

# 지도 시각화



## 지|도 사|각화

- 단계 구분도 : 지역별 통계치를 색깔 차이로 표현한 지도
  - 인구나 소득 같은 통계치가 지역별로 어떻게 다른 지 이해 용이
  - SIG.geojon파일: 행정구역코드, 지역이름, 시군구 경계 위도와 경도 좌표를 담고 있는 GeoJSON파일로 json패키지에서 불러올 수 있음

## 1. 시군구 경계 지도 데이터 준비하기

```
In [4]: 1 import json
        2 geo = json.load(open('SIG.geojson', encoding = 'UTF-8'))
```

```
In [5]: 1 # 행정 구역 코드 출력
        2 geo['features'][0]['properties']
```

```
Out[5]: {'SIG_CD': '42110', 'SIG_ENG_NM': 'Chuncheon-si', 'SIG_KOR_NM': '춘천시'}
```

```
In [6]: 1 # 위도, 경도 좌표 출력
        2 geo['features'][0]['geometry']
```

```
Out[6]: {'type': 'MultiPolygon',  
         'coordinates': [[[[[127.58508551154958, 38.08062321552708],  
                             [127.58565575732702, 38.0802009066172],  
                             [127.58777905808203, 38.080354190085544],  
                             [127.58890487394689, 38.080881783588694],  
                             [127.59031267326897, 38.080596307998306],  
                             [127.59061778023133, 38.08053451807929],  
                             [127.59137292963024, 38.080476965118685],  
                             [127.59349548967889, 38.08031540227777],  
                             [127.594304166838, 38.080225164594665],  
                             [127.59529626584073, 38.08011430122607],  
                             [127.59683797157537, 38.079896764445195],  
                             [127.59703599139921, 38.07914773681181],  
                             [127.59756973764837, 38.07761547763786],  
                             [127.59794006383213, 38.07714541130494],  
                             [127.59816760846913, 38.07685681587356],  
                             [127.59847648881038, 38.076670284099194],  
                             [127.59956152109447, 38.076014995491754],  
                             [127.60022591131202, 38.07568645192285],  
                             [127.60004350173000, 38.075600100500400]
```

# 지도 시각화

- 시군구별 인구 데이터 준비
  - 국가통계포털 KOSIS의 '주민등록인구현황' 자료 중 일부를 추출하여 Population\_SIG.csv를 이용
  - 파일 위치는 현재 작업중인 jupyter폴더와 동일하도록 하는 것이 편리함
  - 자료구조와 숫자를 문자로 변환하여 지도의 내용으로 적용하도록 타입변환 적용함

In [7]:

```
1 import pandas as pd
2 df_pop = pd.read_csv('Population_SIG.csv')
3 df_pop.head()
```

Out[7]:

	code	region	pop
0	11	서울특별시	9509458
1	11110	종로구	144683
2	11140	중구	122499
3	11170	용산구	222953
4	11200	성동구	285990

In [8]:

```
1 df_pop.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 278 entries, 0 to 277
Data columns (total 3 columns):
#   Column  Non-Null Count  Dtype
---  -
0   code    278 non-null     int64
1   region  278 non-null     object
2   pop     278 non-null     int64
dtypes: int64(2), object(1)
memory usage: 6.6+ KB
```

In [9]:

```
1 df_pop['code'] = df_pop['code'].astype(str)
```

# 지도 시각화

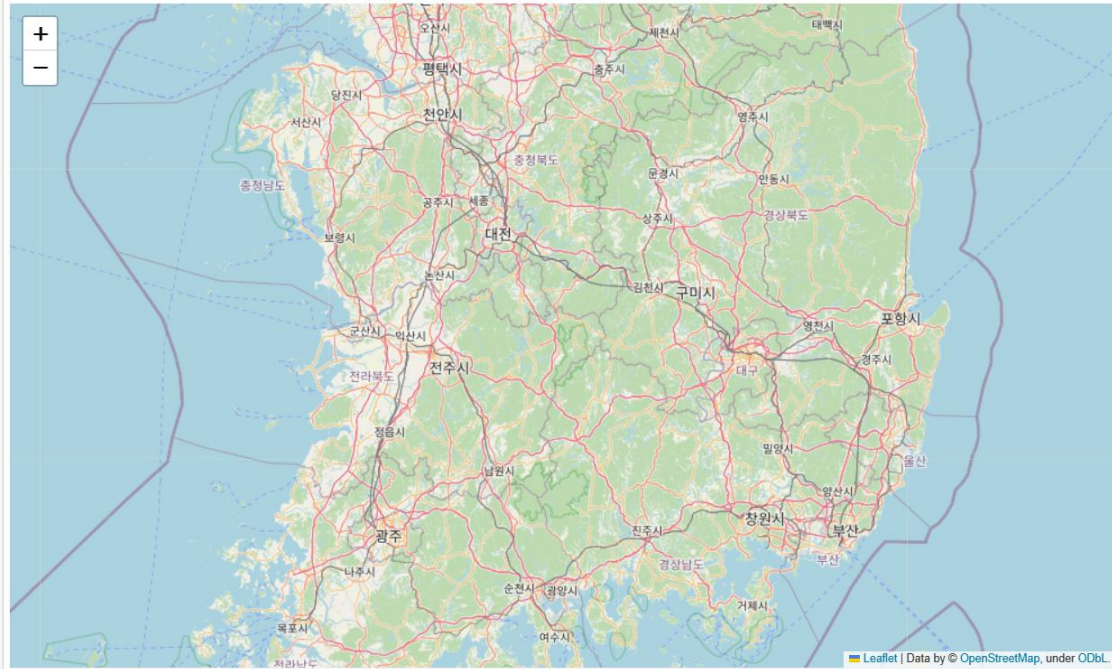
- 단계 구분도 만들기
  - Folium 패키지를 이용하여 단계 구분도를 만듦

```
1 pip install folium
```

## (1) 배경 지도 만들기

In [10]:

```
1 import folium
2 folium.Map(location = [35.95, 127.7], # 지도 중심 좌표
3           zoom_start = 8) # 확대 단계
```

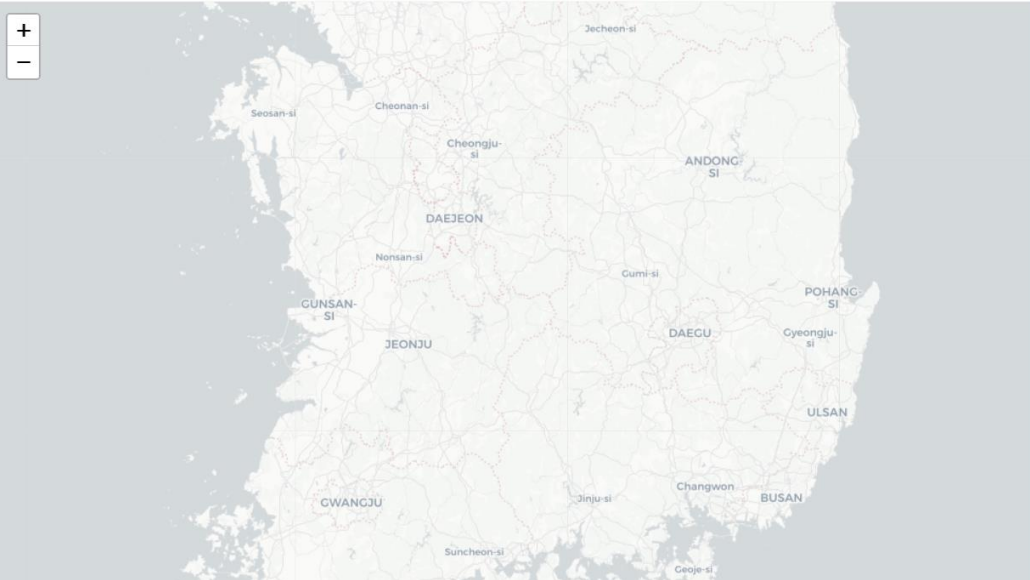




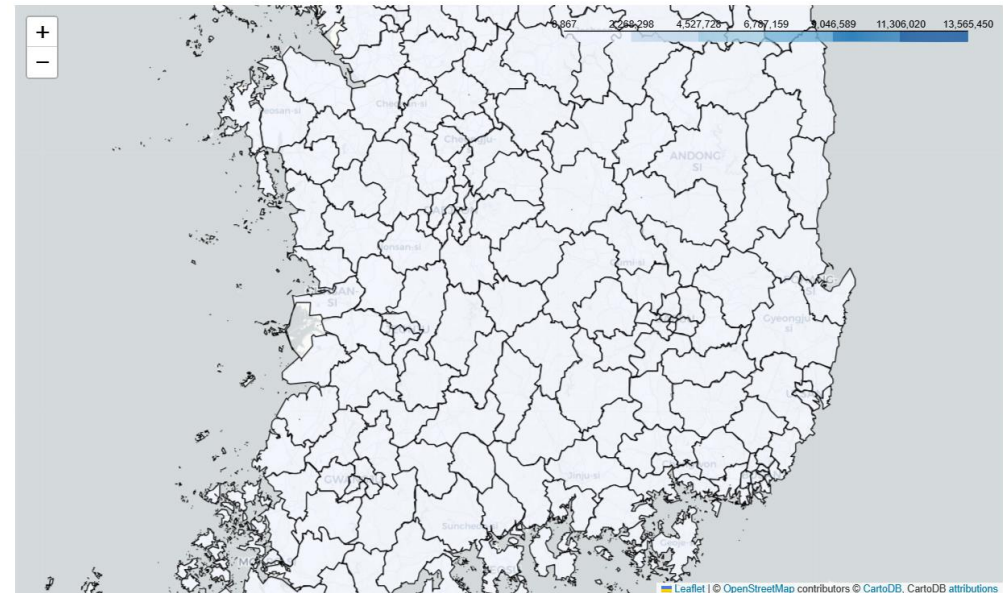
# 지도 시각화

- 단계 구분도를 만들기 위한 배경지로 저장-밖은색으로 변경
- folium.Choropleth()
  - geo\_data : 지도데이터
  - data: 색깔로 표현할 통계 데이터
  - columns: 통계 데이터의 행정 구역 코드 변수, 색깔로 표현할 변수
  - key\_on: 지도 데이터의 행정 구역 코드

```
1 map_sig = folium.Map(location = [35.95, 127.7], # 지도 중심 좌표
2                       zoom_start = 8, # 확대 단계
3                       tiles = 'cartodbpositron') # 지도 종류
4 map_sig
```



```
1 # 지도 데이터
2 # 통계 데이터
3 # df_pop 행정 구역 코드, 인구
4 # geo 행정 구역 코드
5 folium.Choropleth(geo_data = geo,
6                   data = df_pop,
7                   columns = ('code', 'pop'),
8                   key_on = 'feature.properties.SIG_CD') #
9 .add_to(map_sig)
10
11 map_sig
```



# 지도 시각화

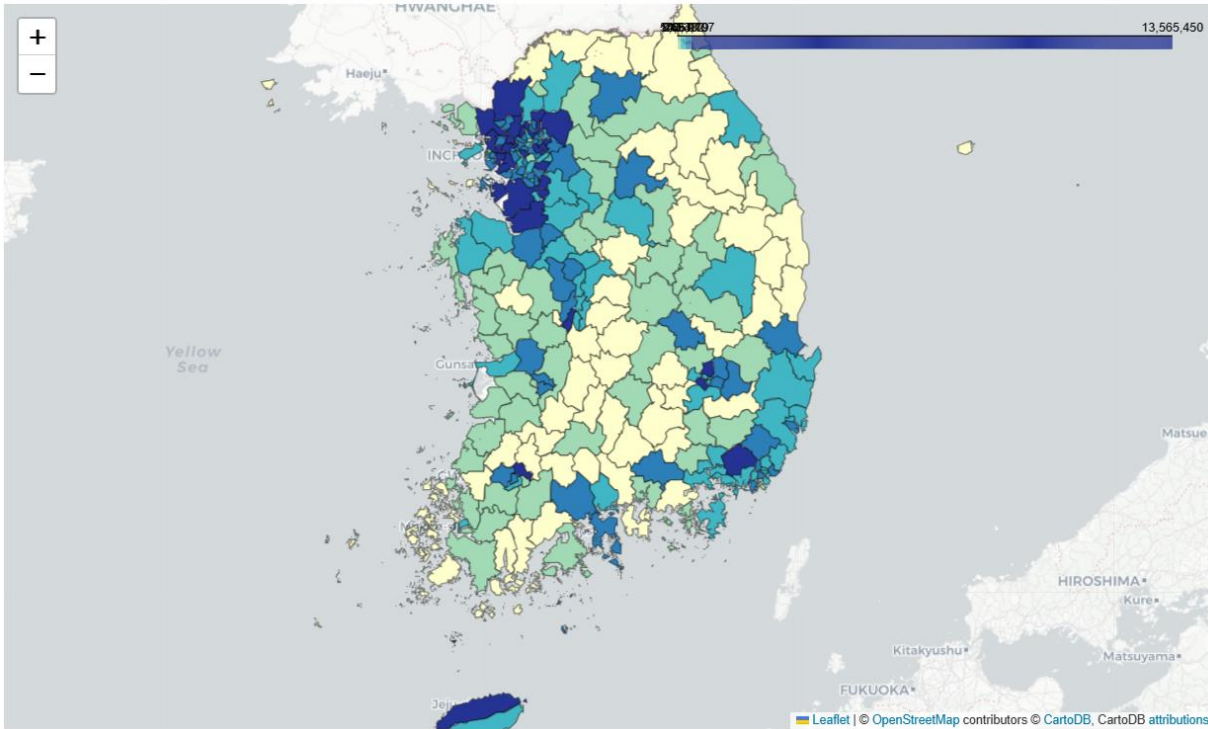
- 계급 구간 정하기
  - 분위수를 이용하여 지역을 나누는 계급 구간을 정함
  - `quantile()` : 5가지 계급구간의 하한값, 상한값을 담은 `bins`를 생성, 5가지 색으로 구분

```
In [13]: 1 bins = list(df_pop['pop'].quantile([0, 0.2, 0.4, 0.6, 0.8, 1]))  
         2 bins
```

```
Out[13]: [8867.0, 50539.6, 142382.200000000004, 266978.6, 423107.200000000024, 13565450.0]
```

# 지도 시각화

- 디자인 수정하기
  - 지도 중심 좌표, 확대단계, 지도 종류 설정
  - 인구가 많을 수록 파란색, 적을 수록 노란색에 가깝게 표현



```
1 ## 배경 지도 만들기
2
3 # 지도 중심 좌표
4 # 확대 단계
5 # 지도 종류
6 map_sig = folium.Map(location = [35.95, 127.7],
7                       zoom_start = 8,
8                       tiles = 'cartodbpositron')
```

```
1 ## 단계 구분도 만들기
2
3 # 지도 데이터
4 # 통계 데이터
5 # df_pop 행정 구역 코드, 인구
6 # geo 행정 구역 코드
7 # 컬러맵
8 # 투명도
9 # 경계선 투명도
10 # 계급 구간 기준값
11 # 배경 지도에 추가
12 folium.Choropleth(geo_data = geo,
13                   data = df_pop,
14                   columns = ('code', 'pop'),
15                   key_on = 'feature.properties.SIG_CD',
16                   fill_color = 'YlGnBu',
17                   fill_opacity = 1,
18                   line_opacity = 0.5,
19                   bins = bins) #
20 .add_to(map_sig)
21
22 map_sig
```