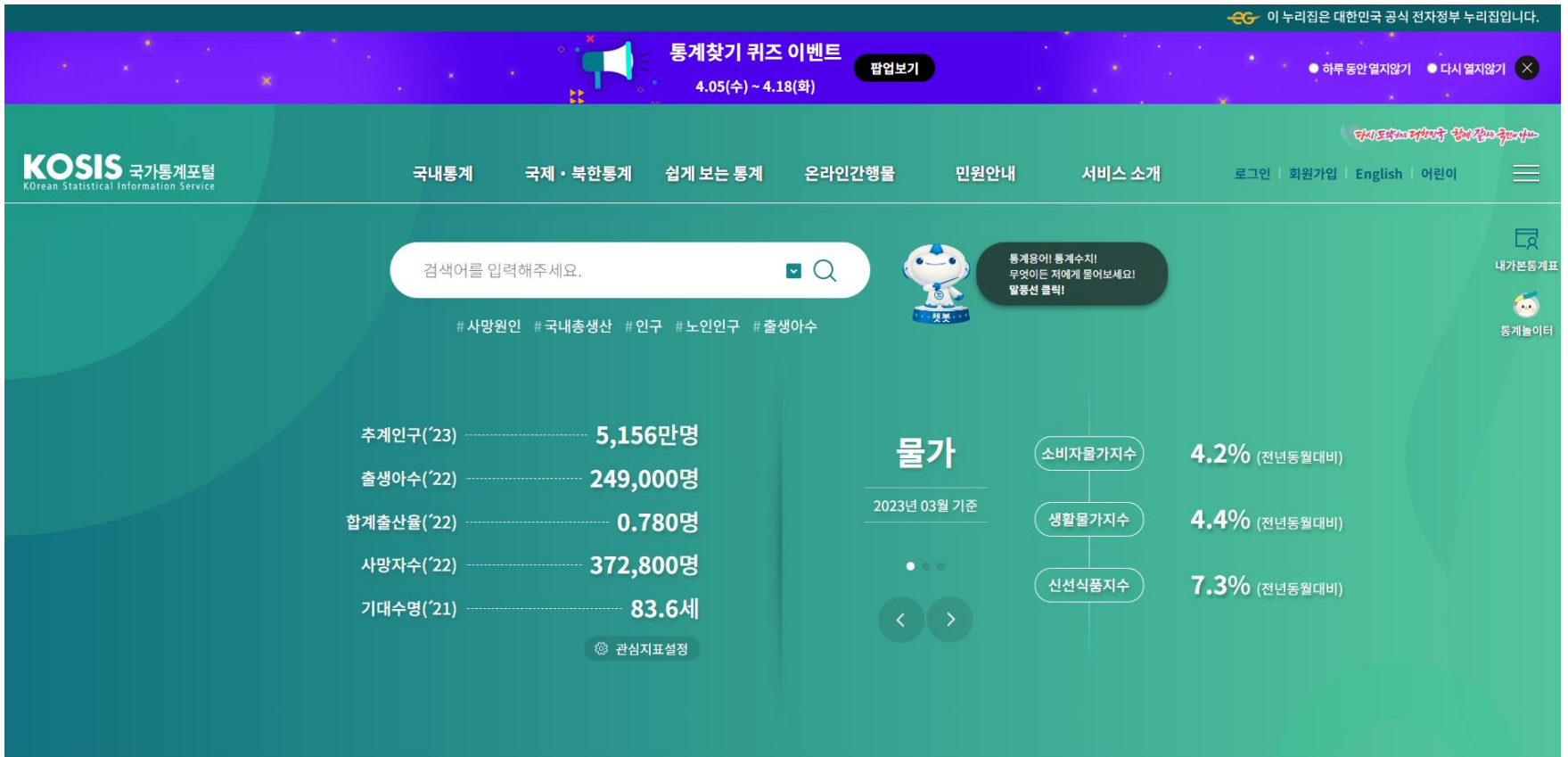
학습목표

- 한국 인구, 머지않아 자연 소멸된다?

1. 데이터 획득 및 준비

데이터 획득

- 국가 통계 포털에서 인구정보를 다운로드
- <https://kosis.kr/index/index.do>





1. 데이터 획득 및 준비

환경 준비

● 한글 처리

파일

소스코드

```
Tf38_cpu  
pip install statsmodels
```

실습환경

```
import matplotlib.pyplot as plt  
import matplotlib  
matplotlib.rcParams['font.family'] = 'Malgun Gothic' # Windows  
# matplotlib.rcParams['font.family'] = 'AppleGothic' # Mac  
matplotlib.rcParams['font.size'] = 15 # 글자 크기  
matplotlib.rcParams['axes.unicode_minus'] = False # 한글 폰트 사용 시, 마이너스 글자가 깨지는 현상을 해결
```

소스코드

결과값1

결과값2

비고



1. 데이터 획득 및 준비

데이터 로드

● 판다스 활용

파일	소스코드
실습환경	Tf38_cpu
소스코드	<pre>import pandas as pd import numpy as np import platform %matplotlib inline path = 'data/population_raw_data.xlsx' data = pd.read_excel(path, header=1) data.head(3)</pre>
결과값1	<pre>행정구역(동읍면)별(1) 행정구역(동읍면)별(2) 항목 계 20 - 24세 25 - 29세 30 - 34세 35 - 39세 65 - 69세 70 - 74세 75 - 79세 80 - 84세 85 - 89세 90 - 94세 95 - 99세 100+ 0 전국 소계 총인구수 (명) 51696216.0 3541061.0 3217367.0 3517868 4016272.0 2237345.0 1781229.0 1457890 909130.0 416164.0 141488.0 34844 17562.0 1 NaN NaN 남자인구수 (명) 25827594.0 1877127.0 1682988.0 1806754 2045265.0 1072395.0 806680.0 600607 319391.0 113221.0 32695.0 7658 4137.0 2 NaN NaN 여자인구수 (명) 25868622.0 1663934.0 1534379.0 1711114 1971007.0 1164950.0 974549.0 857283 589739.0 302943.0 108793.0 27186 13425.0</pre>
비고	

2. 데이터 탐색 및 가공

1. 우리나라 인구 데이터

● 데이터 전처리

파일	소스코드																																																																											
실습환경	Tf38_cpu																																																																											
소스코드	<pre>data.fillna(method='pad', inplace=True) data.rename(columns={'행정구역(동읍면)별(1)': '광역시도', '행정구역(동읍면)별(2)': '시도', '계': '인구수'}, inplace=True) data=data[data['시도']!='소계'] data.head(3)</pre>																																																																											
결과값1	<table><tr><th>광역시도</th><th>시도</th><th>항목</th><th>인구수</th><th>20 - 24세</th><th>25 - 29세</th><th>30 - 34세</th><th>35 - 39세</th><th>65 - 69세</th><th>70 - 74세</th><th>75 - 79세</th><th>80 - 84세</th><th>85 - 89세</th><th>90 - 94세</th><th>95 - 99세</th></tr><tr><td>100+</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>6</td><td>서울특별시</td><td>종로구</td><td>총인구수 (명)</td><td>152737.0</td><td>11379.0</td><td>11891.0</td><td>10684</td><td>10379.0</td><td>7411.0</td><td>6636.0</td><td>5263</td><td>3104.0</td><td>1480.0</td><td>602.0 234 220.0</td></tr><tr><td>7</td><td>서울특별시</td><td>종로구</td><td>남자인구수 (명)</td><td>75201.0</td><td>5620.0</td><td>6181.0</td><td>5387</td><td>5034.0</td><td>3411.0</td><td>3009.0</td><td>2311</td><td>1289.0</td><td>506.0</td><td>207.0 89 73.0</td></tr><tr><td>8</td><td>서울특별시</td><td>종로구</td><td>여자인구수 (명)</td><td>77536.0</td><td>5759.0</td><td>5710.0</td><td>5297</td><td>5345.0</td><td>4000.0</td><td>3627.0</td><td>2952</td><td>1815.0</td><td>974.0</td><td>395.0 145 147.0</td></tr></table>	광역시도	시도	항목	인구수	20 - 24세	25 - 29세	30 - 34세	35 - 39세	65 - 69세	70 - 74세	75 - 79세	80 - 84세	85 - 89세	90 - 94세	95 - 99세	100+															6	서울특별시	종로구	총인구수 (명)	152737.0	11379.0	11891.0	10684	10379.0	7411.0	6636.0	5263	3104.0	1480.0	602.0 234 220.0	7	서울특별시	종로구	남자인구수 (명)	75201.0	5620.0	6181.0	5387	5034.0	3411.0	3009.0	2311	1289.0	506.0	207.0 89 73.0	8	서울특별시	종로구	여자인구수 (명)	77536.0	5759.0	5710.0	5297	5345.0	4000.0	3627.0	2952	1815.0	974.0	395.0 145 147.0
광역시도	시도	항목	인구수	20 - 24세	25 - 29세	30 - 34세	35 - 39세	65 - 69세	70 - 74세	75 - 79세	80 - 84세	85 - 89세	90 - 94세	95 - 99세																																																														
100+																																																																												
6	서울특별시	종로구	총인구수 (명)	152737.0	11379.0	11891.0	10684	10379.0	7411.0	6636.0	5263	3104.0	1480.0	602.0 234 220.0																																																														
7	서울특별시	종로구	남자인구수 (명)	75201.0	5620.0	6181.0	5387	5034.0	3411.0	3009.0	2311	1289.0	506.0	207.0 89 73.0																																																														
8	서울특별시	종로구	여자인구수 (명)	77536.0	5759.0	5710.0	5297	5345.0	4000.0	3627.0	2952	1815.0	974.0	395.0 145 147.0																																																														
비고																																																																												

2. 데이터 탐색 및 가공

1. 우리나라 인구 데이터전처리

● 데이터 전처리

파일	소스코드																																																																											
실습환경	Tf38_cpu																																																																											
소스코드	<pre>data.is_copy = False data.rename(columns={'항목':'구분'}, inplace=True) data.loc[data['구분'] == '총인구수 (명)', '구분'] = '합계' data.loc[data['구분'] == '남자인구수 (명)', '구분'] = '남자' data.loc[data['구분'] == '여자인구수 (명)', '구분'] = '여자' data.head(3)</pre>																																																																											
결과값1	<table><tr><th>광역시도</th><th>시도</th><th>구분</th><th>인구수</th><th>20 - 24세</th><th>25 - 29세</th><th>30 - 34세</th><th>35 - 39세</th><th>65 - 69세</th><th>70 - 74세</th><th>75 - 79세</th><th>80 - 84세</th><th>85 - 89세</th><th>90 - 94세</th><th>95 - 99세</th></tr><tr><td>100+</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>6</td><td>서울특별시</td><td>종로구</td><td>합계</td><td>152737.0</td><td>11379.0</td><td>11891.0</td><td>10684</td><td>10379.0</td><td>7411.0</td><td>6636.0</td><td>5263</td><td>3104.0</td><td>1480.0</td><td>602.0 234 220.0</td></tr><tr><td>7</td><td>서울특별시</td><td>종로구</td><td>남자</td><td>75201.0</td><td>5620.0</td><td>6181.0</td><td>5387</td><td>5034.0</td><td>3411.0</td><td>3009.0</td><td>2311</td><td>1289.0</td><td>506.0</td><td>207.0 89 73.0</td></tr><tr><td>8</td><td>서울특별시</td><td>종로구</td><td>여자</td><td>77536.0</td><td>5759.0</td><td>5710.0</td><td>5297</td><td>5345.0</td><td>4000.0</td><td>3627.0</td><td>2952</td><td>1815.0</td><td>974.0</td><td>395.0 145 147.0</td></tr></table>	광역시도	시도	구분	인구수	20 - 24세	25 - 29세	30 - 34세	35 - 39세	65 - 69세	70 - 74세	75 - 79세	80 - 84세	85 - 89세	90 - 94세	95 - 99세	100+															6	서울특별시	종로구	합계	152737.0	11379.0	11891.0	10684	10379.0	7411.0	6636.0	5263	3104.0	1480.0	602.0 234 220.0	7	서울특별시	종로구	남자	75201.0	5620.0	6181.0	5387	5034.0	3411.0	3009.0	2311	1289.0	506.0	207.0 89 73.0	8	서울특별시	종로구	여자	77536.0	5759.0	5710.0	5297	5345.0	4000.0	3627.0	2952	1815.0	974.0	395.0 145 147.0
광역시도	시도	구분	인구수	20 - 24세	25 - 29세	30 - 34세	35 - 39세	65 - 69세	70 - 74세	75 - 79세	80 - 84세	85 - 89세	90 - 94세	95 - 99세																																																														
100+																																																																												
6	서울특별시	종로구	합계	152737.0	11379.0	11891.0	10684	10379.0	7411.0	6636.0	5263	3104.0	1480.0	602.0 234 220.0																																																														
7	서울특별시	종로구	남자	75201.0	5620.0	6181.0	5387	5034.0	3411.0	3009.0	2311	1289.0	506.0	207.0 89 73.0																																																														
8	서울특별시	종로구	여자	77536.0	5759.0	5710.0	5297	5345.0	4000.0	3627.0	2952	1815.0	974.0	395.0 145 147.0																																																														
비고																																																																												

2. 인구 소멸 위기 지역을 계산해보기

- 인구소멸 위기란?
 - 65세 이상 노인인구와 20~39세 여성 인구를 비교하여 여성인구가 노인 인구의 절반에 미달할 경우
 - 20~30대 인구와 65세 이상인구수 나타내기

파일	소스코드
실습환경	Tf38_cpu
소스코드	<pre>data['20-39세'] = data['20 - 24세'] + data['25 - 29세'] + data['30 - 34세'] + data['35 - 39세'] data['65세이상'] = data['65 - 69세'] + data['70 - 74세'] + data['75 - 79세'] + data['80 - 84세'] \ + data['85 - 89세'] + data['90 - 94세'] + data['95 - 99세'] + data['100+'] data.head(3)</pre>
결과값 ¹	<pre> 광역시도 시도 구분 인구수 20 - 24세 25 - 29세 30 - 34세 35 - 39세 65 - 69세 70 - 74세 75 - 79세 80 - 84세 85 - 89세 90 - 94세 95 - 99세 100+ 20-39세 65세이상 6 서울특별시 종로구 합계 152737.0 11379.0 11891.0 10684 10379.0 7411.0 6636.0 5263 3104.0 1480.0 602.0 234 220.0 44333.0 24950.0 7 서울특별시 종로구 남자 75201.0 5620.0 6181.0 5387 5034.0 3411.0 3009.0 2311 1289.0 506.0 207.0 89 73.0 22222.0 10895.0 8 서울특별시 종로구 여자 77536.0 5759.0 5710.0 5297 5345.0 4000.0 3627.0 2952 1815.0 974.0 395.0 145 147.0 22111.0 14055.0 </pre>
비고	

2. 데이터 탐색 및 가공

2. 인구 소멸 위기 지역을 계산해보기

● 인구소멸 위기란?

파일

소스코드

실습환경

Tf38_cpu

소스코드

```
pop = pd.pivot_table(data, index=['광역시도', '시도'],
                      columns=['구분'],
                      values=['인구수', '20-39세', '65세이상'])
pop.head(3)
```

결과값1

```
20-39세  65세이상  인구수
구분  남자  여자  합계  남자  여자  합계  남자  여자  합계
광역시도  시도
강원도  강릉시  26286.0  23098.0  49384.0  15767.0  21912.0  37679.0  106231.0  107615.0  213846.0
고성군  4494.0  2529.0  7023.0  2900.0  4251.0  7151.0  15899.0  14215.0  30114.0
동해시  11511.0  9753.0  21264.0  6392.0  8732.0  15124.0  47166.0  46131.0  93297.0
```

비고

2. 데이터 탐색 및 가공

2. 인구 소멸 위기 지역을 계산해보기

- 인구 소멸비율: 20-30대 여자 / (65세 이상/2)
- 소멸위기지역: 소멸비율 < 1.0
- multiple columns에서 값 뽑기: df.index.get_level_values(1)을 이용하여, 소멸 위기지역인 곳을 뽑아 낼 수 있다

파일	소스코드
실습환경	Tf38_cpu
소스코드	<pre>pop['소멸비율'] = pop['20-39세', '여자'] / (pop['65세이상', '합계']/2) pop['소멸위기지역'] = pop['소멸비율'] < 1.0 # pop['소멸비율'] < 1.0 하면, 값이 True/False로 나오게 된다. pop[pop['소멸위기지역']==True].index.get_level_values(1)</pre>
결과값1	<pre>Index(['고성군', '삼척시', '양양군', '영월군', '정선군', '평창군', '홍천군', '횡성군', '가평군', '양평군', '연천군', '거창군', '고성군', '남해군', '밀양시', '산청군', '의령군', '창녕군', '하동군', '함안군', '함양군', '합천군', '고령군', '군위군', '문경시', '봉화군', '상주시', '성주군', '영덕군', '영양군', '영주시', '영천시', '예천군', '울릉군', '울진군', '의성군', '청도군', '청송군', '동구', '영도구', '강화군', '옹진군', '강진군', '고흥군', '곡성군', '구례군', '담양군', '보성군', '신안군', '영광군', '영암군', '완도군', '장성군', '장흥군', '진도군', '함평군', '해남군', '화순군', '고창군', '김제시', '남원시', '무주군', '부안군', '순창군', '임실군', '장수군', '정읍시', '진안군', '공주시', '금산군', '논산시', '보령시', '부여군', '서천군', '예산군', '청양군', '태안군', '홍성군', '괴산군', '단양군', '보은군', '영동군', '옥</pre>
비고	

2. 데이터 탐색 및 가공

2. 인구 소멸 위기 지역을 계산해보기

- 인구 소멸비율: 20-30대 여자 / (65세 이상/2)

파일

소스코드

실습환경

Tf38_cpu

소스코드

pop.reset_index(inplace=True)
pop.head(3)

결과값1

광역시도	시도	20-39세	65세이상	인구수	소멸비율	소멸위기지역							
구분	남자	여자	합계	남자	여자	합계	남자	여자	합계				
0	강원도	강릉시	26286.0	23098.0	49384.0	15767.0	21912.0	37679.0	106231.0	107615.0	213846.0	1.226041	False
1	강원도	고성군	4494.0	2529.0	7023.0	2900.0	4251.0	7151.0	15899.0	14215.0	30114.0	0.707314	True
2	강원도	동해시	11511.0	9753.0	21264.0	6392.0	8732.0	15124.0	47166.0	46131.0	93297.0	1.289738	False

비고

2. 인구 소멸 위기 지역을 계산해보기

- 인구 소멸비율: 20-30대 여자 / (65세 이상/2)

파일	소스코드
실습환경	Tf38_cpu
소스코드	<pre>pop.columns = [pop.columns.get_level_values(0)[n] + pop.columns.get_level_values(1)[n] for n in range(0, len(pop.columns.get_level_values(0)))] pop.head(3)</pre>
결과값1	<div> <div>광역시도</div> <div>시도</div> <div>20-39세남자</div> <div>20-39세여자</div> <div>20-39세합계</div> <div>65세이상남자</div> <div>65세이상여자</div> <div>65세이상합계</div> <div>인구수남자</div> <div>인구수여자</div> <div>인구수합계</div> <div>소멸비율</div> </div> <div>소멸위기지역</div> <div> 0 강원도 강릉시 26286.0 23098.0 49384.0 15767.0 21912.0 37679.0 106231.0 107615.0 213846.0 1.226041 False 1 강원도 고성군 4494.0 2529.0 7023.0 2900.0 4251.0 7151.0 15899.0 14215.0 30114.0 0.707314 True 2 강원도 동해시 11511.0 9753.0 21264.0 6392.0 8732.0 15124.0 47166.0 46131.0 93297.0 1.289738 False </div>
비고	

2. 데이터 탐색 및 가공

2. 인구 소멸 위기 지역을 계산해보기

● 데이터 정보 조회

파일

소스코드

실행환경

Tf38_cpu

소스코드

pop.info()

결과값1

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 264 entries, 0 to 263
Data columns (total 13 columns):
#   Column      Non-Null Count  Dtype
---  -
0   광역시도    264 non-null   object
1   시도        264 non-null   object
2   20-39세남자 264 non-null   float64
3   20-39세여자 264 non-null   float64
4   20-39세합계 264 non-null   float64
5   65세이상남자 264 non-null   float64
6   65세이상여자 264 non-null   float64
7   65세이상합계 264 non-null   float64
8   인구수남자  264 non-null   float64
9   인구수여자  264 non-null   float64
10  인구수합계   264 non-null   float64
11  소멸비율     264 non-null   float64
12  소멸위기지역 264 non-null   bool
dtypes: bool(1), float64(10), object(2)
memory usage: 25.1+ KB
```

비고

2. 데이터 탐색 및 가공

2. 인구 소멸 위기 지역을 계산해보기

● 데이터 정보 조회

파일

소스코드

실습환경

Tf38_cpu

소스코드

pop['시도'].unique()

결과값1

```
array(['강릉시', '고성군', '동해시', '삼척시', '속초시', '양구군', '양양군', '영월군', '원주시',  
      '인제군', '정선군', '철원군', '춘천시', '대백시', '평창군', '홍천군', '화천군', '횡성군',  
      '가평군', '고양시', '과천시', '광명시', '광주시', '구리시', '군포시', '권선군', '기흥구',  
      '김포시', '남양주시', '단원구', '덕양구', '동두천시', '동안구', '만안구', '부천시', '분당구',  
      '상록구', '성남시', '소사구', '수원시', '수정구', '수지구', '시흥시', '안산시', '안성시',  
      '안양시', '양주시', '양평군', '여주시', '연천군', '영통구', '오산시', '오정구', '용인시',  
      '원미구', '의왕시', '의정부시', '이천시', '일산동구', '일산서구', '장안군', '충원구', '처인구',  
      '파주시', '팔달구', '평택시', '포천시', '하남시', '화성시', '거제시', '거창군', '김해시',  
      '남해군', '마산합포구', '마산회원구', '밀양시', '사천시', '산청군', '성산군', '양산시', '의령군',  
      '의창구', '진주시', '진해구', '창녕군', '창원시', '통영시', '하동군', '함안군', '함양군',  
      '합천군', '경산시', '경주시', '고령군', '구미시', '군위군', '김천시', '남구', '문경시',  
      '봉화군', '북구', '상주시', '성주군', '안동시', '영덕군', '영양군', '영주시', '영천시',  
      '예천군', '울릉군', '울진군', '의성군', '청도군', '청송군', '철곡군', '포항시', '광산군',  
      '동구', '서구', '달서구', '달성군', '수성구', '중구', '대덕구', '유성구', '강서구', '금정구',  
      '기장군', '동래구', '부산진구', '사상구', '사하구', '수영구', '연제구', '영도구', '해운대구',  
      '강남구', '강동구', '강북구', '관악구', '광진구', '구로구', '금천구', '노원구', '도봉구',  
      '동대문구', '동작구', '마포구', '서대문구', '서초구', '성동구', '성북구', '송파구', '양천구',  
      '영등포구', '용산구', '은평구', '종로구', '중랑구', '세종특별자치시', '울주군', '강화군', '계양구',  
      '남동구', '부평구', '연수구', '웅진군', '강진군', '고흥군', '곡성군', '광양시', '구례군',  
      '나주시', '담양군', '목포시', '무안군', '보성군', '순천시', '신안군', '여수시', '영광군',  
      '영암군', '완도군', '장성군', '장흥군', '진도군', '함평군', '해남군', '화순군', '고창군',  
      '군산시', '김제시', '남원시', '덕진구', '무주군', '부안군', '순창군', '완산구', '완주군',  
      '익산시', '임실군', '장수군', '전주시', '정읍시', '진안군', '서귀포시', '제주시', '계룡시',  
      '공주시', '금산군', '논산시', '당진시', '동남구', '보령시', '부여군', '서북구', '서산시',  
      '서천군', '아산시', '예산군', '천안시', '청양군', '태안군', '홍성군', '괴산군', '단양군',  
      '보은군', '상당구', '서원구', '영동군', '옥천군', '음성군', '제천시', '증평군', '진천군',  
      '청원군', '청주시', '충주시', '충덕구'], dtype=object)
```

비고

2. 데이터 탐색 및 가공

3. 위기 지역을 지도로 보여주기

- 시와 구로 나누어 dictionary에 저장하기
- ~광역시가 아니면서 구를 가지고 있는 시와 행정구를 dict에 저장한다.

파일	소스코드
실습환경	Tf38_cpu
소스코드	<pre>si = [None]*len(pop) tmp_gu_dict = {'수원':['장안구', '권선구', '팔달구', '영통구'], '성남':['수정구', '중원구', '분당구'], '안양':['만안구', '동안구'], '안산':['상록구', '단원구'], '고양':['덕양구', '일산동구', '일산서구'], '용인':['처인구', '기흥구', '수지구'], '청주':['상당구', '서원구', '흥덕구', '청원구'], '천안':['동남구', '서북구'], '전주':['완산구', '덕진구'], '포항':['남구', '북구'], '창원':['의창구', '성산구', '진해구', '마산합포구', '마산회원구'], '부천':['오정구', '원미구', '소사구']}</pre>
결과값1	
비고	

2. 데이터 탐색 및 가공

3. 위기 지역을 지도로 보여주기

- 시와 구로 나누어 dictionary에 저장하기
- ~광역시가 아니면서 구를 가지고 있는 시와 행정구를 dict에 저장한다.

파일

소스코드

실습환경

Tf38_cpu

소스코드

```
for n in pop.index:
    # [-3:] 끝 3글자
    if pop['광역시도'][n][-3:] not in ['광역시', '특별시', '자치시']:
        # [-1] 끝 1글자 빼고 나머지
        if pop['시도'][n][:-1] == '고성' and pop['광역시도'][n] == '강원도':
            si[n] = '고성(강원)'
        elif pop['시도'][n][:-1] == '고성' and pop['광역시도'][n] == '경상남도':
            si[n] = '고성(경남)'
        else:
            try:
                si[n] = pop['시도'][n][:-1]
            except:
                pass

    for keys, values in tmp_gu_dict.items():
        if pop['시도'][n] in values:
            if len(pop['시도'][n]) == 2:
                si[n] = keys + '+' + pop['시도'][n]

            elif pop['시도'][n] in ['마산합포구', '마산회원구']:
                si[n] = keys + '+' + pop['시도'][n][2:-1]
            else:
                si[n] = keys + '+' + pop['시도'][n][:-1]

    elif pop['광역시도'][n] == '세종특별자치시':
        si[n] = '세종'
    else:
        if len(pop['시도'][n]) == 2:
            si[n] = pop['광역시도'][n][2:] + '+' + pop['시도'][n]
        else:
            # '시도'가 3글자 이상이면 보통 뒤에 구, 군, 시 등이 붙으므로 2글자로 맞춰줌
            si[n] = pop['광역시도'][n][2:] + '+' + pop['시도'][n][:-1]
```

si

2. 데이터 탐색 및 가공

3. 위기 지역을 지도로 보여주기

- 시와 구로 나누어 dictionary에 저장하기
- ~광역시가 아니면서 구를 가지고 있는 시와 행정구를 dict에 저장한다.

파일	소스코드
실습환경	Tf38_cpu

소스코드

결과값1

```
['강릉',  
'고성(강원)',  
'동해',  
'삼척',  
'속초',  
'양구',  
'양양',  
'영월',  
'원주',  
'인제',  
'정선',
```

...

비고

2. 데이터 탐색 및 가공

3. 위기 지역을 지도로 보여주기

- 시와 구로 나누어 dictionary에 저장하기
- ~광역시가 아니면서 구를 가지고 있는 시와 행정구를 dict에 저장한다.

파일	소스코드																																																												
실습환경	Tf38_cpu																																																												
소스코드	<pre>pop['ID'] = si pop.head(3)</pre>																																																												
결과값1	<table><tr><th>광역시도</th><th>시도</th><th>20-39세남자</th><th>20-39세여자</th><th>20-39세합계</th><th>65세이상남자</th><th>65세이상여자</th><th>65세이상합계</th><th>인구수남자</th><th>인구수여자</th><th>인구수합계</th><th>소멸비율</th></tr><tr><td>소멸위기지역</td><td>ID</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>0</td><td>강원도 강릉시</td><td>26286.0</td><td>23098.0</td><td>49384.0</td><td>15767.0</td><td>21912.0</td><td>37679.0</td><td>106231.0</td><td>107615.0</td><td>213846.0</td><td>1.226041</td></tr><tr><td>1</td><td>강원도 고성군</td><td>4494.0</td><td>2529.0</td><td>7023.0</td><td>2900.0</td><td>4251.0</td><td>7151.0</td><td>15899.0</td><td>14215.0</td><td>30114.0</td><td>0.707314</td></tr><tr><td>2</td><td>강원도 동해시</td><td>11511.0</td><td>9753.0</td><td>21264.0</td><td>6392.0</td><td>8732.0</td><td>15124.0</td><td>47166.0</td><td>46131.0</td><td>93297.0</td><td>1.289738</td></tr></table>	광역시도	시도	20-39세남자	20-39세여자	20-39세합계	65세이상남자	65세이상여자	65세이상합계	인구수남자	인구수여자	인구수합계	소멸비율	소멸위기지역	ID											0	강원도 강릉시	26286.0	23098.0	49384.0	15767.0	21912.0	37679.0	106231.0	107615.0	213846.0	1.226041	1	강원도 고성군	4494.0	2529.0	7023.0	2900.0	4251.0	7151.0	15899.0	14215.0	30114.0	0.707314	2	강원도 동해시	11511.0	9753.0	21264.0	6392.0	8732.0	15124.0	47166.0	46131.0	93297.0	1.289738
광역시도	시도	20-39세남자	20-39세여자	20-39세합계	65세이상남자	65세이상여자	65세이상합계	인구수남자	인구수여자	인구수합계	소멸비율																																																		
소멸위기지역	ID																																																												
0	강원도 강릉시	26286.0	23098.0	49384.0	15767.0	21912.0	37679.0	106231.0	107615.0	213846.0	1.226041																																																		
1	강원도 고성군	4494.0	2529.0	7023.0	2900.0	4251.0	7151.0	15899.0	14215.0	30114.0	0.707314																																																		
2	강원도 동해시	11511.0	9753.0	21264.0	6392.0	8732.0	15124.0	47166.0	46131.0	93297.0	1.289738																																																		
비고																																																													

2. 데이터 탐색 및 가공

3. 위기 지역을 지도로 보여주기

- 필요없는 데이터 columns들을 지워주자
- 저장

파일	소스코드																																												
실습환경	Tf38_cpu																																												
소스코드	<pre>del pop['20-39세남자'] del pop['65세이상남자'] del pop['65세이상여자'] pop.head(3) pop.to_excel('data/draw_korea_raw.xlsx', index=False)</pre>																																												
결과값1	<table><tr><th>광역시도</th><th>시도</th><th>20-39세여자</th><th>20-39세합계</th><th>65세이상합계</th><th>인구수남자</th><th>인구수여자</th><th>인구수합계</th><th>소멸비율</th><th>소멸위기지역</th><th>ID</th></tr><tr><td>0</td><td>강원도 강릉시</td><td>23098.0</td><td>49384.0</td><td>37679.0</td><td>106231.0</td><td>107615.0</td><td>213846.0</td><td>1.226041</td><td>False</td><td>강릉</td></tr><tr><td>1</td><td>강원도 고성군</td><td>2529.0</td><td>7023.0</td><td>7151.0</td><td>15899.0</td><td>14215.0</td><td>30114.0</td><td>0.707314</td><td>True</td><td>고성(강원)</td></tr><tr><td>2</td><td>강원도 동해시</td><td>9753.0</td><td>21264.0</td><td>15124.0</td><td>47166.0</td><td>46131.0</td><td>93297.0</td><td>1.289738</td><td>False</td><td>동해</td></tr></table>	광역시도	시도	20-39세여자	20-39세합계	65세이상합계	인구수남자	인구수여자	인구수합계	소멸비율	소멸위기지역	ID	0	강원도 강릉시	23098.0	49384.0	37679.0	106231.0	107615.0	213846.0	1.226041	False	강릉	1	강원도 고성군	2529.0	7023.0	7151.0	15899.0	14215.0	30114.0	0.707314	True	고성(강원)	2	강원도 동해시	9753.0	21264.0	15124.0	47166.0	46131.0	93297.0	1.289738	False	동해
광역시도	시도	20-39세여자	20-39세합계	65세이상합계	인구수남자	인구수여자	인구수합계	소멸비율	소멸위기지역	ID																																			
0	강원도 강릉시	23098.0	49384.0	37679.0	106231.0	107615.0	213846.0	1.226041	False	강릉																																			
1	강원도 고성군	2529.0	7023.0	7151.0	15899.0	14215.0	30114.0	0.707314	True	고성(강원)																																			
2	강원도 동해시	9753.0	21264.0	15124.0	47166.0	46131.0	93297.0	1.289738	False	동해																																			
비고																																													



3. 위기 지역을 지도로 보여주기

- ID로 나눈, excel파일 모습

[illegible]

2. 데이터 탐색 및 가공

3. 위기 지역을 지도로 보여주기

● 지도 보여주기

파일	소스코드
실습환경	Tf38_cpu
소스코드	<pre>draw_korea_raw = pd.read_excel('./data/draw_korea_raw.xlsx') draw_korea_raw.head(3)</pre>
결과값1	<pre>0 1 2 3 4 5 6 7 8 9 10 11 12 13 0 NaN NaN NaN NaN NaN NaN NaN NaN 철원 화천 양구 고성(강원) NaN NaN NaN 1 NaN NaN NaN 양주 동두천 연천 포천 의정부 인제 춘천 속초 NaN NaN NaN 2 NaN NaN NaN 고양 덕양 고양 일산동 서울 도봉 서울 노원 남양주 홍천 횡성 양양 NaN NaN NaN</pre>
비고	



2. 데이터 탐색 및 가공

3. 위기 지역을 지도로 보여주기

● 지도 보여주기

파일	소스코드
실습환경	Tf38_cpu
소스코드	<pre>draw_korea_raw_starcked = pd.DataFrame(draw_korea_raw.stack()) draw_korea_raw_starcked.reset_index(inplace=True) draw_korea_raw_starcked.rename(columns={'level_0':'y', 'level_1':'x', 'o':'ID'}, inplace=True) draw_korea_raw_starcked.head() draw_korea = draw_korea_raw_starcked</pre>
결과값1	<pre>y x ID 0 0 7 철원 1 0 8 화천 2 0 9 양구 3 0 10 고성(강원) 4 1 3 양주</pre>
비고	

2. 데이터 탐색 및 가공

3. 위기 지역을 지도로 보여주기

● 지도 보여주기

파일

소스코드

실습환경

Tf38_cpu

소스코드

```
BORDER_LINES = [[(5, 1), (5, 2), (7, 2), (7, 3), (11,3), (11, 0)],
[(5, 4), (5, 5), (2, 5), (2, 7), (4, 7), (4, 9), (7, 9), (7, 7), (9, 7), (9, 5), (10, 5), (10, 4), (5, 4)],
[(1, 7), (1, 8), (3, 8), (3, 10), (10, 10), (10, 7), (12, 7), (12, 6), (11, 6), (11, 5), (12, 5), (12, 4), (11, 4), (11, 3)],
[(8, 10), (8, 11), (6, 11), (6, 12)],
[(12, 5), (13, 5), (13, 4), (14, 4), (14, 5), (15, 5), (15, 4), (16, 4), (16, 2)],
[(16, 4), (17, 4), (17, 5), (16, 5), (16, 6), (19, 6), (19, 5), (20, 5), (20, 4), (21, 4), (21, 3), (19, 3), (19, 1)],
[(13, 5), (13, 6), (16, 6)],
[(13, 5), (14, 5)],
[(21, 2), (21, 3), (22, 3), (22, 4), (24, 4), (24, 2), (21, 2)],
[(20, 5), (21, 5), (21, 6), (23, 6)],
[(10, 8), (12, 8), (12, 9), (14, 9), (14, 8), (16, 8), (16, 6)],
[(14, 9), (14, 11), (14, 12), (13, 12), (13, 13)],
[(15, 8), (17, 8), (17, 10), (16, 10), (16, 11), (14, 11)],
[(17, 9), (18, 9), (18, 8), (19, 8), (19, 9), (20, 9), (20, 10), (21, 10)],
[(16, 11), (16, 13)],
[(27, 5), (27, 6), (25, 6)],
]
```

결과값1

비고



2. 데이터 탐색 및 가공

3. 위기 지역을 지도로 보여주기

● 지도 보여주기

파일	소스코드
실습환경	Tf38_cpu
소스코드	<pre>plt.figure(figsize=(8, 11)) for idx, row in draw_korea.iterrows(): if len(row['ID'].split())==2: dispname = '{}\n{}'.format(row['ID'].split()[0], row['ID'].split()[1]) elif row['ID'][:2] == '고성': dispname = '고성' else: dispname = row['ID'] if len(dispname.splitlines()[-1]) >= 3: fontsize, linespacing = 9.5, 1.5 else: fontsize, linespacing = 11, 1.2 plt.annotate(dispname, (row['x']+0.5, row['y']+0.5), weight='bold', fontsize= fontsize, ha='center', va='center', linespacing=linespacing) for path in BORDER_LINES: ys, xs = zip(*path) plt.plot(xs, ys, c='black', lw=1.5) plt.gca().invert_yaxis() plt.axis('off') plt.tight_layout() plt.show()</pre>

결과값1

비고

2. 데이터 탐색 및 가공

3. 위기 지역을 지도로 보여주기

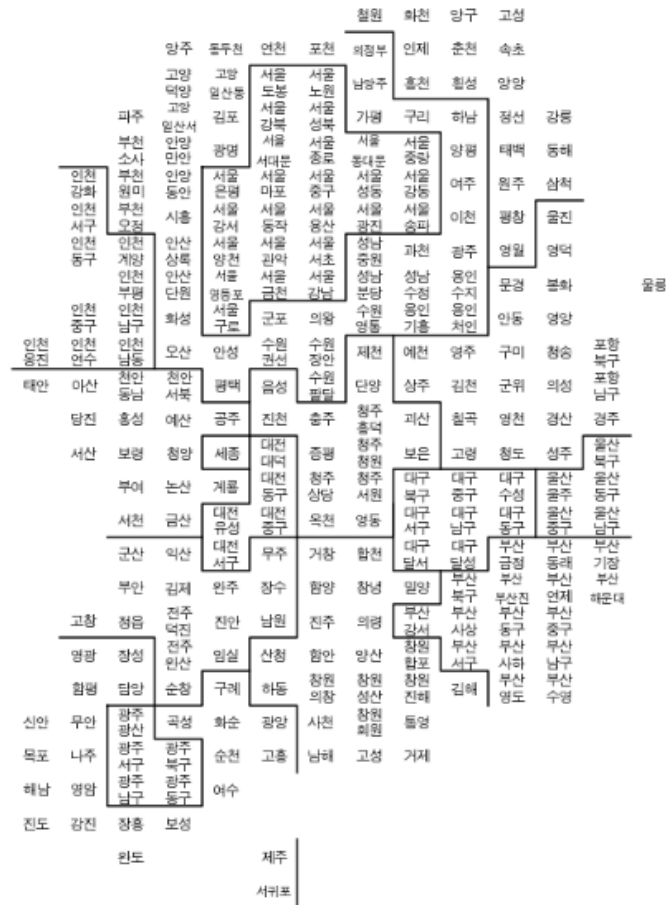
● 지도 보여주기

파일 소스코드
실습환경 Tf38_cpu

소스코드

결과값1

비고





2. 데이터 탐색 및 가공

3. 위기 지역을 지도로 보여주기

- draw_korea와 pop을 합쳐보자
- pop['ID']에서 우리가 만든 draw_korea['ID']에 없는 내용은 뺀다.

파일	소스코드
실습환경	Tf38_cpu
소스코드	<pre>tmp_list = list(set(pop['ID'].unique()) - set(draw_korea['ID'].unique())) for tmp in tmp_list: pop = pop.drop(pop[pop['ID']==tmp].index) tmp_list</pre>
결과값1	성남, '포항', '수원', '청주', '안산', '용인', '부천', '창원', '천안', '안양', '고양', '전주']
비고	

2. 데이터 탐색 및 가공

3. 위기 지역을 지도로 보여주기

- draw_korea와 pop을 합쳐보자
- pop에 있는 데이터들 + draw_korea에 있던 좌표까지 입력

파일	소스코드
실습환경	Tf38_cpu
소스코드	<pre>pop = pd.merge(pop, draw_korea, how='left', on=['ID']) pop.head()</pre>

결과값1	광역시도 시도 20-39세여자 20-39세합계 65세이상합계 인구수남자 인구수여자 인구수합계 소멸비율 소멸위기지역 ID y x													
	광역시도	시도	20-39세여자	20-39세합계	65세이상합계	인구수남자	인구수여자	인구수합계	소멸비율	소멸위기지역	ID	y	x	
0	강원도	강릉시	23098.0	49384.0	37679.0	106231.0	107615.0	213846.0	1.226041	False	강릉	3	11	
1	강원도	고성군	2529.0	7023.0	7151.0	15899.0	14215.0	30114.0	0.707314	True	고성(강원)	0	10	
2	강원도	동해시	9753.0	21264.0	15124.0	47166.0	46131.0	93297.0	1.289738	False	동해	4	11	
3	강원도	삼척시	7115.0	15823.0	14610.0	35253.0	34346.0	69599.0	0.973990	True	삼척	5	11	
4	강원도	속초시	8752.0	18708.0	12752.0	40288.0	41505.0	81793.0	1.372647	False	속초	1	10	

비고

2. 데이터 탐색 및 가공

3. 위기 지역을 지도로 보여주기

● 정보 그리기 함수

파일

소스코드

실습환경

Tf38_cpu

소스코드

```
def drawKorea(targetData, blockedMap, cmapname):
    gamma = .75
    whitelabelmin = (max(blockedMap[targetData]) - min(blockedMap[targetData]))*0.25 + min(blockedMap[targetData])
    datalabel = targetData

    vmin = min(blockedMap[targetData])
    vmax = max(blockedMap[targetData])

    mapdata = blockedMap.pivot_table(index='y', columns='x', values=targetData)
    masked_mapdata = np.ma.masked_where(np.isnan(mapdata), mapdata)

    plt.figure(figsize=(6, 8))
    plt.pcolor(masked_mapdata, vmin=vmin, vmax=vmax, cmap=cmapname, edgecolor='#aaaaaa', linewidth=0.5)

    for idx, row in blockedMap.iterrows():
        if len(row['ID'].split())==2:
            dispname = '{}\n{}'.format(row['ID'].split()[0], row['ID'].split()[1])
        elif row['ID'][:2] != '고성':
            dispname = '고성'
        else:
            dispname = row['ID']

        if len(dispname.splitlines()[-1]) >= 3:
            fontsize, linespacing = 8, 1.1
        else:
            fontsize, linespacing = 9, 0.9

        annocolor = 'white' if row[targetData] > whitelabelmin else 'black'
        plt.annotate(dispname, (row['x']+0.5, row['y']+0.5), weight='bold', fontsize=fontsize, ha='center', va='center', linespacing=linespacing)

    for path in BORDER_LINES:
        ys, xs = zip(*path)
        plt.plot(xs, ys, c='black', lw=2)

    plt.gca().invert_yaxis()
    plt.axis('off')
    cb=plt.colorbar(shrink=.1, aspect=10)
    cb.set_label(datalabel)

    plt.tight_layout()
    plt.show()
```

결과값1

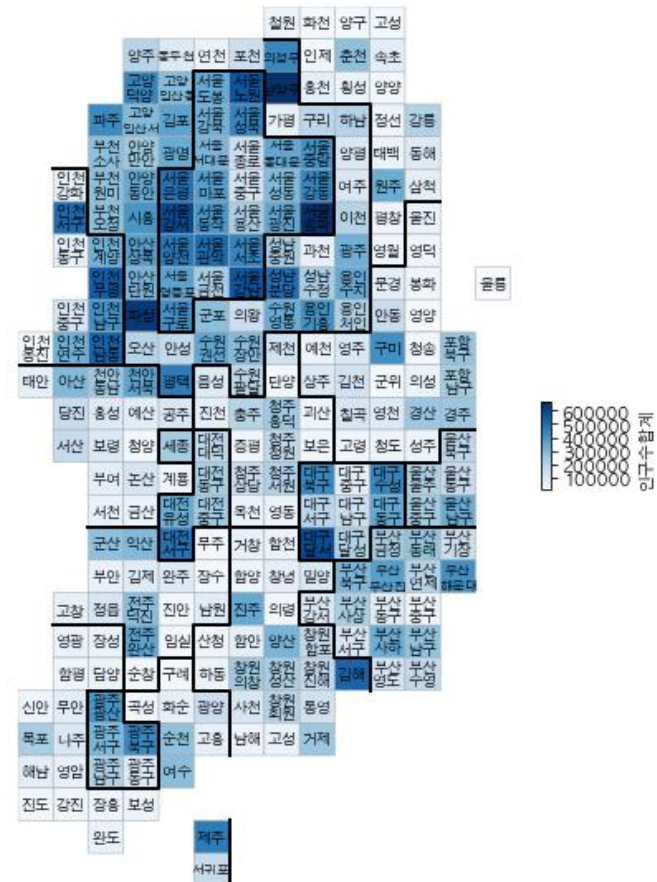
비고

2. 데이터 탐색 및 가공

3. 위기 지역을 지도로 보여주기

● 정보 그리기 함수

파일	소스코드
실습환경	Tf38_cpu
소스코드	<code>drawKorea('인구수합계', pop,</code>
결과값1	
비고	

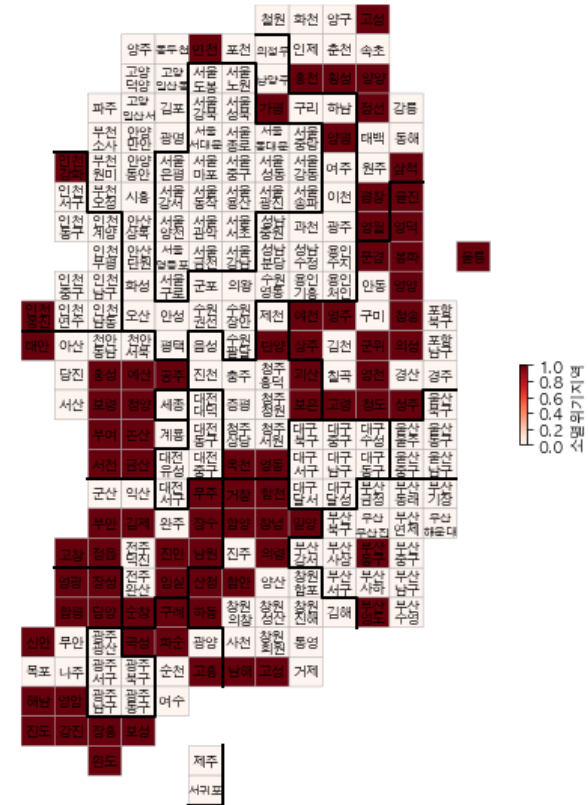


2. 데이터 탐색 및 가공

4. 소멸위기지역

- True, False를 1, 0으로 나타내기

파일	소스코드
실습환경	Tf38_cpu
소스코드	<pre>pop['소멸위기지역'] = [1 if con else 0 for con in pop['소멸위기지역']] drawKorea('소멸위기지역', pop, 'Reds')</pre>
결과값1	
비고	



2. 데이터 탐색 및 가공

5. 여성비

파일

소스코드

실습환경

Tf38_cpu

```
def drawKorea2(targetData, blockedMap, cmapname):
    gamma = .75
    whitelabelmin = 20.
    datalabel = targetData

    #음수가 존재할 수 있기 때문에 절대값으로 변경
    tmp_max = max([np.abs(min(blockedMap[targetData])), np.abs(max(blockedMap[targetData]))])
    vmin, vmax = -tmp_max, tmp_max

    mapdata = blockedMap.pivot_table(index='y', columns='x', values=targetData)
    masked_mapdata = np.ma.masked_where(np.isnan(mapdata), mapdata)

    plt.figure(figsize=(6, 8))
    plt.pcolor(masked_mapdata, vmin=vmin, vmax=vmax, cmap=cmapname, edgecolor='#aaaaaa', linewidth=0.5)

    for idx, row in blockedMap.iterrows():
        if len(row['ID'].split())==2:
            dispname = '{}\n{}'.format(row['ID'].split()[0], row['ID'].split()[1])
        elif row['ID'][:2] == '고성':
            dispname = '고성'
        else:
            dispname = row['ID']

        if len(dispname.splitlines()[-1]) >= 3:
            fontsize, linespacing = 8, 1.1
        else:
            fontsize, linespacing = 9, 0.9

        annocolor = 'white' if row[targetData] > whitelabelmin else 'black'
        plt.annotate(dispname, (row['x']+0.5, row['y']+0.5), weight='bold', fontsize=fontsize, ha='center', va='center', color=annocolor, linespacing=linespacing)

    for path in BORDER_LINES:
        ys, xs = zip(*path)
        plt.plot(xs, ys, c='black', lw=2)

    plt.gca().invert_yaxis()
    plt.axis('off')
    cb = plt.colorbar(shrink=.1, aspect=10)
    cb.set_label(datalabel)

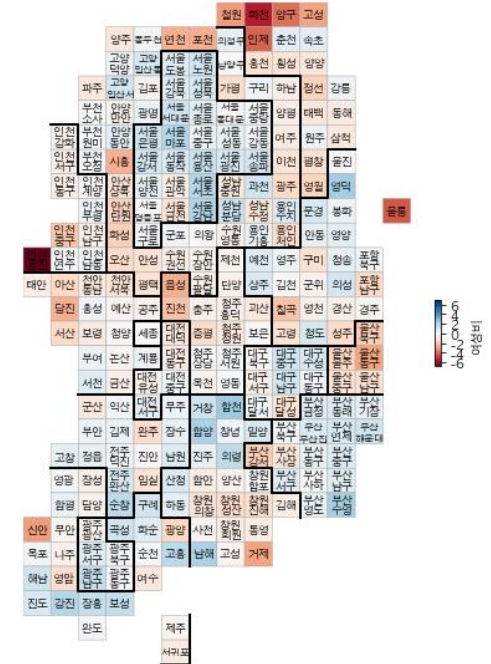
    plt.tight_layout()
    plt.show()
```

소스코드

2. 데이터 탐색 및 가공

5. 여성비

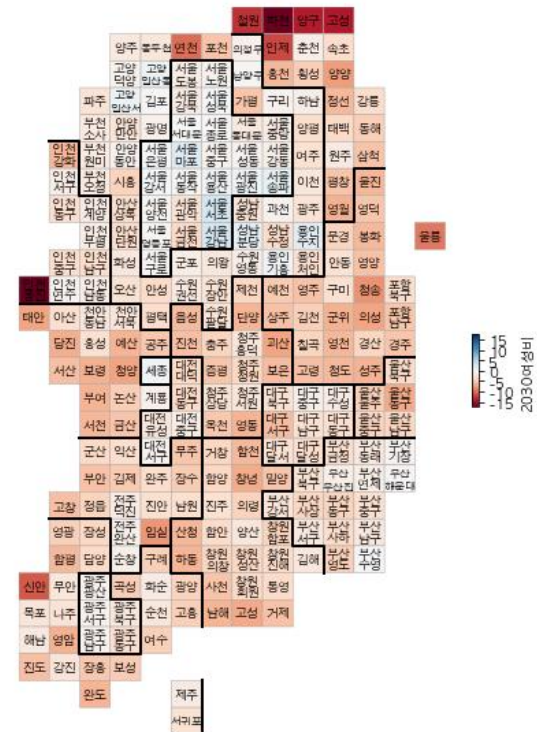
파일	소스코드
실행환경	Tf38_cpu
소스코드	$\text{pop['여성비']} = (\text{pop['인구수여자']}/\text{pop['인구수합계']} - 0.5) * 100$ $\text{drawKorea2('여성비', pop, 'RdBu')}$
결과값1	
비고	



2. 데이터 탐색 및 가공

6. 2030여성비

파일	소스코드
실습환경	Tf38_cpu
소스코드	<pre>pop['2030여성비'] = (pop['20-39세여자']/pop['20-39세합계']-0.5)*100 drawKorea2('2030여성비', pop, 'RdBu')</pre>
결과값1	
비고	



Unit 2



Quiz



❖ 대화하는 프로그램 만들기

- ❖ 자동 판매기를 시뮬레이션하는 프로그램을 작성하여 보자. 사용자는 1000원짜리 지폐와 500원짜리 동전, 100원짜리 동전을 사용할 수 있다. 물건값을 입력하고 1000원권, 500원짜리 동전, 100원짜리 동전의 개수를 입력하면 거스름돈을 계산하여서 동전으로 반환한다.

물건값을 입력하시오: 750

1000원 지폐개수: 1

500원 동전개수: 0

100원 동전개수: 0

500원= 0 100원= 2 10원= 5 1원= 0

```
itemPrice = int(input("물건값을 입력하시오: "))
note = int(input("1000원 지폐개수: "))
coin500 = int(input("500원 동전개수: "))
coin100 = int(input("100원 동전개수: "))

change = note*1000 + coin500*500 + coin100*100 - itemPrice

# 거스름돈(500원 동전 개수)을 계산한다.
nCoin500 = change//500
change = change%500

# 거스름돈(100원 동전 개수)을 계산한다.
nCoin100 = change//100
change = change%100

# 거스름돈(10원 동전 개수)을 계산한다.
nCoin10 = change//10
change = change%10

# 거스름돈(1원 동전 개수)을 계산한다.
nCoin1 = change

print("500원=", nCoin500, "100원=", nCoin100, "10원=", nCoin10, "1원=", nCoin1)
```

```
물건값을 입력하시오: 10
1000원 지폐개수: 10
500원 동전개수: 50
100원 동전개수: 10
500원= 71 100원= 4 10원= 9 1원= 0
```

- ❖ 연락처 관리 프로그램
- ❖ 파이썬을 이용하여 연락처를 관리하는 프로그램을 작성하여 보자. 연락처 관리 프로그램은 다음과 같은 메뉴를 가져야 한다.

1. 친구 리스트 출력

2. 친구추가

3. 친구삭제

4. 이름변경

9. 종료

메뉴를 선택하시오: 2

이름을 입력하시오: 홍길동

1. 친구 리스트 출력

2. 친구추가

3. 친구삭제

4. 이름변경

9. 종료

메뉴를 선택하시오: 1

['홍길동']

...

```

menu = 0
friends = []
while menu != 9:
    print("-----")
    print("1. 친구 리스트 출력")
    print("2. 친구추가")
    print("3. 친구삭제")
    print("4. 이름변경")
    print("9. 종료")
    menu = int(input("메뉴를 선택하시오: "))
    if menu == 1:
        print(friends)
    elif menu == 2:
        name = input("이름을 입력하시오: ")
        friends.append(name)
menu = 0
friends = []
while menu != 9:
    print("-----")
    print("1. 친구 리스트 출력")
    print("2. 친구추가")
    print("3. 친구삭제")
    print("4. 이름변경")
    print("9. 종료")
    menu = int(input("메뉴를 선택하시오: "))
    if menu == 1:
        print(friends)
    elif menu == 2:
        name = input("이름을 입력하시오: ")
        friends.append(name)

```

```

-----
1. 친구 리스트 출력
2. 친구추가
3. 친구삭제
4. 이름변경
9. 종료
메뉴를 선택하시오: 1

```

- ❖ 파일에서 중복되지 않은 단어의 개수
- ❖ 텍스트 파일을 읽어서 단어를 얼마나 다양하게 사용하여 문서를 작성하였는지를 계산하는 프로그램을 작성해보자.

입력 파일 이름: proverbs.txt

사용된 단어의 개수= 18

{'travels', 'half', 'that', 'news', 'alls', 'well', 'fast', 'feather', 'flock', 'bad', 'together', 'ends', 'is', 'a', 'done', 'begun', 'birds', 'of'}

```
# 단어에서 구두점을 제거하고 소문자로 만든다.
def process(w):
    output = ""
    for ch in w:
        if( ch.isalpha() ):
            output += ch
    return output.lower()

words = set()
# 파일을 연다.
fname = input("입력 파일 이름: ")
file = open(fname, "r")
# 파일의 모든 줄에 대하여 반복한다.
for line in file:
    lineWords = line.split()
    for word in lineWords:
        words.add(process(word)) # 단어를 세트에 추가한다.
print("사용된 단어의 개수=", len(words))
print(words)
```


❖ 단어 카운터

- ❖ 사용자가 지정하는 파일을 읽어서 파일에 저장된 각각의 단어가 몇 번이나 나오는지 계산하는 프로그램을 작성하여 보자.

파일 이름: proverbs.txt

```
{'a': 1, 'done.': 1, 'that': 1, 'well.': 1, 'ends': 1, 'Well': 1, 'flock': 1, 'feather': 1, "All's": 1, 'Birds': 1, 'together.': 1, 'of': 1, 'fast.': 1, 'begun': 1, 'half': 1, 'well': 1, 'travels': 1, 'news': 1, 'is': 1, 'Bad': 1}
```

```
fname = input("파일 이름: ")
file = open(fname, "r")

table = dict()
for line in file:
    words = line.split()
    for word in words:
        if word not in table:
            table[word] = 1
        else:
            table[word] += 1

print(table)
```

❖ 은행 계좌

- ❖ 우리는 은행 계좌에 돈을 저금할 수 있고 인출할 수도 있다. 은행 계좌를 클래스로 모델링하여 보자. 은행 계좌는 현재 잔액(balance)만을 인스턴스 변수로 가진다. 생성자와 인출 메소드 `withdraw()`와 저축 메소드 `deposit()` 만을 가정하자.

통장에서 100 가 출금되었음
통장에 10 가 입금되었음

```
class BankAccount:
    def __init__(self):
        self.__balance = 0

    def withdraw(self, amount):
        self.__balance -= amount
        print("통장에 ", amount, "가 입금되었음")
        return self.__balance

    def deposit(self, amount):
        self.__balance += amount
        print("통장에서 ", amount, "가 출금되었음")
        return self.__balance

a = BankAccount()
a.deposit(100)
a.withdraw(10)
```

통장에서 100 가 출금되었음
통장에 10 가 입금되었음
90