

Chapter 07. Population



배경

어느날 문득...

기사 한 줄을 읽게 됩니다...

228곳 지자체 소멸위험도는?

2017년 소멸위험지수

0.2 미만	소멸 고위험
0.2~0.5 미만	소멸 위험 진입
0.5~1.0 미만	소멸 주의 단계
1.0~1.5 미만	정상 단계
1.5 이상	소멸 저위험

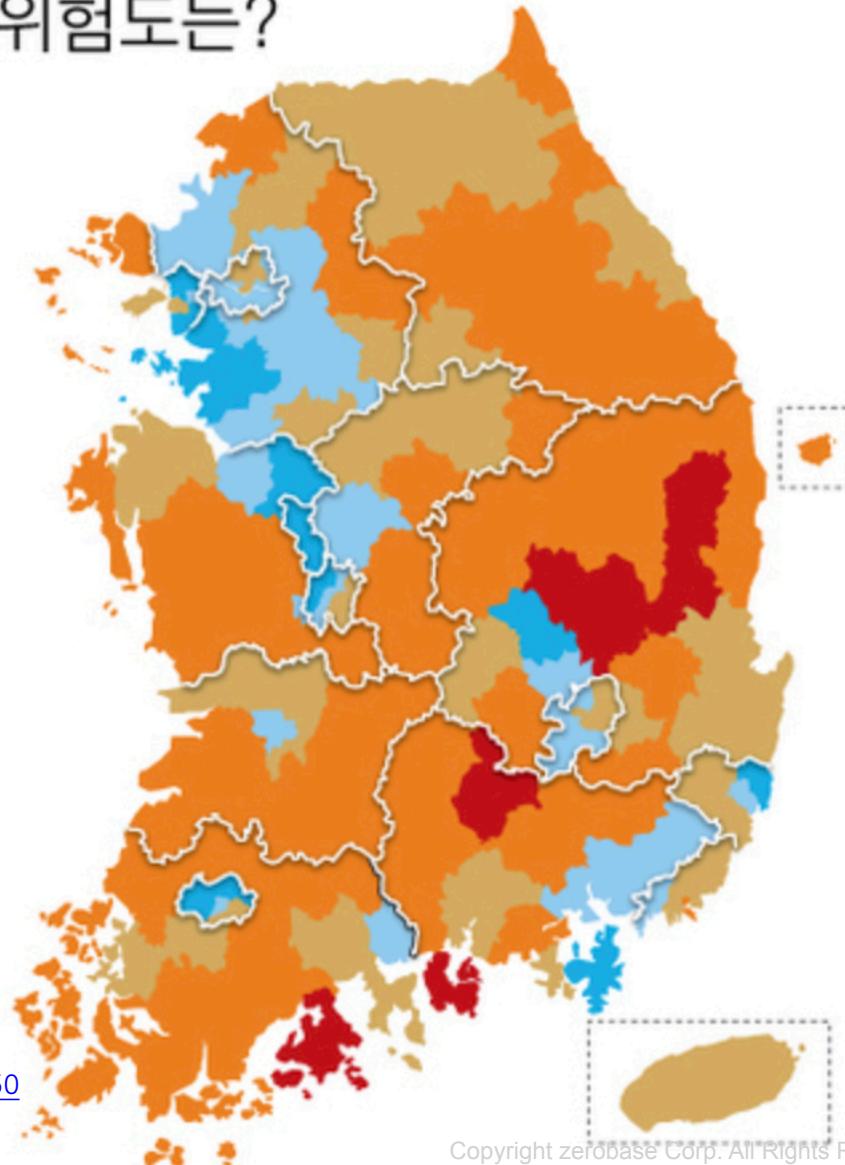
소멸위험지수는?

65세 고령인구 대비 20~39세

여성인구 비중. **0.5 이하**면

30년 내 소멸 가능성 크다고 판단

출처: <https://news.joins.com/article/21902650>



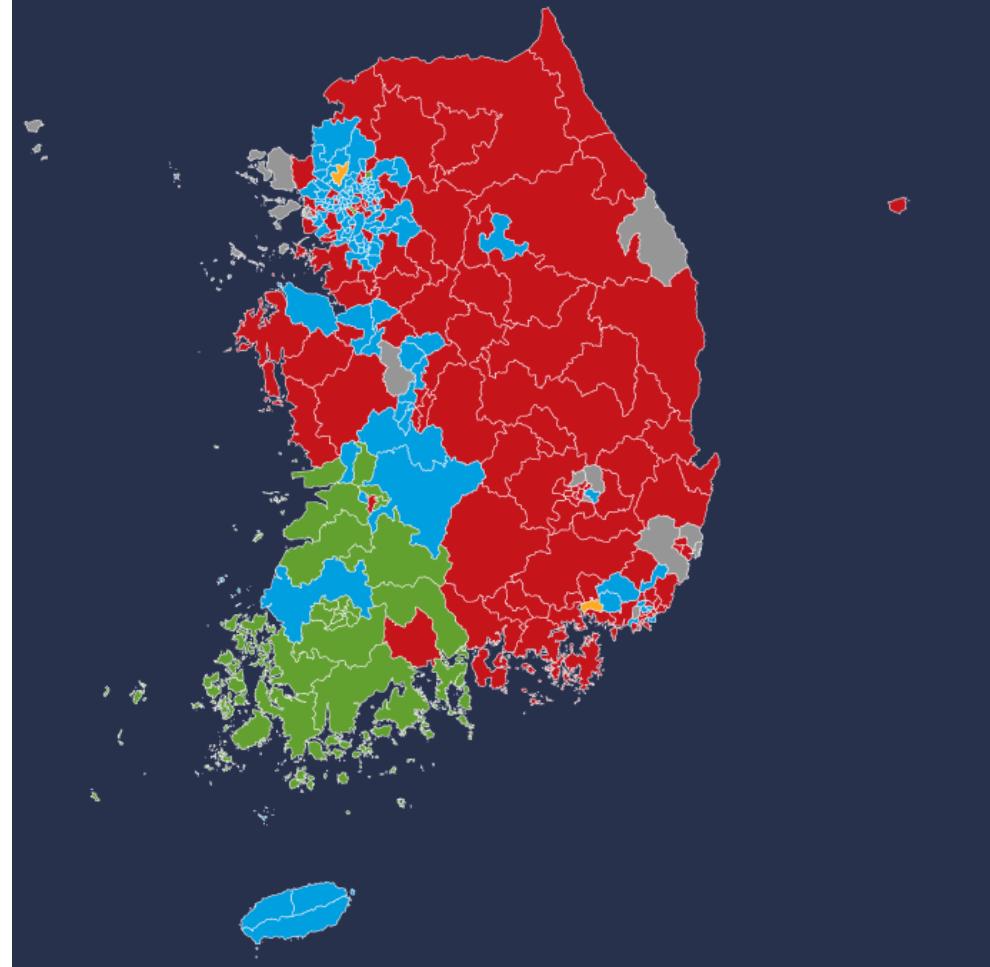
인구 소멸 위기 지역:

‘한국의 ‘지방소멸’에 관한 7가지 분석’ 보고서를 쓴 이상호 한국고용정보원 부연구위원의 분석 방법을 이용. 65세 이상 노인 인구와 20~39세 여성 인구를 비교해 젊은 여성 인구가 노인 인구의 절반에 미달할 경우 ‘소멸 위험 지역’으로 분류하는 방식이다.

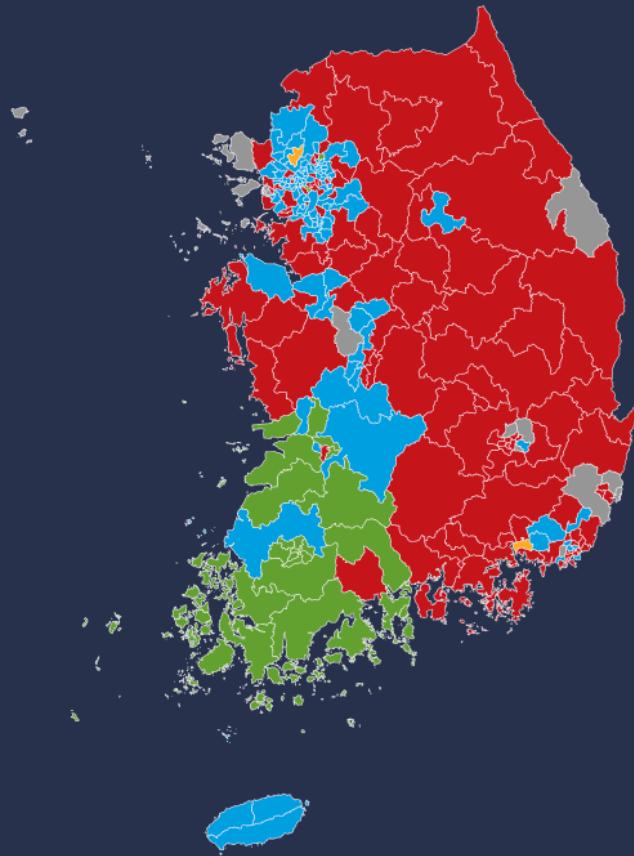
그리고 또 하나의 글도 읽습니다.
그 때가 2016년 지방선거 후...

당시
한나라당이
압승했나???

제20대 총선 당선자



제20대 총선 당선자



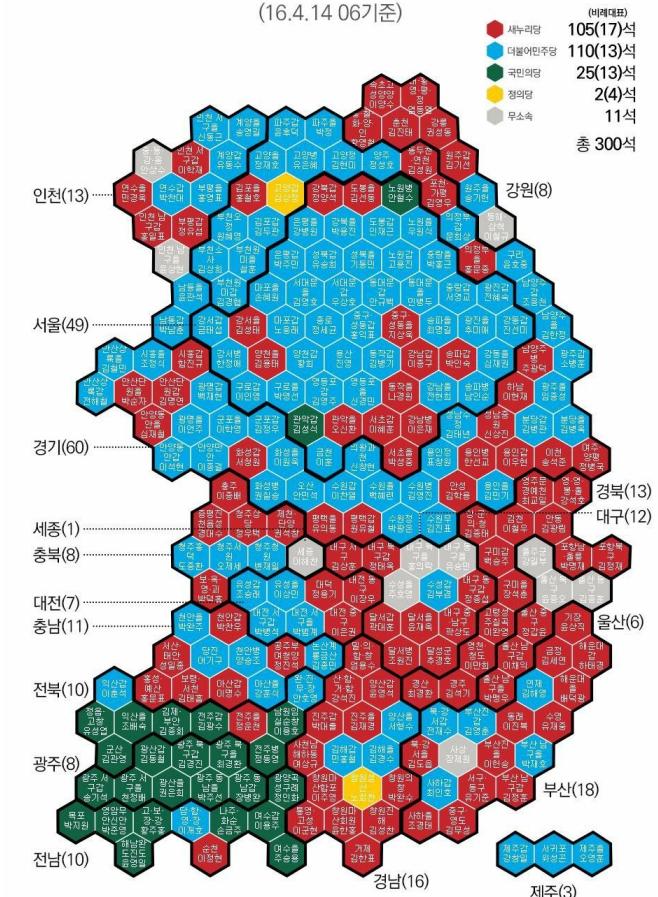
정당	지역구	비례대표	총 의석수
더불어민주당	110	13	123
새누리당	105	17	122
국민의당	25	13	38
정의당	2	4	6
무소속	11	-	11

지도의 영역과
선거 결과가 정반대의
의미로 다가옴

그래서
저런 스타일로
만들어 보고 싶었던 거죠...
일명 카르토그램

제20대 국회의원 총선거
개표결과

(16.4.14 06기준)



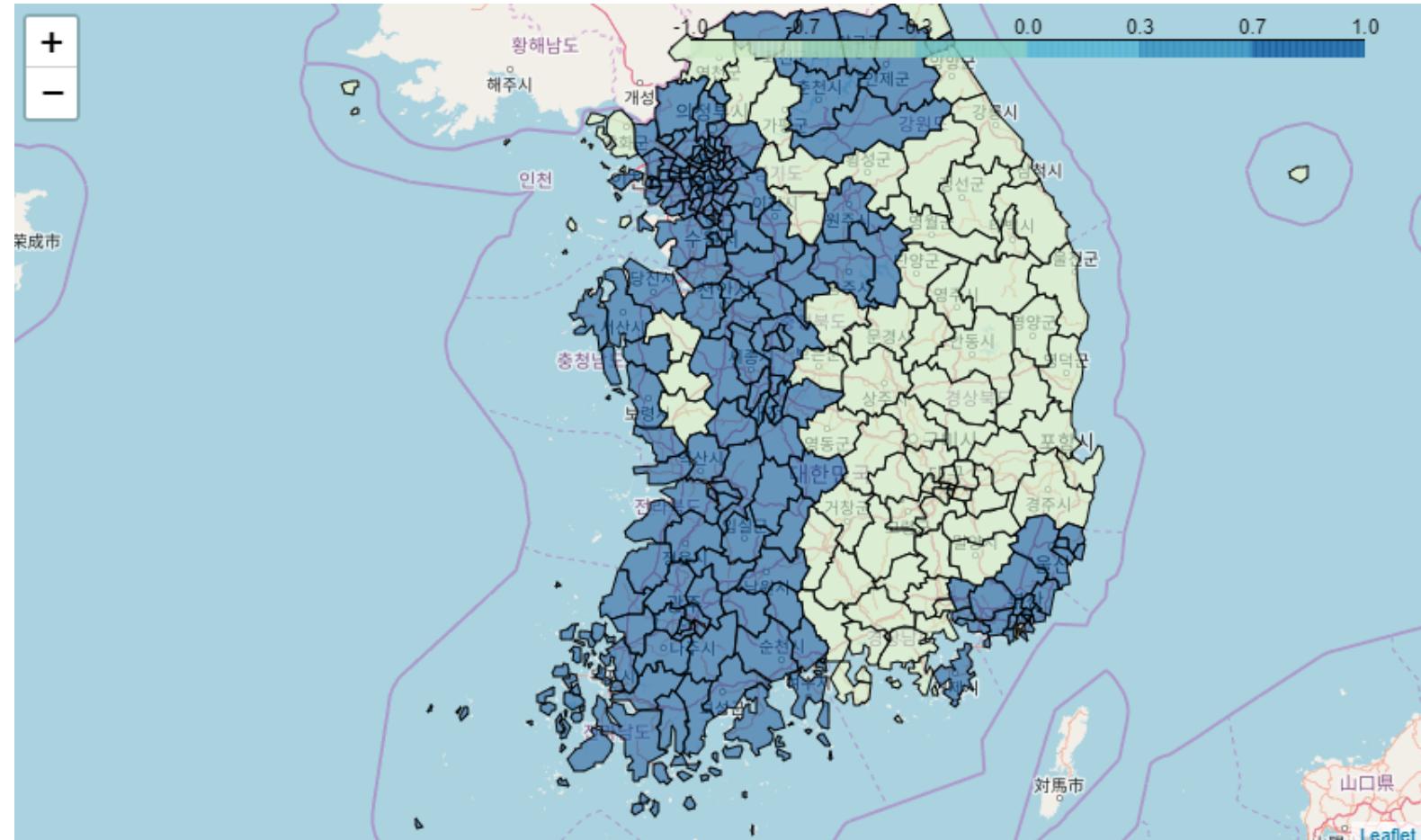
위 그림은 카트로그램(Cartogram)을 이용하여 모든 지역구를 같은 크기로 만들어 개표결과 당선인의 정당별 색깔별로 표시한 것입니다. (출처 : 중앙선거관리위원회)

제작 이종호

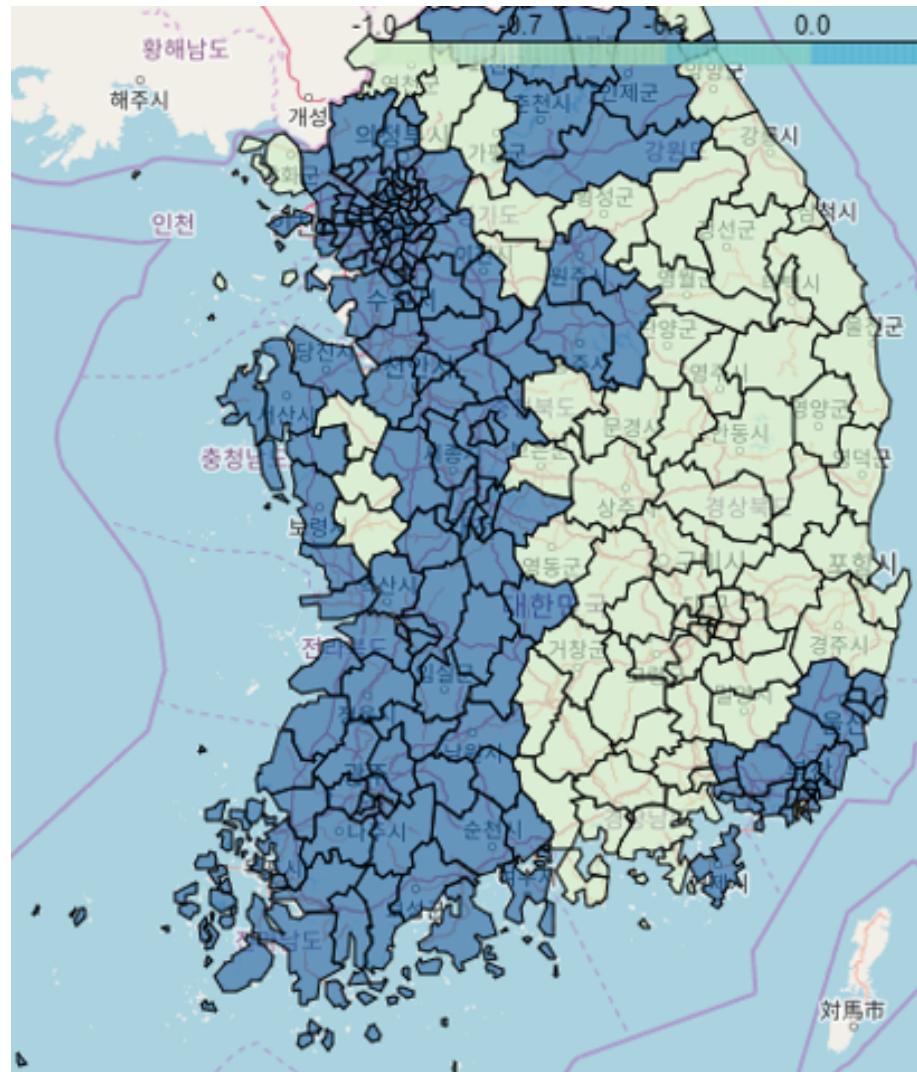
그리피고 캐릭터 디자인

OhmyNews

출처: https://namu.wiki/w/제20대_국회의원_선거/정당별_결과
Copyright zerobase Corp. All Rights Reserved



- 19대 대선 당시 문재인 후보 대 홍준표 후보의 대결 결과 마치...
- 두 사람은 박빙의 승부를 한 듯 보인다.



	양주	동두천	연천	포천	의정부	철원	화천	양구	고성
인천	고양 덕양	고양 일산서	서울 도봉	서울 노원	인제	춘천	속초		
강화	부천 소사	인양 만안	서울 강북	서울 성북	남양주	홍천	횡성	양양	
옹진	부천 일이	인양 동안	서울 은평	서울 종로	서울 동대문	서울 중랑	양평	태백	동해
인천	부천 오정	시흥	서울 강서	서울 마포	서울 성동	서울 강동	여주	원주	삼척
중구	인천 구로	인천 상록	서울 양천	서울 관악	서울 용산	서울 광진	이천	평창	울진
중구	인천 남구	인천 단원	서울 금천	서울 서초	성남 중원	과천	광주	영월	영덕
옹진	인천 연수	화성	군포	의왕	성남 성남	성남 분당	용인 수지	문경	봉화
태안	아산	오산	안성	수원 권선	수원 장안	제천	예천	영주	구미
	당진	당진	평택	수원 팔달	단양	상주	김천	군위	청송
서산	보령	청양	음성	진천	충주	청주 흥덕	괴산	칠곡	영천
	부여	세종	세종	충주	증평	청주 청원	보은	고령	청도
서천	금산	계룡	대전 유성	대전 동구	청주 상당	청주 서원	대구 북구	성주	성주
	군산	익산	대전 서구	대전 동구	옥천	영동	대구 서구	대구 달서	부산 금정
부안	김제	완주	무주	대구 남구	거창	함천	대구 달성	부산 부산	부산 동래
고창	정읍	전주 덕진	진안	부산 북구	함양	창녕	부산 부산	부산 사상	부산 기장
영광	장성	전주 완산	남원	부산 연제	밀양	밀양	부산 부산	부산 동구	부산 중구
	함평	임실	진주	부산 부산	부산 합포	부산 부산	부산 부산	부산 사하	부산 남구
신안	무안	광주 광산	의령	부산 부산	부산 부산	부산 부산	부산 부산	부산 부산	부산 부산
목포	나주	광주 서구	광주 북구	부산 부산	부산 부산	부산 부산	부산 부산	부산 부산	부산 부산
해남	영암	광주 남구	광주 동구	여수	여수	여수	여수	여수	여수
진도	강진	장흥	보성	완도	완도	완도	제주	제주	제주
							서귀포		

목표.

1. 인구 소멸 위기 지역 파악
2. 인구 소멸 위기 지역의 지도 표현
3. 지도 표현에 대한 카르토그램 표현

먼저...

Folium으로 그리기 위해서는

→ 각 시도별 경계선이 있는

json 파일이 필요

- 지도시각화에 필요한 json 파일을 구해야 한다
- 현재 무료로 구할 수 있는 json 파일은 Lucy Park님의 자료가 유일하다

The screenshot shows a GitHub repository page for 'southkorea/southkorea-maps'. The repository has 38 stars, 373 forks, and 164 issues. It contains 1 branch and 0 tags. The code history shows several commits from 'e9t' and others, including merges from 'achimkoh/master'. The repository description states: "South Korea administrative divisions in ESRI Shapefile, GeoJSON and TopoJSON formats." It includes tags for map, topojson, shapefile, geographical, korean, and south-korea. The README.md file is present with the title "South Korea Maps" and a brief description of the data format.

Search or jump to... Pull requests Issues Marketplace Explore

southkorea / southkorea-maps

Code Issues 6 Pull requests Actions Projects Wiki Security Insights

master 1 branch 0 tags Go to file Add file Code

e9t Merge pull request #9 from achimkoh/master ... fe65e05 on 26 Dec 2018 125 commits

gadm Update gadm downloader 6 years ago

kostat upload 2018 kostat data 3 years ago

license Merge branch 'master' of github.com:southkorea/southkorea-maps 6 years ago

popong Remove temp files 5 years ago

static Add precinct map screenshot 5 years ago

wikimedia Upload wikimedia map file 6 years ago

.gitignore Merge branch 'master' of github.com:southkorea/southkorea-maps 6 years ago

Makefile Read unicode in CSVs 8 years ago

README.md Update README.md 5 years ago

README.md

South Korea Maps

This repo hosts South Korea administrative division geodata in open formats that can be used to build static and interactive maps (e.g. with D3), and was inspired by swiss-maps.

To see maps for Seoul, go to <https://github.com/southkorea/seoul-maps>

About

South Korea administrative divisions in ESRI Shapefile, GeoJSON and TopoJSON formats.

map topojson shapefile
geographical korean south-korea

Readme

Releases

No releases published

Packages

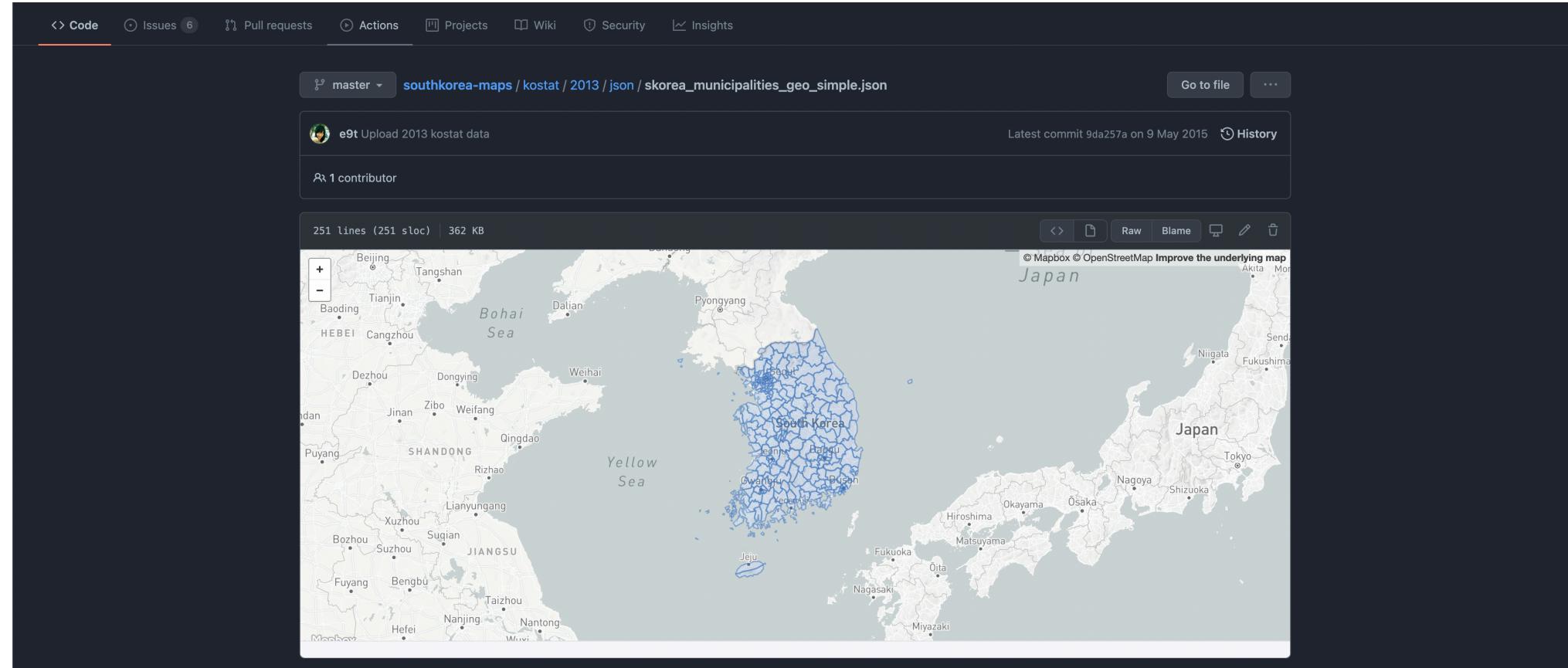
No packages published

Contributors 4

e9t Lucy Park
minsukkahng Minsuk Kahng
cornchz Cheol Kang
achimkoh 고아침

출처: <https://github.com/southkorea/southkorea-maps>

- 오래된 데이터라 한계는 있지만..
- 서울을 대상으로 하는 것에는 문제가 없다



- https://github.com/southkorea/southkorea-maps/blob/master/kostat/2013/json/skorea_municipalities_geo_simple.json

```

1 {"type":"FeatureCollection","features":[{"type":"Feature","properties":{"code": "39020","name": "서귀포시","name_eng": "Seogwipo-si","base_year": "2013"},"geometry":{}}
2 {"type":"Feature","properties":{"code": "39010","name": "제주시","name_eng": "Jeju-si","base_year": "2013"},"geometry": {"type": "MultiPolygon","coordinates": [[[126.91
3 {"type": "Feature","properties": {"code": "38400","name": "합천군","name_eng": "Hapcheon-gun","base_year": "2013"}, "geometry": {"type": "Polygon","coordinates": [[[128.160
4 {"type": "Feature","properties": {"code": "38390","name": "거창군","name_eng": "Geochang-gun","base_year": "2013"}, "geometry": {"type": "Polygon","coordinates": [[[128.080
5 {"type": "Feature","properties": {"code": "38380","name": "함양군","name_eng": "Hamyang-gun","base_year": "2013"}, "geometry": {"type": "Polygon","coordinates": [[[127.681
6 {"type": "Feature","properties": {"code": "38370","name": "산청군","name_eng": "Sancheong-gun","base_year": "2013"}, "geometry": {"type": "Polygon","coordinates": [[[127.850
7 {"type": "Feature","properties": {"code": "38360","name": "하동군","name_eng": "Hadong-gun","base_year": "2013"}, "geometry": {"type": "Polygon","coordinates": [[[127.6909
8 {"type": "Feature","properties": {"code": "38350","name": "남해군","name_eng": "Namhae-gun","base_year": "2013"}, "geometry": {"type": "MultiPolygon","coordinates": [[[128.0
9 {"type": "Feature","properties": {"code": "38340","name": "고성군","name_eng": "Goseong-gun","base_year": "2013"}, "geometry": {"type": "Polygon","coordinates": [[[128.350
10 {"type": "Feature","properties": {"code": "38330","name": "창녕군","name_eng": "Changnyeong-gun","base_year": "2013"}, "geometry": {"type": "Polygon","coordinates": [[[128.3
11 {"type": "Feature","properties": {"code": "38320","name": "함안군","name_eng": "Haman-gun","base_year": "2013"}, "geometry": {"type": "Polygon","coordinates": [[[128.43872
12 {"type": "Feature","properties": {"code": "38310","name": "의령군","name_eng": "Uiryeong-gun","base_year": "2013"}, "geometry": {"type": "Polygon","coordinates": [[[128.37
13 {"type": "Feature","properties": {"code": "38115","name": "창원시진해구","name_eng": "Jinhaegu","base_year": "2013"}, "geometry": {"type": "Polygon","coordinates": [[[128.74
14 {"type": "Feature","properties": {"code": "38114","name": "창원시마산회원구","name_eng": "Masanhewongu","base_year": "2013"}, "geometry": {"type": "Polygon","coordinates": [[[128.74
15 {"type": "Feature","properties": {"code": "38113","name": "창원시마산합포구","name_eng": "Masanhappogu","base_year": "2013"}, "geometry": {"type": "Polygon","coordinates": [[[128.74
16 {"type": "Feature","properties": {"code": "38112","name": "창원시성산구","name_eng": "Seongsangu","base_year": "2013"}, "geometry": {"type": "Polygon","coordinates": [[[128.74
17 {"type": "Feature","properties": {"code": "38111","name": "창원시의창구","name_eng": "Uichanggu","base_year": "2013"}, "geometry": {"type": "Polygon","coordinates": [[[128.5
18 {"type": "Feature","properties": {"code": "38100","name": "양산시","name_eng": "Yangsan-si","base_year": "2013"}, "geometry": {"type": "Polygon","coordinates": [[[128.9967

```

```

skorea_municipalities_geo_simple.json  ●  05. korea_pw.json  ✘
16 {"type": "Feature", "id": "창원 합포", "properties": {"code": "38113", "name": "창원시마산합포구", "name_en": "Masan-Happo-gu, Changwon-si", "name_eng": "Masan-Happo-gu, Changwon-si"}, "name_ko": "창원 합포", "name_kor": "창원시 마산합포구", "name_english": "Masan-Happo-gu, Changwon-si"}, "name_ko": "창원 합포", "name_kor": "창원시 마산합포구", "name_english": "Masan-Happo-gu, Changwon-si"}, "name_ko": "창원 성산", "name_kor": "창원시 성산구", "name_english": "Seongsan-gu, Changwon-si"}, "name_ko": "창원 의창", "name_kor": "창원시 의창구", "name_english": "Uichang-gu, Changwon-si"}, "name_ko": "양산", "name_kor": "양산시", "name_english": "Yangsan-si"}, "name_ko": "거제", "name_kor": "거제시", "name_english": "Geojesi"}, "name_ko": "밀양", "name_kor": "밀양시", "name_english": "Miryang-si"}, "name_ko": "김해", "name_kor": "김해시", "name_english": "Gimhae-si"}, "name_ko": "사천", "name_kor": "사천시", "name_english": "Sacheon-si"}, "name_ko": "통영", "name_kor": "통영시", "name_english": "Tongyeong-si"}, "name_ko": "진주", "name_kor": "진주시", "name_english": "Jinju-si"}, "name_ko": "울릉", "name_kor": "울릉군", "name_english": "Uljin-gun"}, "name_ko": "울진", "name_kor": "울진군", "name_english": "Uljin-gun"}, "name_ko": "봉화", "name_kor": "봉화군", "name_english": "Bonghwa-gun"}, "name_ko": "예천", "name_kor": "예천군", "name_english": "Yecheon-gun"}, "name_ko": "칠곡", "name_kor": "칠곡군", "name_english": "Chilgok-gun"}, "name_ko": "성주", "name_kor": "성주군", "name_english": "Seongju-gun"}, "name_ko": "고령", "name_kor": "고령군", "name_english": "Goryeong-gun"}, "name_ko": "청도", "name_kor": "청도군", "name_english": "Cheongdo-gun"}, "name_ko": "영덕", "name_kor": "영덕군", "name_english": "Yeongdeok-gun"}, "name_ko": "영양", "name_kor": "영양군", "name_english": "Yeongnang-gun"}, "name_ko": "청송", "name_kor": "청송군", "name_english": "Cheongsong-gun"}, "name_ko": "의성", "name_kor": "의성군", "name_english": "Uiseong-gun"}, "name_ko": "군위", "name_kor": "군위군", "name_english": "Gunwi-gun"}, "name_ko": "경산", "name_kor": "경산시", "name_english": "Gyeongsan-si"}]
```

• 이건 제가 만들어서 배포합니다~😊

Folium용 Json 파일에는 ID가 준비되었니
인구현황 데이터에도 ID가 필요하다
추후 Cartogram으로 그릴때도
ID를 동일하게 맞추면 좋겠다

high-throughput

혜식이의 떼로 보는 세상

- 혜식이라는 분의 블로그에서 해답을 얻음~~~~
- <http://openlook.org/wp/2016/12/>

옥스포드 나노포어 12월 제품 업데이트

December 17, 2016 by hyeshik

2007년 스티브 잡스 발표 기억하시나요? 화면에 전화기, 터치스크린 아이팟, 인터넷 장비 셋을 띄워놓고 빙글 빙글 돌리면서 “Are you getting it?” 하다가 아이폰을 뽕! 하고 발표한 그 맥월드 키노트 말이죠. 이 키노트 이후로는 IT에 크게 관심 없는 사람도 애플이 신제품에 대해 발표를 할 때마다 괜히 설레며 새벽에 보기도 하고, 아침에 기사 검색도 해 보곤 합니다. 요새 바이오텍 업계에서 그런 회사가 하나 등장했습니다. 바로 DNA시퀀서를 만드는 옥스포드 나노포어 (ONT)입니다. 이 회사에서 제품 업데이트를 발표할 때마다 수천 명이 라이브스 트림으로 보고, 트위터가 떠들썩해집니다. 아직 엔드유저 제품도 아닌 걸 고려하면 굉장한 반응입니다.

ONT는 보통 분기당 한 번씩 제품 업데이트를 발표하고 있습니다. 업데이트가 너무 빨라서 관심을 기울이고 있어도 따라가기 힘들 지경인데요. 신제품 소식을 듣고 주문하고 받아서 실험하고 잠깐 분석 좀 하다가, 다음 실험을 위해 또 주문하려고 가 보면 전에 주문했던 버전은 벌써 단종되고 없을 정도입니다. -.-. 얼마 전 12월 2일까

찾아보기

Search ...

누구?

장혜식

(Hyeshik Chang)

내일을 사랑하는 소년(!)



JUPYTER

FAQ

</>

"버거 지수"는 진짜 도시의 발전 수준을 반영할까?

by Hyeshik on Dec 26, 2014

2015년 1월 26일 1차 개정

밀폭도 @Godtsune_miku

"한 도시의 발전 수준은 (버거킹의 개수+맥도날드의 개수+KFC의 개수)/롯데리아의 개수를 계산하여 높게 나올수록 더 발전된 도시라고 할 수 있다"

https://twitter.com/godtsune_miku/status/513648274406789120

사람이 적은 곳일 수록 롯데리아의 개수가 많은 것 같긴 한데.. 진짜로 그럴려나?

자 그럼..

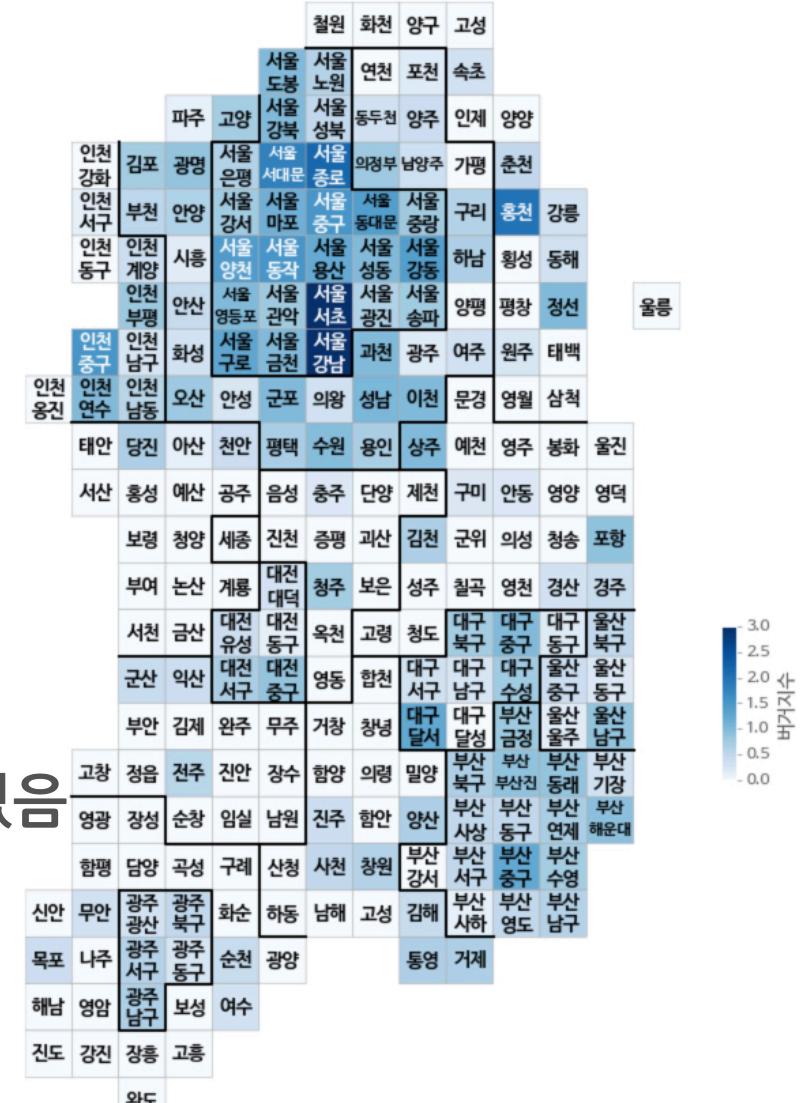
$\xi = \frac{B+M+K}{L}$ 을 계산해 보자. 시군구 단위로. (ξ 는 버거의 옆모습을 닮아서..ㅋ)

일단 준비~

- 실제로는 이 글에서 영감을 받음

그러나 혜식님은 본인의 코드는 공개했지만
그 코드가 만들어진 원리는 공개하지 않아서
우리에게 딱 맞게 사용할려면 노력이 필요...

혜식님의 방식에서도
ID 역할을 하는 것이 필요하다는 것은 알 수 있음



```
In [74]: def draw_blockcolormap(tbl, datacol, vmin, vmax, whitelabelmin, cmapname, gamma, datalabel, dataticks):
    cmap = colors.LinearSegmentedColormap(cmapname + 'custom',
                                           getattr(_cm, '_%s_data' % format(cmapname)), gamma=gamma)
    cmap.set_bad('white', 1.)

    mapdata = tbl.pivot(index='y', columns='x', values=datacol)
    masked_mapdata = np.ma.masked_where(np.isnan(mapdata), mapdata)

    plt.figure(figsize=(9, 16))
    plt.pcolor(masked_mapdata, vmin=vmin, vmax=vmax, cmap=cmap,
               edgecolor='aaaaaa', linewidth=0.5)

    # 지역 이름 표시
    for idx, row in tbl.iterrows():
        annocolor = 'white' if row[datacol] > whitelabelmin else 'black'

        # 광역시는 구 이름이 겹치는 경우가 많아서 시단위 이름도 같이 표시한다. (중구, 서구)
        if row['d1'].endswith('시') and not row['d1'].startswith('세종'):
            dispname = '%s %s' % (row['d1'][-2:], row['d2'][:-1])
            if len(row['d2']) <= 2:
                dispname += row['d2'][-1]
        else:
            dispname = row['d2'][:-1]

        # 서대문구, 서귀포시 같이 이름이 3자 이상인 경우에 작은 글자로 표시한다.
        if len(dispname.splitlines())[-1] >= 3:
            fontsize, linespacing = 12, 1.2
        else:
            fontsize, linespacing = 14, 1.03

        plt.annotate(dispname, (row['x']+0.5, row['y']+0.5), weight='bold',
                     fontsize=fontsize, ha='center', va='center', color=annocolor,
                     linespacing=linespacing)

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

    plt.gca().invert_yaxis()
    plt.gca().set_aspect(1)

    plt.axis('off')
```

혜식님의 원본코드는 결과만 나와있어서
그 과정을 알 수 없으니
원하는데로 편집하기가 쉽지 않다…

일단, 데이터를 정리하고
다시 보자…

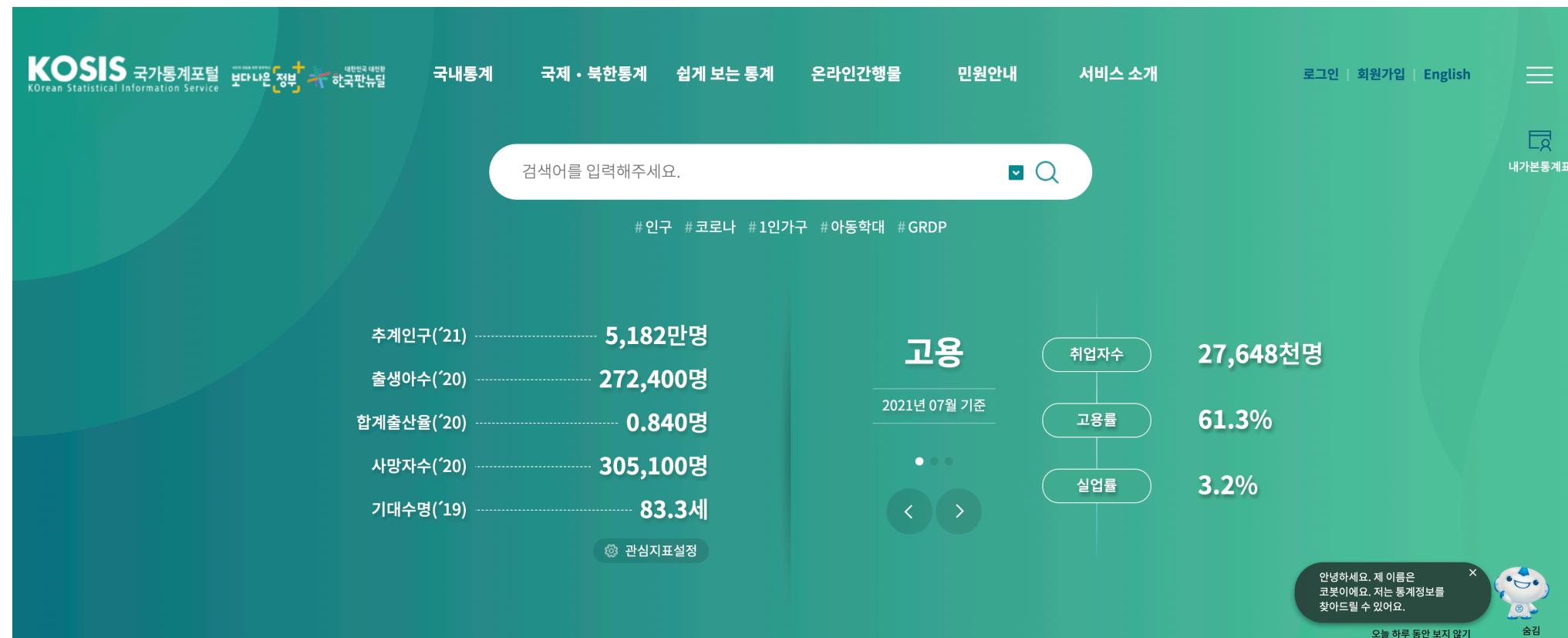
OK~~ ! 뭐든…

일단, 데이터를 확보하고…

인구 소멸 위기 지역을 계산하고

각 지역별 고유한 ID를 부여하는…

순서로 작업하면 될 듯~~~



- <https://kosis.kr/index/index.do>
- 본래는 여기서 얻으면 되는데… 데이터를 얻는 과정이 각자 너무 다를 확률이 있음

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P		
1	행정구역(동읍면)별(1)	행정구역(동읍면)별(2)	항목	2016													
				계	20 - 24세	25 - 29세	30 - 34세	35 - 39세	65 - 69세	70 - 74세	75 - 79세	80 - 84세	85 - 89세	90 - 94세	95 - 99세	100+	
3	전국	소계	총인구수 (명)	#####	#####	#####	#####	#####	#####	#####	#####	#####	909,130	416,164	141,488	34,844	17,562
4			남자인구수 (명)	#####	#####	#####	#####	#####	#####	806,680	600,607	319,391	113,221	32,695	7,658	4,137	
5			여자인구수 (명)	#####	#####	#####	#####	#####	#####	974,549	857,283	589,739	302,943	108,793	27,186	13,425	
6	서울특별시	소계	총인구수 (명)	9,930,616	690,728	751,973	803,507	817,467	448,956	350,580	251,961	141,649	66,067	24,153	7,058	5,475	
7			남자인구수 (명)	4,876,789	347,534	372,249	402,358	410,076	211,568	163,766	112,076	54,033	19,595	6,146	1,900	1,406	
8			여자인구수 (명)	5,053,827	343,194	379,724	401,149	407,391	237,388	186,814	139,885	87,616	46,472	18,007	5,158	4,069	
9		종로구	총인구수 (명)	152,737	11,379	11,891	10,684	10,379	7,411	6,636	5,263	3,104	1,480	602	234	220	
10			남자인구수 (명)	75,201	5,620	6,181	5,387	5,034	3,411	3,009	2,311	1,289	506	207	89	73	
11			여자인구수 (명)	77,536	5,759	5,710	5,297	5,345	4,000	3,627	2,952	1,815	974	395	145	147	
12		중구	총인구수 (명)	125,249	8,216	9,529	10,332	10,107	6,399	5,313	4,127	2,502	1,260	469	158	160	
13			남자인구수 (명)	62,204	4,142	4,792	5,192	5,221	3,113	2,405	1,752	929	414	132	56	51	
14			여자인구수 (명)	63,045	4,074	4,737	5,140	4,886	3,286	2,908	2,375	1,573	846	337	102	109	
15		용산구	총인구수 (명)	230,241	14,317	16,972	19,032	19,127	10,675	9,093	7,477	4,553	2,254	916	264	315	
16			남자인구수 (명)	111,601	6,937	8,373	9,455	9,434	4,834	3,975	3,094	1,739	750	284	102	88	
17			여자인구수 (명)	118,640	7,380	8,599	9,577	9,693	5,841	5,118	4,383	2,814	1,504	632	162	227	
18		성동구	총인구수 (명)	299,259	20,813	23,383	25,507	25,979	12,938	10,734	7,989	4,450	1,944	678	209	198	
19			남자인구수 (명)	148,326	10,609	11,848	12,692	13,154	6,090	4,863	3,480	1,698	580	184	56	54	
20			여자인구수 (명)	150,933	10,204	11,535	12,815	12,825	6,848	5,871	4,509	2,752	1,364	494	153	144	

- 데이터를 살펴보자… 인덱스와 컬럼 역할을 하는 축의 특성을 유심히…

데이터 읽고 인구 소멸 지역 계산하기



```
In [1]: import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import set_matplotlib_hangul  
  
get_ipython().run_line_magic("matplotlib", "inline")
```

Hangul OK in your MAC !!!

```
In [2]: population = pd.read_excel("../data/07_population_raw_data.xlsx", header=1)
population.fillna(method="pad", inplace=True)
population.head()
```

Out[2]:

	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	전국	소계	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	전국	소계	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
3	서울특별시	소계	총인구수(명)	9930616.0	690728.0	751973.0	803507	817467.0	448956.0	350580.0	251961	141649.0	66067.0	24153.0	7058	5475.0
4	서울특별시	소계	남자인구수(명)	4876789.0	347534.0	372249.0	402358	410076.0	211568.0	163766.0	112076	54033.0	19595.0	6146.0	1900	1406.0

- 엑셀을 읽어보자.

```
In [3]: population.rename(
    columns={"행정구역(동읍면)별(1)": "광역시도", "행정구역(동읍면)별(2)": "시도", "계": "인구수"}, inplace=True
)
population.head()
```

Out[3]:

	광역시 도	시 도	항목	인구수	20 - 24세	25 - 29세	30 - 34 세	35 - 39세	65 - 69세	70 - 74세	75 - 79 세	80 - 84세	85 - 89세	90 - 94세	95 - 99 세	100+
0	전국	소 계	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	전국	소 계	남자인구 수(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	전국	소 계	여자인구 수(명)	25868622.0	1663934.0	1534379.0	1711114	1971007.0	1164950.0	974549.0	857283	589739.0	302943.0	108793.0	27186	13425.0
3	서울특 별시	소 계	총인구수 (명)	9930616.0	690728.0	751973.0	803507	817467.0	448956.0	350580.0	251961	141649.0	66067.0	24153.0	7058	5475.0
4	서울특 별시	소 계	남자인구 수(명)	4876789.0	347534.0	372249.0	402358	410076.0	211568.0	163766.0	112076	54033.0	19595.0	6146.0	1900	1406.0

- 일부 컬럼의 이름도 바꿔주자

A	B	C	D	E	F	G	H	I	J	K	
1	행정구역(동읍면)별(1)	행정구역(동읍면)별(2)	항목	2016							
2			계	20 - 24세	25 - 29세	30 - 34세	35 - 39세	65 - 69세	70 - 74세	75 - 79세	
3	전국	소계	총인구수 (명)	51,696,216	3,541,061	3,217,367	3,517,868	4,016,272	2,237,345	1,781,229	1,457,890
4			남자인구수 (명)	25,827,594	1,877,127	1,682,988	1,806,754	2,045,265	1,072,395	806,680	600,607
5			여자인구수 (명)	25,868,622	1,663,934	1,534,379	1,711,114	1,971,007	1,164,950	974,549	857,283
6	서울특별시	소계	총인구수 (명)	9,930,616	690,728	751,973	803,507	817,467	448,956	350,580	251,961
7			남자인구수 (명)	4,876,789	347,534	372,249	402,358	410,076	211,568	163,766	112,076
8			여자인구수 (명)	5,053,827	343,194	379,724	401,149	407,391	237,388	186,814	139,885
9	종로구	총인구	총인구수 (명)	152,737	11,379	11,891	10,684	10,379	7,411	6,636	5,263
10			남자인구수 (명)	75,201	5,620	6,181	5,387	5,034	3,411	3,009	2,311
11			여자인구수 (명)	77,536	5,759	5,710	5,297	5,345	4,000	3,627	2,952
12	중구	총인구	총인구수 (명)	125,249	8,216	9,529	10,332	10,107	6,399	5,313	4,127
13			남자인구수 (명)	62,204	4,142	4,792	5,192	5,221	3,113	2,405	1,752
14			여자인구수 (명)	63,045	4,074	4,737	5,140	4,886	3,286	2,908	2,375
15	용산구	총인구	총인구수 (명)	230,241	14,317	16,972	19,032	19,127	10,675	9,093	7,477
16			남자인구수 (명)	111,601	6,937	8,373	9,455	9,434	4,834	3,975	3,094
17			여자인구수 (명)	118,640	7,380	8,599	9,577	9,693	5,841	5,118	4,383
18	성동구	총인구	총인구수 (명)	299,259	20,813	23,383	25,507	25,979	12,938	10,734	7,989
19			남자인구수 (명)	148,326	10,609	11,848	12,692	13,154	6,090	4,863	3,480
20			여자인구수 (명)	150,933	10,204	11,535	12,815	12,825	6,848	5,871	4,509
21	광진구	총인구	총인구수 (명)	357,215	27,368	32,092	32,302	29,496	14,402	11,300	8,121
22			남자인구수 (명)	174,797	13,317	15,555	16,034	14,875	6,775	5,376	3,693
23			여자인구수 (명)	182,418	14,051	16,537	16,268	14,621	7,627	5,924	4,428

```
In [4]: population = population[(population["시도"] != "소계")]
population.head()
```

Out[4]:

	광역시도	시도	항목	인구수	20 - 24 세	25 - 29 세	30 - 34 세	35 - 39 세	65 - 69 세	70 - 74 세	75 - 79 세	80 - 84 세	85 - 89 세	90 - 94 세	95 - 99 세	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
9	서울특별시	중구	총인구수 (명)	125249.0	8216.0	9529.0	10332	10107.0	6399.0	5313.0	4127	2502.0	1260.0	469.0	158	160.0
10	서울특별시	중구	남자인구수 (명)	62204.0	4142.0	4792.0	5192	5221.0	3113.0	2405.0	1752	929.0	414.0	132.0	56	51.0

- 소계는 제거하자 ~~~

```
In [5]: population.is_copy = False  
  
population.rename(columns={"항목": "구분"}, inplace=True)  
  
population.loc[population["구분"] == "총인구수 (명)", "구분"] = "합계"  
population.loc[population["구분"] == "남자인구수 (명)", "구분"] = "남자"  
population.loc[population["구분"] == "여자인구수 (명)", "구분"] = "여자"  
  
population.head()
```

Out[5]:

	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
9	서울특별시	중구	합계	125249.0	8216.0	9529.0	10332	10107.0	6399.0	5313.0	4127	2502.0	1260.0	469.0	158	160.0
10	서울특별시	중구	남자	62204.0	4142.0	4792.0	5192	5221.0	3113.0	2405.0	1752	929.0	414.0	132.0	56	51.0

```
In [6]: population["20-39세"] = (
    population["20 - 24세"]
    + population["25 - 29세"]
    + population["30 - 34세"]
    + population["35 - 39세"]
)

population["65세이상"] = (
    population["65 - 69세"]
    + population["70 - 74세"]
    + population["75 - 79세"]
    + population["80 - 84세"]
    + population["85 - 89세"]
    + population["90 - 94세"]
    + population["95 - 99세"]
    + population["100+"]
)

population.head(10)
```

Out[6]:

	광역시 도	시도 구	구 분	인구수	20 - 24 세	25 - 29 세	30 - 34 세	35 - 39 세	65 - 69 세	70 - 74 세	75 - 79 세	80 - 84 세	85 - 89 세	90 - 94 세	95 - 99 세	100+	20-39세	65세이 상
6	서울특 별시	종로 구	합 계	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	서울특 별시	종로 구	남 자	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	서울특 별시	종로 구	여 자	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
9	서울특 별시	중구	합 계	125249.0	8216.0	9529.0	10332	10107.0	6399.0	5313.0	4127	2502.0	1260.0	469.0	158	160.0	38184.0	20388.0

- 소멸지역을 조사하기 위한 데이터를 만들어 두자

원하는 데이터만 가지고, 인덱스의 지역 구분도 ...
뭔가 정리가 필요함...
휴~~ 데이터의 모양이 가로에 지역,
세로에 연령대별, 남여구분이 모두 위치하는 게 좋겠다.
이럴 때 사용하는 마법의 단어는???

```
In [7]: pop = pd.pivot_table(
    population, index=["광역시도", "시도"], columns=["구분"], values=["인구수", "20-39세", "65세이상"]
)
pop
```

Out[7]:

	구분	20-39세			65세이상			인구수		
		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
	삼척시	8708.0	7115.0	15823.0	5892.0	8718.0	14610.0	35253.0	34346.0	69599.0
	속초시	9956.0	8752.0	18708.0	5139.0	7613.0	12752.0	40288.0	41505.0	81793.0
...	
충청북도	진천군	9391.0	7622.0	17013.0	4731.0	6575.0	11306.0	36387.0	33563.0	69950.0
	청원구	32216.0	27805.0	60021.0	8417.0	11914.0	20331.0	97006.0	93807.0	190813.0
천주시		128318.0	115719.0	244037.0	37882.0	53671.0	91553.0	419323.0	415874.0	835197.0

- pivot_table이다 ~~~

```
In [8]: pop[ "소멸비율" ] = pop[ "20-39세", "여자" ] / (pop[ "65세이상", "합계" ] / 2)
pop.head()
```

Out[8]:

광역시도	시도	20-39세			65세이상			인구수			소멸비율
		구분	남자	여자	합계	남자	여자	합계	남자	여자	
강원도	강릉시	26286.0	23098.0	49384.0	15767.0	21912.0	37679.0	106231.0	107615.0	213846.0	1.226041
	고성군	4494.0	2529.0	7023.0	2900.0	4251.0	7151.0	15899.0	14215.0	30114.0	0.707314
	동해시	11511.0	9753.0	21264.0	6392.0	8732.0	15124.0	47166.0	46131.0	93297.0	1.289738
	삼척시	8708.0	7115.0	15823.0	5892.0	8718.0	14610.0	35253.0	34346.0	69599.0	0.973990
	속초시	9956.0	8752.0	18708.0	5139.0	7613.0	12752.0	40288.0	41505.0	81793.0	1.372647

- 드디어 소멸 비율을 계산할 수 있게 되었다~~

```
In [9]: pop[ "소멸위기지역" ] = pop[ "소멸비율" ] < 1.0
pop.head()
```

Out[9]:

	구분	20-39세			65세이상			인구수			소멸비율	소멸위기지역
		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	1.226041	False
	고성군	4494.0	2529.0	7023.0	2900.0	4251.0	7151.0	15899.0	14215.0	30114.0	0.707314	True
	동해시	11511.0	9753.0	21264.0	6392.0	8732.0	15124.0	47166.0	46131.0	93297.0	1.289738	False
	삼척시	8708.0	7115.0	15823.0	5892.0	8718.0	14610.0	35253.0	34346.0	69599.0	0.973990	True
	속초시	9956.0	8752.0	18708.0	5139.0	7613.0	12752.0	40288.0	41505.0	81793.0	1.372647	False

- 소멸 위기 지역인지도 체크해 둘 수 있다 ~~~

- 소멸위기지역을 바로 조회해볼수 있다. 대부분 “군” 지역

```
In [10]: pop[pop[ "소멸위기지역" ] == True].index.get_level_values(1)
```

```
Out[10]: 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', '옥천군'],  
    dtype='object', name='시도')
```

```
In [11]: pop.reset_index(inplace=True)  
pop.head()
```

```
Out[11]:
```

구분	광역시도	시도	20-39세			65세이상			인구수			소멸비율	소멸위기지역
			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
3	강원도	삼척시	8708.0	7115.0	15823.0	5892.0	8718.0	14610.0	35253.0	34346.0	69599.0	0.973990	True
4	강원도	속초시	9956.0	8752.0	18708.0	5139.0	7613.0	12752.0	40288.0	41505.0	81793.0	1.372647	False

```
In [12]: tmp_coloumns = [
    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.columns = tmp_coloumns
pop.head()
```

Out[12]:

	광역시도	시도	20-39세남자	20-39세여자	20-39세합계	65세이상남자	65세이상여자	65세이상합계	인구수남자	인구수여자	인구수합계	소멸비율	소멸위기지역
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
3	강원도	삼척시	8708.0	7115.0	15823.0	5892.0	8718.0	14610.0	35253.0	34346.0	69599.0	0.973990	True
4	강원도	속초시	9956.0	8752.0	18708.0	5139.0	7613.0	12752.0	40288.0	41505.0	81793.0	1.372647	False

지도 시각화를 위한 지역별 ID 만들기

```
In [13]: pop.info()
```

```
<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
```

```
In [14]: pop[ "시도" ].unique()
```

```
Out[14]: array(['강릉시', '고성군', '동해시', '삼척시', '속초시', '양구군', '양양군', '영월군', '원주시', '인제군', '정선군', '철원군', '춘천시', '태백시', '평창군', '홍천군', '화천군', '횡성군', '가평군', '고양시', '과천시', '광명시', '광주시', '구리시', '군포시', '권선구', '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', '장안구', '중원구', '처인구', '파주시', '팔달구', '평택시', '포천시', '하남시', '화성시', '거제시', '거창군', '김해시', '남해군', '마산합포구', '마산회원구', '밀양시', '사천시', '산청군', '성산구', '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', '칠곡군', '포항시', '광산구', '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', '종량구', '세종특별자치시', '을주군', '강화군', '계양구', '남동구', '부평구', '연수구', '옹진군', '강진군', '고흥군', '곡성군', '광양시', '구례군', '나주시', '답양군', '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', '청원구', '청주시', '충주시', '홍덕구'], dtype=object)
```

- 그런데... ID를 어떻게 만들지, 서울시도 아니고 전국규모인데ㅠㅠ.

만들어야 할 ID는 아래와 같이 정한다

- 서울 중구
- 서울 서초
- 통영
- 남양주
- 포항 북구
- 인천 남동
- 안양 만안
- 안양 동안
- 안산 단원

대한민국의 구 목록

위키백과, 우리 모두의 백과사전.

다음 표는 대한민국의 구(자치구, 일반구)를 구 설치 순으로 배열한 것이다. 면적은 2011년 1월 1일 기준, 인구는 2011년 7월 주민등록인구통계 기준.

- 설치일 : 최초 구로 설치된 일자
- — : 폐지된 구
- 구의 등급
 - 자치구 - 일반구

설치일	구명	관계 법령	면적 (km ²)	인구 (명)	인구밀도 (명/km ²)	현재 구등급
1943년 6월 10일	서울 중구	조선총독부령 제163호 (1943년 6월 9일)	9.96	132,224	13,276	자치구
1943년 6월 10일	서울 종로구	조선총독부령 제163호 (1943년 6월 9일)	23.91	169,217	7,077	자치구
1943년 6월 10일	서울 동대문구	조선총독부령 제163호 (1943년 6월 9일)	14.20	366,633	25,819	자치구
1943년 6월 10일	서울 성동구	조선총독부령 제163호 (1943년 6월 9일)	16.85	127,748	7,581	자치구
1943년 6월 10일	서울 서대문구	조선총독부령 제163호 (1943년 6월 9일)	17.60	318,467	18,095	자치구
1943년 6월 10일	서울 용산구	조선총독부령 제163호 (1943년 6월 9일)	21.87	247,206	11,303	자치구
1943년 6월 10일	서울 영등포구	조선총독부령 제163호 (1943년 6월 9일)	24.56	403,062	16,411	자치구
1944년 11월 1일	서울 마포구	조선총독부령 제350호 (1944년 10월 23일)	23.87	392,635	16,449	자치구
1949년 8월 13일	서울 성북구	대통령령 제159호 (1949년 8월 13일)	24.57	488,036	19,863	자치구
1957년 1월 1일	부산 중구	법률 제407호 (1956년 12월 17일)	2.82	49,011	17,380	자치구
1957년 1월 1일	부산 서구	법률 제407호 (1956년 12월 17일)	13.88	124,896	8,998	자치구
1957년 1월 1일	부산 영도구	법률 제407호 (1956년 12월 17일)	14.13	144,852	10,251	자치구

In [15]:

```
1 si_name = [None] * len(pop)
2
3 tmp_gu_dict = {
4     "수원": ["장안구", "권선구", "팔달구", "영통구"],
5     "성남": ["수정구", "중원구", "분당구"],
6     "안양": ["만안구", "동안구"],
7     "안산": ["상록구", "단원구"],
8     "고양": ["덕양구", "일산동구", "일산서구"],
9     "용인": ["처인구", "기흥구", "수지구"],
10    "청주": ["상당구", "서원구", "홍덕구", "청원구"],
11    "천안": ["동남구", "서북구"],
12    "전주": ["완산구", "덕진구"],
13    "포항": ["남구", "북구"],
14    "창원": ["의창구", "성산구", "진해구", "마산합포구", "마산회원구"],
15    "부천": ["오정구", "원미구", "소사구"],
16 }
```

- 자치구는 어차피 자료에서 나타나니, 행정구를 가진 지역만 신경쓰면 된다

```
In [16]: 1 pop.head()
```

```
Out[16]:
```

	광역시도	시도	20-39세남자	20-39세여자	20-39세합계	65세이상남자	65세이상여자	65세이상합계	인구수남자	인구수여자	인구수합계	소멸비율	소멸위기지역
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
3	강원도	삼척시	8708.0	7115.0	15823.0	5892.0	8718.0	14610.0	35253.0	34346.0	69599.0	0.973990	True
4	강원도	속초시	9956.0	8752.0	18708.0	5139.0	7613.0	12752.0	40288.0	41505.0	81793.0	1.372647	False

```
In [17]: 1 pop["광역시도"].unique()
```

```
Out[17]: array(['강원도', '경기도', '경상남도', '경상북도', '광주광역시', '대구광역시', '대전광역시', '부산광역시',
   '서울특별시', '세종특별자치시', '울산광역시', '인천광역시', '전라남도', '전라북도', '제주특별자치도',
   '충청남도', '충청북도'], dtype=object)
```

```
In [18]: 1 pop["시도"].unique()
```

```
Out[18]: array(['강릉시', '고성군', '동해시', '삼척시', '속초시', '양구군', '양양군', '영월군', '원주시',  
    '인제군', '정선군', '철원군', '춘천시', '태백시', '평창군', '홍천군', '화천군', '횡성군',  
    '가평군', '고양시', '과천시', '광명시', '광주시', '구리시', '군포시', '권선구', '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', '장안구', '중원구', '처인구',  
    '파주시', '팔달구', '평택시', '포천시', '하남시', '화성시', '거제시', '거창군', '김해시',  
    '남해군', '마산합포구', '마산회원구', '밀양시', '사천시', '산청군', '성산구', '양산시', '의령군',
```

In [19]:

```
1 for idx, row in pop.iterrows():
2     if row["광역시도"][-3:] not in ["광역시", "특별시", "자치시"]:
3         si_name[idx] = row["시도"][:-1]
4
5     elif row["광역시도"] == "세종특별자치시":
6         si_name[idx] = "세종"
7
8     else:
9         if len(row["시도"]) == 2:
10             si_name[idx] = row["광역시도"][:2] + " " + row["시도"]
11         else:
12             si_name[idx] = row["광역시도"][:2] + " " + row["시도"][:-1]
```

- 여기서는 일반 시의 이름과 세종시, 그리고 광역시도의 일반 구를 정리한다

In [19]:

```
1 for idx, row in pop.iterrows():
2     if row["광역시도"][-3:] not in ["광역시", "특별시", "자치시"]:
3         si_name[idx] = row["시도"][:-1]
4
5     elif row["광역시도"] == "세종특별자치시":
6         si_name[idx] = "세종"
7
8     else:
9         if len(row["시도"]) == 2:
10             si_name[idx] = row["광역시도"][:2] + " " + row["시도"]
11         else:
12             si_name[idx] = row["광역시도"][:2] + " " + row["시도"][:-1]
```

- 강릉시 => 강릉
- 춘천시 => 춘천 … 이런 식으로 정리한다

In [19]:

```
1 for idx, row in pop.iterrows():
2     if row["광역시도"][-3:] not in ["광역시", "특별시", "자치시"]:
3         si_name[idx] = row["시도"][:-1]
4
5     elif row["광역시도"] == "세종특별자치시":
6         si_name[idx] = "세종"
7
8     else:
9         if len(row["시도"]) == 2:
10             si_name[idx] = row["광역시도"][:2] + " " + row["시도"]
11         else:
12             si_name[idx] = row["광역시도"][:2] + " " + row["시도"][:-1]
```

- 세종시는 특별히 관리하고~~~

In [19]:

```
1 for idx, row in pop.iterrows():
2     if row["광역시도"][-3:] not in ["광역시", "특별시", "자치시"]:
3         si_name[idx] = row["시도"][:-1]
4
5     elif row["광역시도"] == "세종특별자치시":
6         si_name[idx] = "세종"
7
8     else:
9         if len(row["시도"]) == 2:
10             si_name[idx] = row["광역시도"][:2] + " " + row["시도"]
11         else:
12             si_name[idx] = row["광역시도"][:2] + " " + row["시도"][:-1]
```

- 이 부분은
- 중구 => 중구
- 강남구 => 강남과 같이 정리한다

In [20]:

```
1 for idx, row in pop.iterrows():
2     if row["광역시도"][-3:] not in ["광역시", "특별시", "자치시"]:
3         for keys, values in tmp_gu_dict.items():
4             if row["시도"] in values:
5                 if len(row["시도"]) == 2:
6                     si_name[idx] = keys + " " + row["시도"]
7                 elif row["시도"] in ["마산합포구", "마산회원구"]:
8                     si_name[idx] = keys + " " + row["시도"][2:-1]
9                 else:
10                     si_name[idx] = keys + " " + row["시도"][:-1]
```

- 행정구에 대해 특별히 다시 계산한다

In [20]:

```
1 for idx, row in pop.iterrows():
2     if row["광역시도"][-3:] not in ["광역시", "특별시", "자치시"]:
3         for keys, values in tmp_gu_dict.items():
4             if row["시도"] in values:
5                 if len(row["시도"]) == 2:
6                     si_name[idx] = keys + " " + row["시도"]
7                 elif row["시도"] in ["마산합포구", "마산회원구"]:
8                     si_name[idx] = keys + " " + row["시도"][2:-1]
9                 else:
10                     si_name[idx] = keys + " " + row["시도"][:-1]
```

- 광역시나 특별시, 자치시가 아닌 경우의 행정구에 대해서만 적용한다

In [20]:

```
1 for idx, row in pop.iterrows():
2     if row["광역시도"][-3:] not in ["광역시", "특별시", "자치시"]:
3         for keys, values in tmp_gu_dict.items():
4             if row["시도"] in values:
5                 if len(row["시도"]) == 2:
6                     si_name[idx] = keys + " " + row["시도"]
7                 elif row["시도"] in ["마산합포구", "마산회원구"]:
8                     si_name[idx] = keys + " " + row["시도"][2:-1]
9                 else:
10                     si_name[idx] = keys + " " + row["시도"][:-1]
```

- 행정구를 지정한 dict형 자료에 있는 지역인지 검색하고

In [20]:

```
1 for idx, row in pop.iterrows():
2     if row["광역시도"][-3:] not in ["광역시", "특별시", "자치시"]:
3         for keys, values in tmp_gu_dict.items():
4             if row["시도"] in values:
5                 if len(row["시도"]) == 2:
6                     si_name[idx] = keys + " " + row["시도"]
7                 elif row["시도"] in ["마산합포구", "마산회원구"]:
8                     si_name[idx] = keys + " " + row["시도"][2:-1]
9                 else:
10                     si_name[idx] = keys + " " + row["시도"][:-1]
```

- 분당구 => 분당, 북구=> 북구
- 특별히 너무 긴 이름의 구는 짧게 처리한다

In [21]:

```
1 for idx, row in pop.iterrows():
2     if row["광역시도"][-3:] not in ["광역시", "특별시", "자치시"]:
3         if row["시도"][:-1] == "고성" and row["광역시도"] == "강원도":
4             si_name[idx] = "고성(강원)"
5         elif row["시도"][:-1] == "고성" and row["광역시도"] == "경상남도":
6             si_name[idx] = "고성(경남)"
```

- 특별히 고성군에 대해 한 번 더 신경써준다

In [22]: 1 si_name

Out[22]: ['강릉',
'고성(강원)',
'동해',
'삼척',
'속초',
'양구',
'양양',
'영월',
'원주',
'인제',
'정선',
'철원',

```
In [23]: 1 | pop["ID"] = si_name
```

```
In [24]: 1 del pop["20-39세남자"]
2 del pop["65세이상남자"]
3 del pop["65세이상여자"]
4
5 pop.head()
```

Out[24]:

	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
3	강원도	삼척시	7115.0	15823.0	14610.0	35253.0	34346.0	69599.0	0.973990	True	삼척
4	강원도	속초시	8752.0	18708.0	12752.0	40288.0	41505.0	81793.0	1.372647	False	속초

- 인구 소멸 위기 지역 파악도 끝냈고,
- 지도에 그리기 위한 ID도 다 생성했다

카르토그램에 대해 고민하기



앞서 Carto-gram 처럼 그리는 원본 코드는 혜식님의 코드라고 했다
문제는 혜식님의 코드는 결과만 있어서 그 시작 원리를 알 수 없다

더 큰 문제는 우리는 그 코드를 그대로 사용할 수도 없다😢

지역적 이슈도 있고, 원하는 형태가 살짝 바뀌었다

그럼…

역 추적해야지~~~ 고민 고민…

```
In [74]: def draw_blockcolormap(tbl, datacol, vmin, vmax, whitelabelmin, cmapname, gamma, datalabel, dataticks):
    cmap = colors.LinearSegmentedColormap(cmapname + 'custom',
                                           getattr(_cm, '_%s_data' % format(cmapname)), gamma=gamma)
    cmap.set_bad('white', 1.)

    mapdata = tbl.pivot(index='y', columns='x', values=datacol)
    masked_mapdata = np.ma.masked_where(np.isnan(mapdata), mapdata)

    plt.figure(figsize=(9, 16))
    plt.pcolor(masked_mapdata, vmin=vmin, vmax=vmax, cmap=cmap,
               edgecolor='#aaaaaa', linewidth=0.5)

    # 지역 이름 표시
    for idx, row in tbl.iterrows():
        annocolor = 'white' if row[datacol] > whitelabelmin else 'black'
        dispname = 'white' if row['d1'].endswith('시') and not row['d1'].startswith('세종'):

            if row['d1'].endswith('시') and not row['d1'].startswith('세종'):
                dispname = '%s %s'.format(row['d1'][:-1], row['d2'][-1])
            if len(row['d2']) <= 2:
                dispname += row['d2'][-1]
        else:
            dispname = row['d2'][:-1]

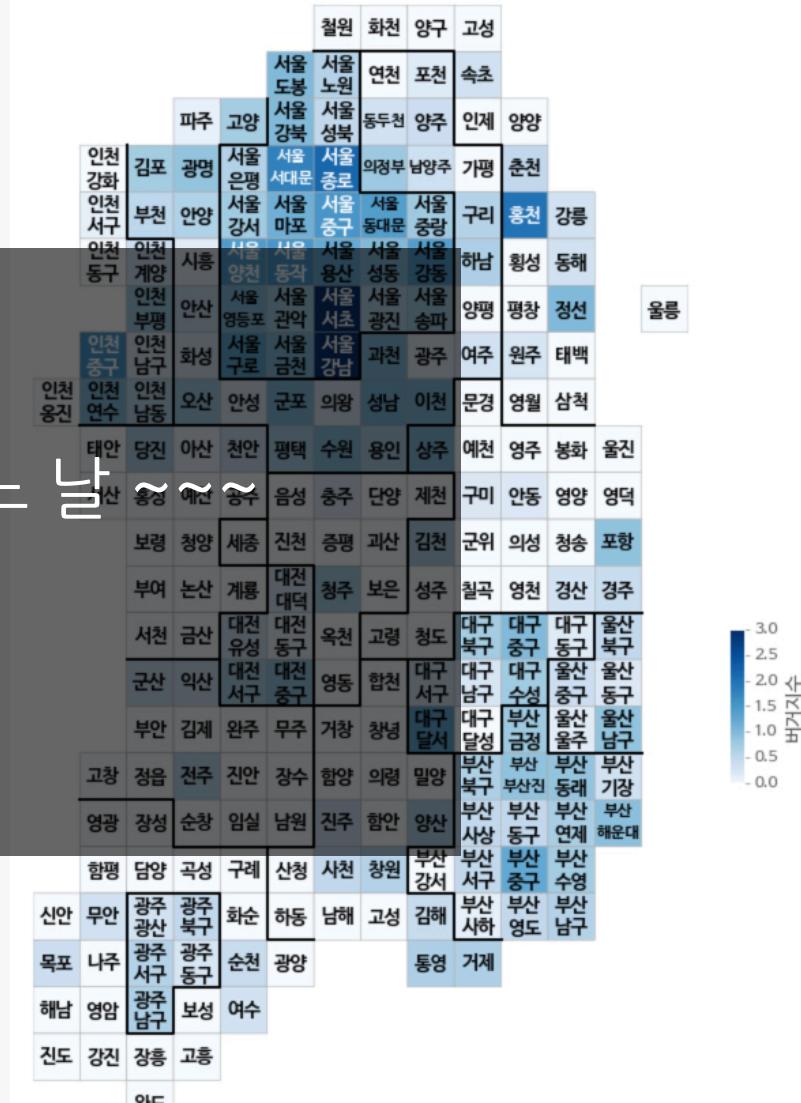
        # 서대문구, 서귀포시 같이 이름이 3자 이상인 경우에 작은 글자로 표시합니다.
        if len(dispname.splitlines())[-1] >= 3:
            fontsize, linespacing = 12, 1.2
        else:
            fontsize, linespacing = 14, 1.05

        plt.annotate(dispname, (row['x']+0.5, row['y']+0.5), weight='bold',
                     fontsize=fontsize, ha='center', va='center', color=annocolor,
                     linespacing=linespacing)

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

    plt.gca().invert_yaxis()
    plt.gca().set_aspect(1)

    plt.axis('off')
```



0	1	2	3	4	5	6	7	8	9	10	11	12	13
							철원	화천	양구	고성 (강원)			
			양주	동두천	연천	포천	의정부	인제	춘천	속초			
			고양	고양	서울	서울	남양주	홍천	횡성	양양			
			더양	일산동	도봉	노원							
			고양	김포	서울	서울							
			일산서		강북	성북							
			부천	안양	서울	서울							
			만안	광명	종로	서울							
			인천	부천	서울	마포							
			강화	안양	운평	중구							
			인천	동안	서울	서울							
			인천	시흥	강서	동작							
			인천	오정	서울	서울							
			인천	계양	상록	양천							
			인천	부평	안산	판악							
			인천	단원	서울	서초							
			인천	영등포	금천	서초							
			인천	화성	서울	강남							
			인천	구로	군포	군포							
			인천	중구	의왕	의왕							
			인천	인천	인천	인천							
			인천	남구	오산	안성							
			인천	남동	천안	평택							
			태안	아산	천안	음성							
			태안	동남	서북	수원							
			당진	홍성	예산	공주	진천	충주	청주	괴산	칠곡	영천	경산
			세종	세종	세종	세종	대전	대전	청주	청주	보은	고령	청도
			보령	청양	세종	세종	대전	대전	청주	청주	보은	고령	청주
			부여	논산	계룡	계룡	대전	대전	청주	청주	대구	대구	성주
			서천	금산	금산	금산	동구	동구	상당	상당	서원	서원	서구
			군산	익산	익산	익산	대전	대전	옥천	옥천	영동	영동	영동
			부안	김제	김제	김제	부안	부안	창녕	창녕	창녕	밀양	밀양
			고창	정읍	정읍	정읍	전주	전주	전주	전주	전주	전주	전주
			영광	장성	장성	장성	전주	전주	전주	전주	전주	전주	전주
			함평	담양	담양	담양	전주	전주	전주	전주	전주	전주	전주
			신안	무안	광주	광주	곡성	화순	화순	화순	화순	화순	화순
			목포	나주	광주	광주	광주	광주	광주	광주	광주	광주	광주
			해남	영암	영암	영암	영암	영암	영암	영암	영암	영암	영암
			진도	강진	강진	강진	보성						
							완도						
								제주					
									서귀포				

일단 먼저,

엑셀로 그리자 ~

```
In [25]: 1 draw_korea_raw = pd.read_excel("../data/07_draw_korea_raw.xlsx")
2 draw_korea_raw
```

Out[25]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	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 일산 동	서울 도봉	서울 노원	남양주	홍천	횡성	양양	NaN	NaN	NaN
3	NaN	NaN	파주	고양 일산 서	김포	서울 강북	서울 성북	가평	구리	하남	정선	강릉	NaN	NaN

- 엑셀에서 그린 지도 모양을 읽어온다

07. Population

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	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
3	NaN	NaN	파주	고양 일산서	김포	서울 강북	서울 성북	가평	구리	하남	정선	강릉	NaN	NaN
4	NaN	NaN	부천 소사	안양 만안	광명	서울 서대문	서울 종로	서울 동대문	서울 중랑	양평	태백	동해	NaN	NaN
5	NaN	인천 강화	부천 원미	안양 동안	서울 은평	서울 마포	서울 종구	서울 성동	서울 강동	여주	원주	삼척	NaN	NaN
6	NaN	인천 서구	부천 오정	시흥	서울 강서	서울 동작	서울 용산	서울 광진	서울 송파	이천	평창	울진	NaN	NaN
7	NaN	인천 동구	인천 계양	안산 상록	서울 양천	서울 관악	서울 서초	성남 중원	과천	광주	영월	영덕	NaN	NaN
8	NaN	NaN	인천 부평	안산 단원	서울 영등포	서울 금천	서울 강남	성남 분당	성남 수정	용인 수지	문경	봉화	NaN	울릉
9	NaN	인천 중구	인천 남구	화성	서울 구로	군포	의왕	수원 영통	용인 기흥	용인 처인	안동	영양	NaN	NaN
10	인천 옹진	인천 연수	인천 남동	오산	안성	수원 권선	수원 장안	제천	예천	영주	구미	청송	포항 북구	NaN
11	태안	아산	천안 동남	천안 서북	평택	음성	수원 팔달	단양	상주	김천	군위	의성	포항 남구	NaN
12	NaN	당진	홍성	예산	공주	진천	충주	청주 흥덕	괴산	칠곡	영천	경산	경주	NaN
13	NaN	서산	보령	청양	세종	대전 대덕	증평	청주 청원	보은	고령	청도	성주	울산 북구	NaN
14	NaN	NaN	부여	논산	계룡	대전 동구	청주 상당	청주 서원	대구 북구	대구 중구	대구 수성	울산 울주	울산 동구	NaN
15	NaN	NaN	서천	금산	대전 유성	대전 중구	옥천	영동	대구 서구	대구 남구	대구 동구	울산 중구	울산 남구	NaN
16	NaN	NaN	군산	익산	대전 서구	무주	거창	합천	대구 달서	대구 달성	부산 금정	부산 동래	부산 기장	NaN
17	NaN	NaN	부안	김제	완주	장수	함양	창녕	밀양	부산 북구	부산 부산진	부산 연제	부산 해운대	NaN
18	NaN	고창	정읍	전주 덕진	진안	남원	진주	의령	부산 강서	부산 사상	부산 동구	부산 중구	NaN	NaN
19	NaN	영광	장성	전주 완산	임실	산청	함안	양산	창원 합포	부산 서구	부산 사하	부산 남구	NaN	NaN
20	NaN	함평	담양	순창	구례	하동	창원 의창	창원 성산	창원 진해	김해	부산 영도	부산 수영	NaN	NaN
21	신안	무안	광주 광산	곡성	화순	광양	사천	창원 회원	통영	NaN	NaN	NaN	NaN	NaN
22	목포	나주	광주 서구	광주 북구	순천	고흥	남해	고성(경남)	거제	NaN	NaN	NaN	NaN	NaN
23	해남	영암	광주 남구	광주 동구	여수	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
24	진도	강진	장흥	보성	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
25	NaN	NaN	완도	NaN	NaN	제주	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
26	NaN	NaN	NaN	NaN	NaN	서귀포	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

zero-base /

```
In [26]: 1 draw_korea_raw_stacked = pd.DataFrame(draw_korea_raw.stack())
          2 draw_korea_raw_stacked
```

Out[26]:



0	1	2	3	4	5	6	7	8	9	10	11	12	13
고양	고양	서울	서울	서울	서울	서울	서울	서울	서울	서울	서울	서울	서울
덕양	고양	서울	서울	서울	서울	서울	서울	서울	서울	서울	서울	서울	서울
부천	인양	인산동	도봉	노원	구	강서	강동	동대	중랑	양평	태백	동해	강릉
파주	인양	김포	강북	성북	가평	구리	하남	정선	원주	원주	원주	원주	원주
인천	부천	만안	광명	서대문	종로	서울							
강화	부천	동안	우정	마포	중구	성동	강동	성남	성남	성남	용인	용인	용인
인천	인천	동안	시흥	서울	서울	서울	서울	서울	서울	서울	서울	서울	서울
서구	인천	오정	동자	용산	용산	광진	송파	성남	성남	성남	평창	평창	평창
인천	인천	인천	제양	인천	인천	관악	서초	서초	서초	서초	영월	영월	영월
동구	부평	인천	상록	양천	양천	성남	성남	성남	성남	성남	봉화	봉화	봉화
인천	인천	남구	단원	단원	영등포	금천	강남	분당	수원	수원	문경	문경	문경
중구	화성	화성	구로	의왕	의왕	수원	수원	수원	수원	수원	포항	포항	포항
인천	인천	남동	오산	수원	수원	장	제천	제천	제천	제천	북구	북구	북구
옹진	인천	천안	안성	수원	수원	장	수원	수원	수원	수원	청송	청송	청송
태안	아산	천안	천안	평택	평택	단양	상주	김천	김천	김천	군위	군위	군위
당진	당진	당진	예산	금강	금강	청주	괴산	칠곡	영천	영천	경산	경주	경주
서산	보령	보령	청양	세종	세종	청주	청주	보은	고령	고령	성주	성주	성주
부여	부여	논산	계룡	대전	대덕	청주	청주	대구	대구	대구	대구	대구	대구
동구	동구	동구	동구	동구	동구	대구	대구	대구	대구	대구	울산	울산	울산
서천	서천	금산	금산	대전	대전	청주	청주	대구	대구	대구	대구	대구	대구
중구	중구	유성	유성	대전	대전	대전	대전	대구	대구	대구	대구	대구	대구
군산	군산	익산	익산	무주	거창	합천	대구	대구	대구	대구	부산	부산	부산
부안	부안	김제	완주	장수	장수	장수	장수	밀양	밀양	밀양	부산	부산	부산
고창	고창	정읍	정읍	진안	진안	부산							
영광	영광	영광	영광	전주	전주	부산							
함평	함평	담양	순창	순창	구례	장수	장수	장수	장수	장수	남구	남구	남구
신안	신안	무안	광주	광주	광주	광주	광주	광주	광주	광주	광주	광주	광주
목포	목포	나주	광주	광주	광주	광주	광주	광주	광주	광주	부산	부산	부산
해남	해남	영암	영암	영암	영암	영암	영암	영암	영암	영암	영동	영동	영동
진도	진도	강진	장흥	보성	보성	제주							
					원도								

- 각 지역별 위치가 나타났다

In [27]:

```
1 draw_korea_raw_stacked.reset_index(inplace=True)
2 draw_korea_raw_stacked
```

Out[27]:

	level_0	level_1	0
0	0	7	철원
1	0	8	화천
2	0	9	양구
3	0	10	고성(강원)

- 인덱스로 나타난 좌표를 데이터로 사용하기 위해 `reset_index`

In [28]:

```
1 draw_korea_raw_stacked.rename(  
2     columns={"level_0": "y", "level_1": "x", 0: "ID"}, inplace=True  
3 )  
4 draw_korea_raw_stacked
```

Out[28]:

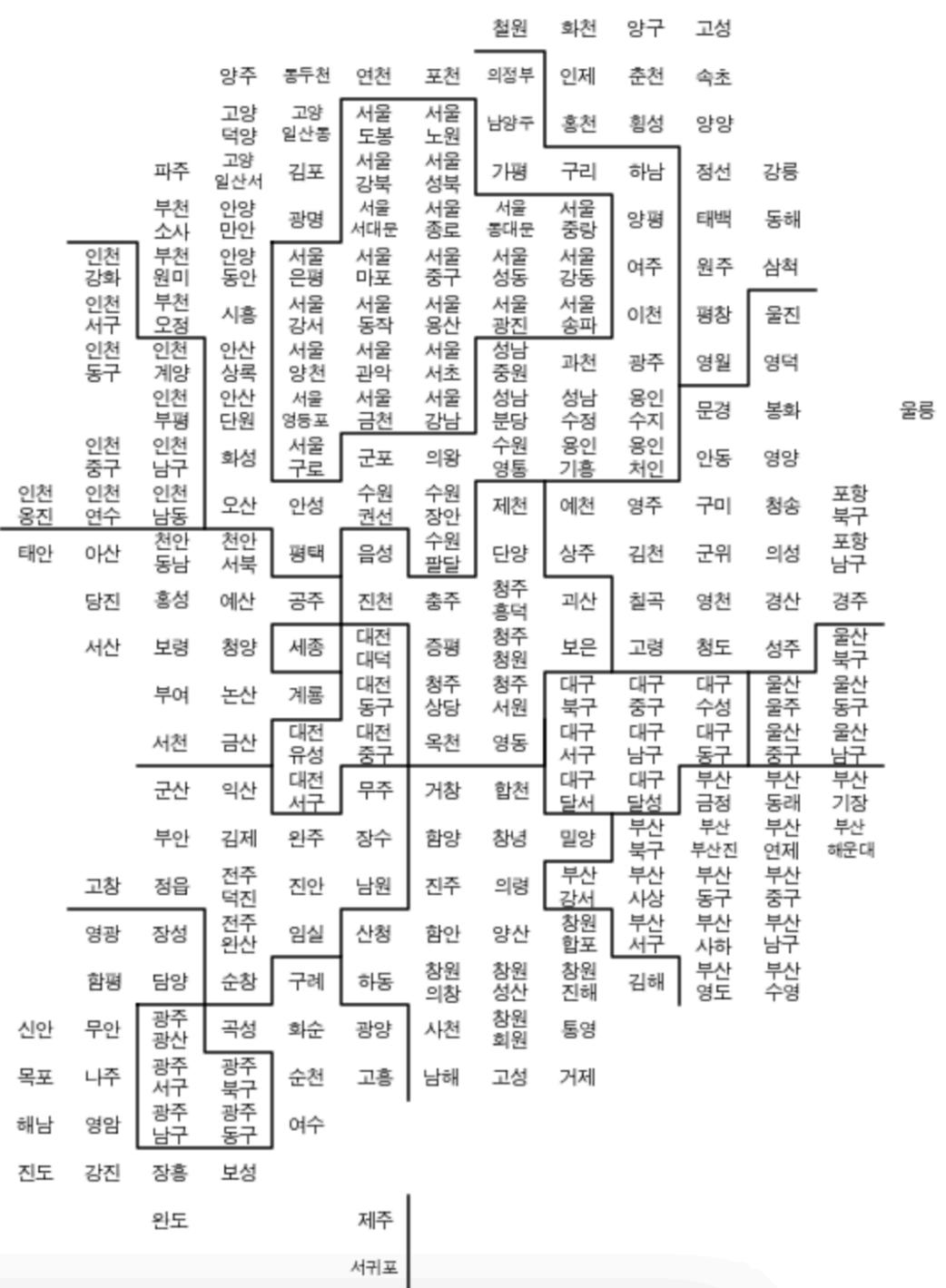
	y	x	ID
0	0	7	철원
1	0	8	화천
2	0	9	양구

- 이름도 바꿔주고~😊

```
In [29]: 1 draw_korea = draw_korea_raw_stacked
```

- 변수 이름이 마음에 안들어서~~😊

- 이제 이런 그림을 그릴건데,



```
In [30]: BORDER_LINES = [
    [(5, 1), (5, 2), (7, 2), (7, 3), (11, 3), (11, 0)], # 인천
    [(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)], # 서울
    [(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)], # 경기도
    [(8, 10), (8, 11), (6, 11), (6, 12)], # 강원도
    [(12, 5),(13, 5),(13, 4),(14, 4),(14, 5),(15, 5),(15, 4),(16, 4),(16, 2)], # 충청북도
    [(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)], # 전라북도
    [(13, 5), (13, 6), (16, 6)],
    [(13, 5), (14, 5)], # 대전시 #세종시
    [(21, 2), (21, 3), (22, 3), (22, 4), (24, 4), (24, 2), (21, 2)], # 광주
    [(20, 5), (21, 5), (21, 6), (23, 6)], # 전라남도
    [(10, 8), (12, 8), (12, 9), (14, 9), (14, 8), (16, 8), (16, 6)], # 충청북도
    [(14, 9), (14, 11), (14, 12), (13, 12), (13, 13)], # 경상북도
    [(15, 8), (17, 8), (17, 10), (16, 10), (16, 11), (14, 11)], # 대구
    [(17, 9), (18, 9), (18, 8), (19, 8), (19, 9), (20, 9), (20, 10), (21, 10)], # 부산
    [(16, 11), (16, 13)],
    [(27, 5), (27, 6), (25, 6)],
]
```

- 먼저 경계선은 한땀한땀 손으로.ㅠㅠ.

```
In [31]: def plot_text_simple(draw_korea):
    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"]

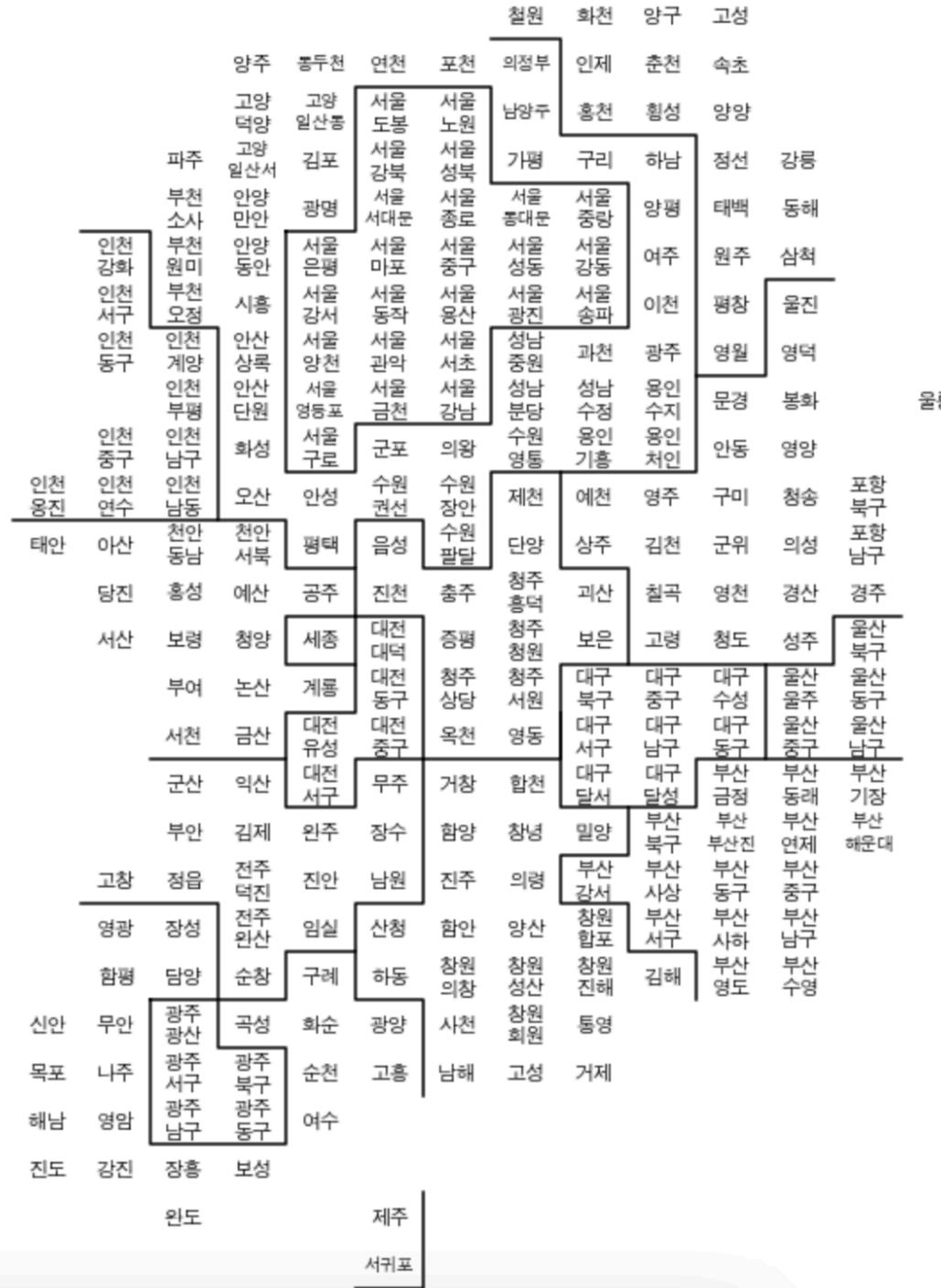
        if len(dispname.splitlines())[-1]) >= 3:
            fontsize, linespacing = 9.5, 1.5
        else:
            fontsize, linespacing = 11, 1.2

        plt.annotate(
            dispname,
            (row["x"] + 0.5, row["y"] + 0.5),
            weight="bold",
            fontsize=fontsize,
            ha="center",
            va="center",
            linespacing=linespacing,
        )
```

- 시도의 이름을 표현하는 함수

- 간단히 경계선과 시도 이름만 먼저 도전

```
In [32]: def simpleDraw(draw_korea):  
    plt.figure(figsize=(8, 11))  
  
    plot_text_simple(draw_korea)  
  
    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()  
  
simpleDraw(draw_korea)
```



```
In [34]: set(draw_korea[ "ID" ].unique()) - set(pop[ "ID" ].unique())  
Out[34]: set()
```

- 이 간단한 코드가 하고 싶었던 일은 뭘까?

```
In [35]: set(pop[ "ID" ].unique()) - set(draw_korea[ "ID" ].unique())
```

```
Out[35]: {'고양', '부천', '성남', '수원', '안산', '안양', '용인', '전주', '창원', '천안', '청주', '포항'}
```

- 근데 이 결과는 왜 또 다르지?
- 차집합은 교환법칙이 성립하지 않는다.
- 지금 나열된 데이터들은 어떻게 하지?

```
In [36]: 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)

print(set(pop["ID"].unique()) - set(draw_korea["ID"].unique()))

set()
```

- 이 경우는 그냥 지우면 된다. 왜?

```
In [38]: pop = pd.merge(pop, draw_korea, how="left", on=[ "ID" ])  
pop.head()
```

Out[38]:

	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

- 그리고, 지도를 그리기 위한 데이터와 인구현황 데이터를 합치자

```
In [38]: def get_data_info(targetData, blockedMap):
    whitelabelmin = (
        max(blockedMap[targetData]) - min(blockedMap[targetData])
    ) * 0.25 + min(blockedMap[targetData])
    vmin = min(blockedMap[targetData])
    vmax = max(blockedMap[targetData])

    mapdata = blockedMap.pivot_table(index="y", columns="x", values=targetData)

    return mapdata, vmax, vmin, whitelabelmin
```

- 그림을 그리기 위한 데이터를 계산하는 함수
- 이 함수는 색상을 만들때 최소값을 흰색으로 한다.
- blockedMap은 인구현황이고, targetData는 그리고 싶은 컬럼

```
In [39]: def get_data_info_for_zero_center(targetData, blockedMap):
    whitelabelmin = 5
    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)

    return mapdata, vmax, vmin, whitelabelmin
```

- 그림을 그리기 위한 데이터를 계산하는 함수
- 이 함수는 색상을 만들때 중간값을 흰색으로 한다.
- blockedMap은 인구현황이고, targetData는 그리고 싶은 컬럼

```
In [40]: def plot_text(targetData, blockedMap, whitelabelmin):
    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 = 9.5, 1.5
        else:
            fontsize, linespacing = 11, 1.2

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

```
In [41]: def drawKorea(targetData, blockedMap, cmapname, zeroCenter=False):
    if zeroCenter:
        masked_mapdata, vmax, vmin, whitelabelmin = get_data_info_for_zero_center(
            targetData, blockedMap
        )
    if not zeroCenter:
        masked_mapdata, vmax, vmin, whitelabelmin = get_data_info(
            targetData, blockedMap
        )

    plt.figure(figsize=(9, 11))
    plt.pcolor(
        masked_mapdata, vmin=vmin, vmax=vmax, cmap=cmapname, edgecolor="#aaaaaa",
        linewidth=0.5,
    )

    plot_text(targetData, blockedMap, whitelabelmin)

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

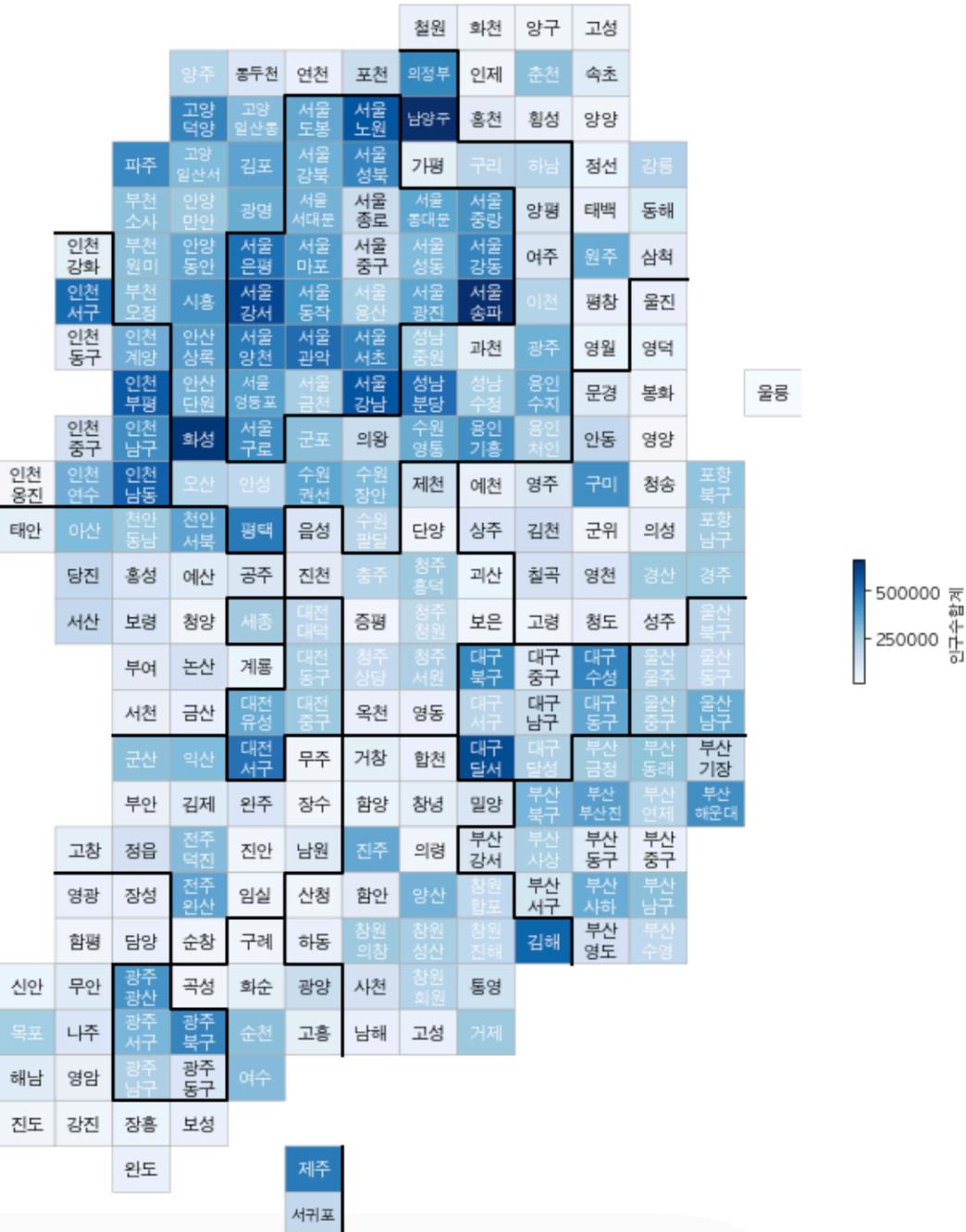
    plt.gca().invert_yaxis()
    plt.axis("off")
    cb = plt.colorbar(shrink=0.1, aspect=10)
    cb.set_label(targetData)
    plt.tight_layout()
    plt.show()
```

07.

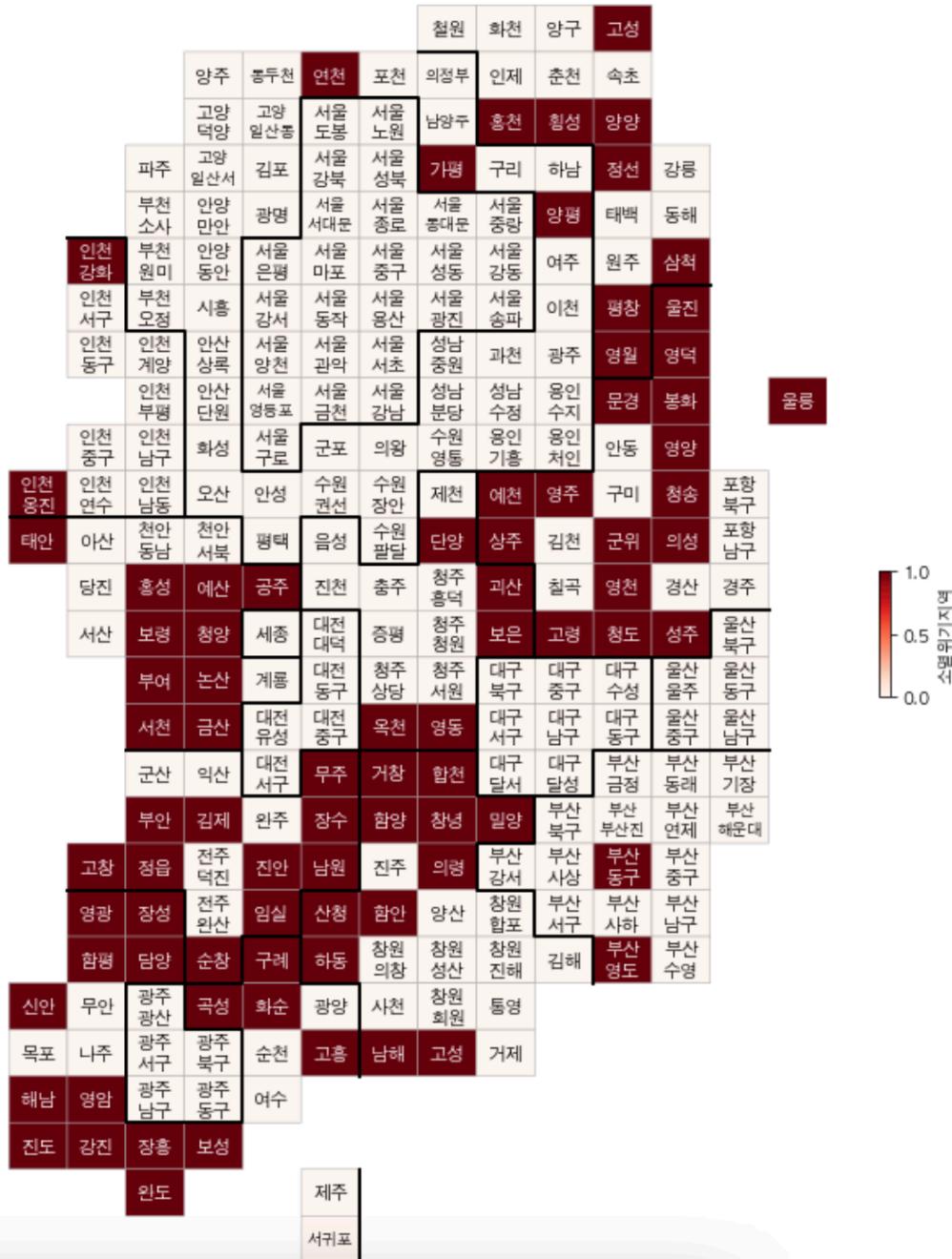
Population

```
In [42]: drawKorea("인구수합계", pop, "Blues")
```

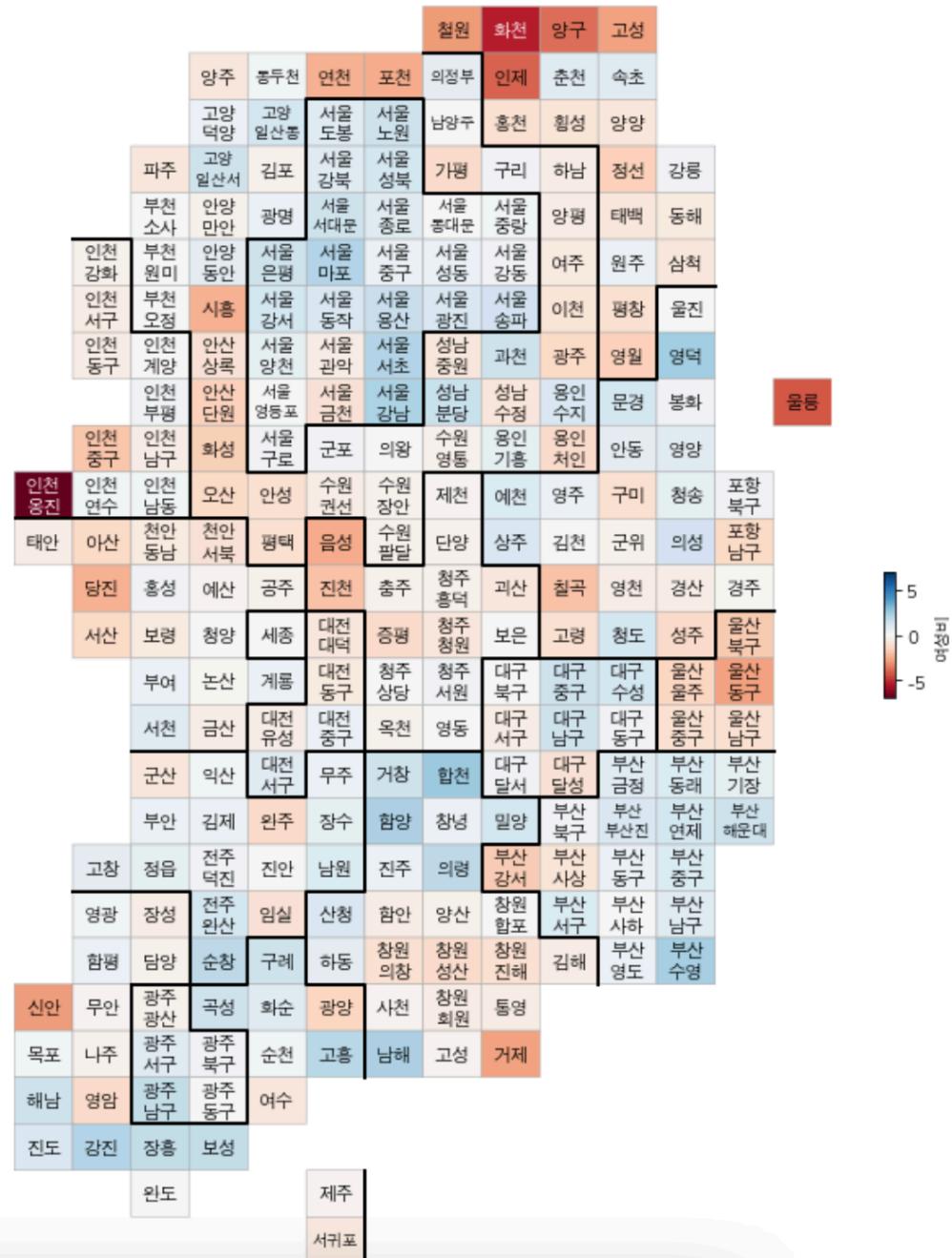
zero-base /



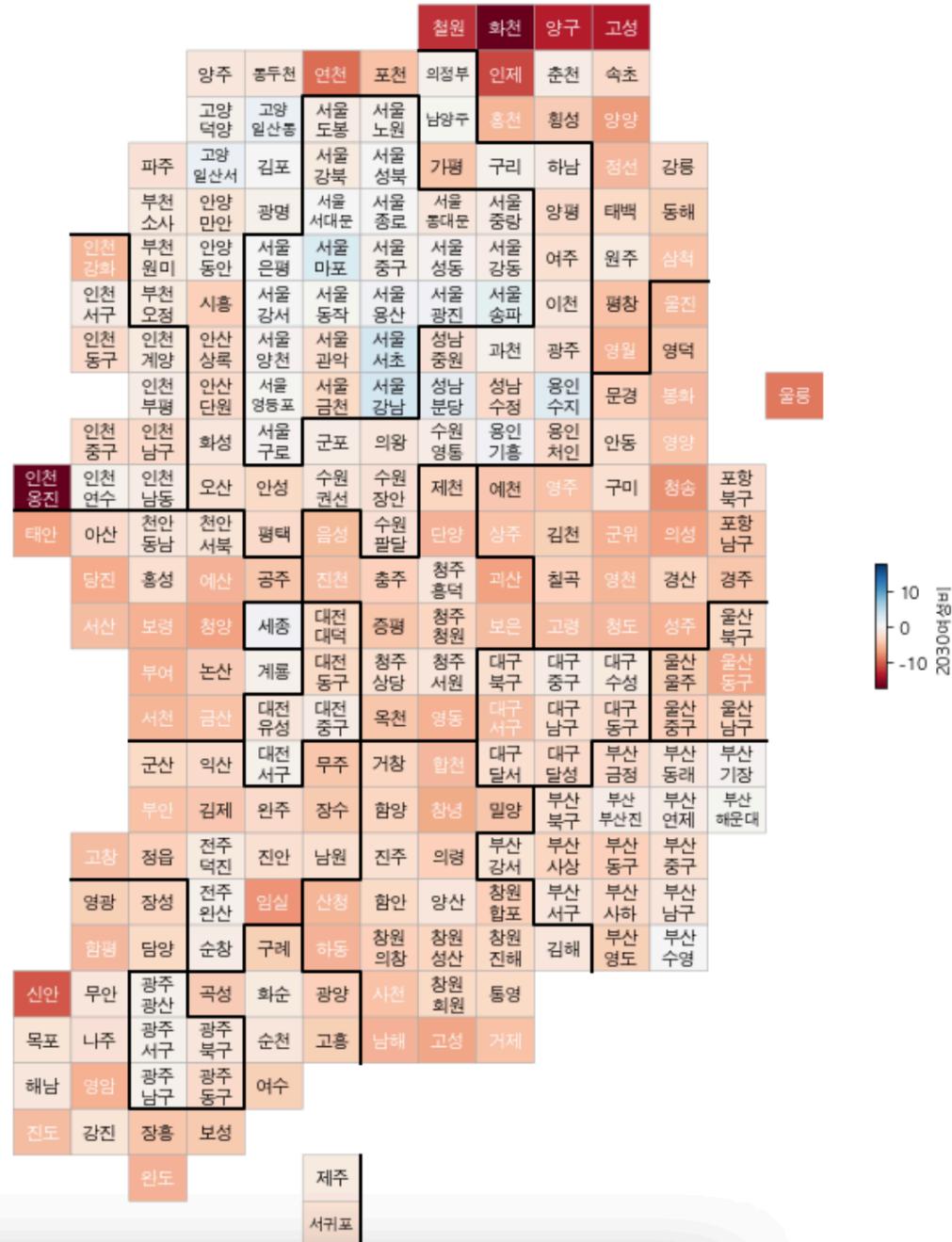
```
In [43]: pop[ "소멸위기지역" ] = [ 1 if con else 0 for con in pop[ "소멸위기지역" ] ]
drawKorea("소멸위기지역", pop, "Reds")
```



```
In [45]: pop["여성비"] = (pop["인구수여자"] / pop["인구수합계"] - 0.5) * 100
drawKorea("여성비", pop, "RdBu", zeroCenter=True)
```



```
In [46]: pop["2030여성비"] = (pop["20-39세여자"] / pop["20-39세합계"] - 0.5) * 100
drawKorea("2030여성비", pop, "RdBu", zeroCenter=True)
```



```
In [47]: import folium
import json

pop_folium = pop.set_index("ID")
pop_folium.head()
```

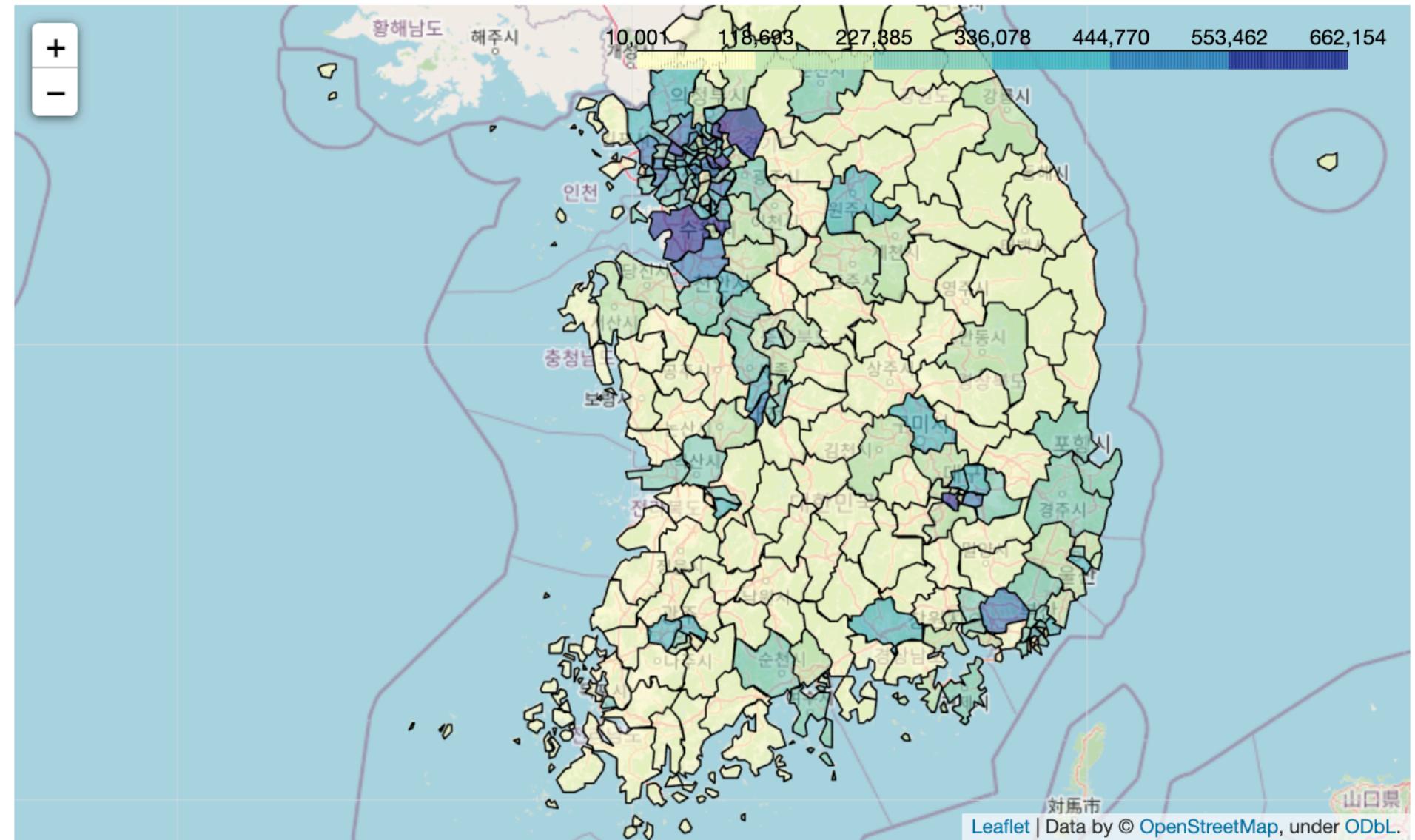
Out[47]:

광역 시도	시 도	20-39세 여자	20-39세 합계	65세이상 합계	인구수남 자	인구수여 자	인구수합 계	소멸비율	소멸위 기지역	y	x	여성비	2030여성비
ID													
강릉	강원 도	강 릉 시	23098.0	49384.0	37679.0	106231.0	107615.0	213846.0	1.226041	0	3	11	0.323597 -3.227766
고성 (강원)	강원 도	고 성 군	2529.0	7023.0	7151.0	15899.0	14215.0	30114.0	0.707314	1	0	10	-2.796042 -13.989748
동해	강원 도	동 해 시	9753.0	21264.0	15124.0	47166.0	46131.0	93297.0	1.289738	0	4	11	-0.554680 -4.133747
삼척	강원 도	삼 척 시	7115.0	15823.0	14610.0	35253.0	34346.0	69599.0	0.973990	1	5	11	-0.651590 -5.033812
속초	강원 도	속 초 시	8752.0	18708.0	12752.0	40288.0	41505.0	81793.0	1.372647	0	1	10	0.743951 -3.217875

```
In [48]: geo_path = ".../data/07_skorea_municipalities_geo_simple.json"
geo_str = json.load(open(geo_path, encoding="utf-8"))

mymap = folium.Map(location=[36.2002, 127.054], zoom_start=7)
mymap.choropleth(
    geo_data=geo_str,
    data=pop_folium[ "인구수합계" ],
    columns=[pop_folium.index, pop_folium[ "인구수합계" ]],
    fill_color="YlGnBu", # PuRd, YlGnBu
    key_on="feature.id",
)

mymap
```



```
In [49]: mymap = folium.Map(location=[36.2002, 127.054], zoom_start=7)
mymap.choropleth(
    geo_data=geo_str,
    data=pop_folium["소멸위기지역"],
    columns=[pop_folium.index, pop_folium["소멸위기지역"]],
    fill_color="PuRd", # PuRd, YlGnBu
    key_on="feature.id",
)

mymap
```

