

지도 시각화 II

(GeoJson과 Choropleth 활용)

최수연 교수

mibm400@hanmail.net

학습목표

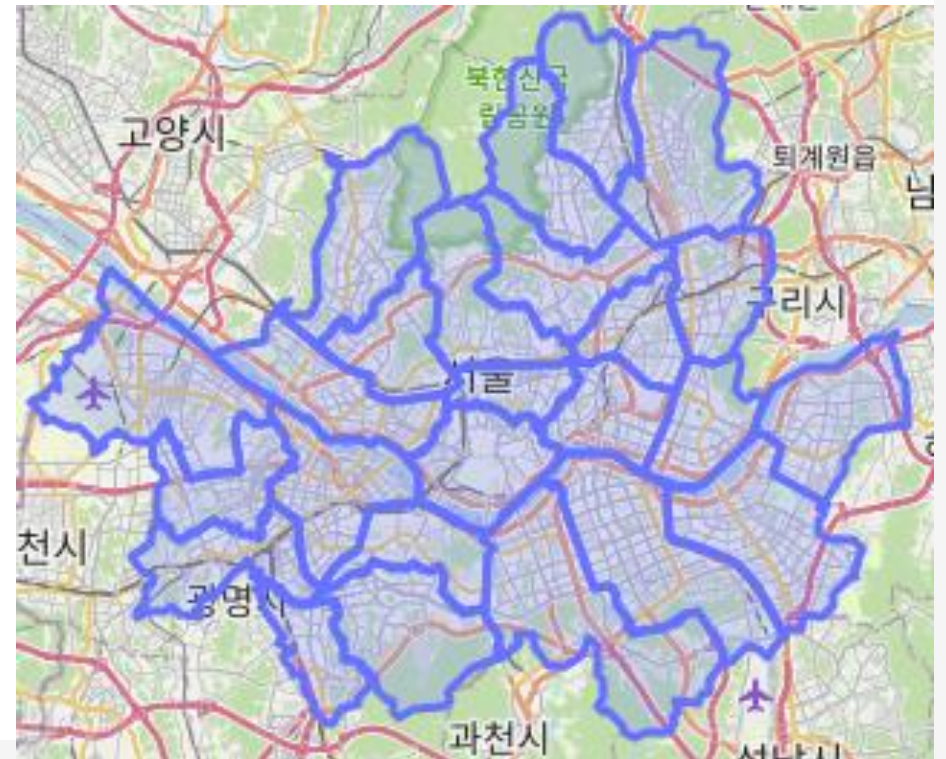
- GeoJson과 단계구분도를 설명할 수 있다.
- GeoJson과 공공데이터를 활용하여 단계구분도를 포함한 지도를 생성할 수 있다.

목차

- GeoJSON와 단계구분도 이해하기
- 지도에 경계선 표시하기
- 데이터에 따른 단계구분도 추가하기

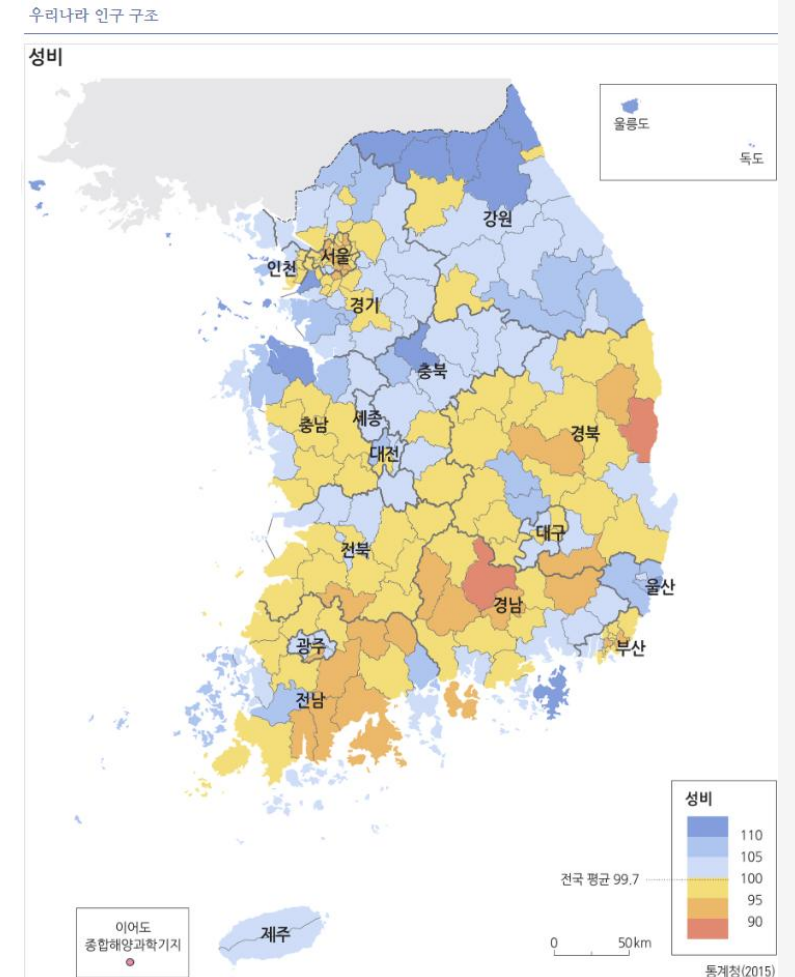
GeoJson

- 위치 정보를 기반으로 지형을 표현하기 위해 설계된 개방형 공개 표준 형식
- 행정구역과 같은 경계선이나 경로를 표현하는데 활용
- 위치 정보가 **(경도, 위도, (고도))** 순서로 저장됨
- ✓ 구글맵이나 OSM에서는 (위도, 경도) 순으로 저장됨
- 확장자 : *.json, *.geojson
- 웹 또는 모바일에서 데이터 시각화 가능
- 다른 포맷에 비해 위치 데이터 용량이 적음



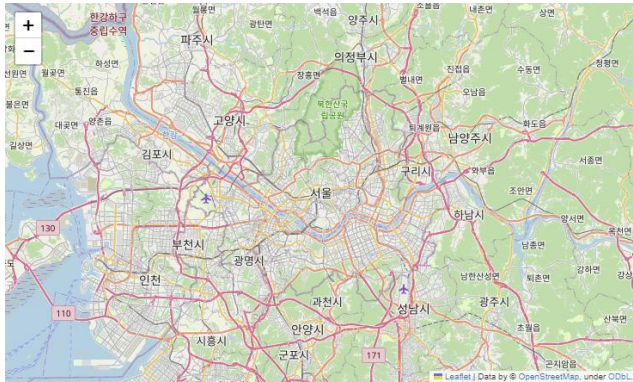
단계 구분도

- 단계 구분도(Choropleth Map)
 - ✓ 지역별간의 분포 차이 및 통계치를 색깔이나 그라데이션으로 구분하여 표현한 지도
 - ✓ 수치적인 데이터를 색상으로 단계적으로 표현한 지도

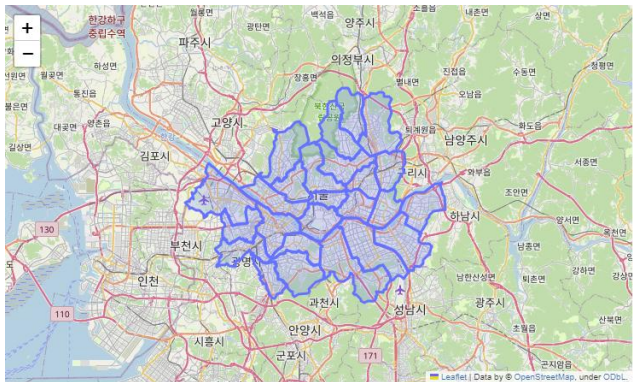


GeoJson과 단계구분도를 이용한 시각화 순서

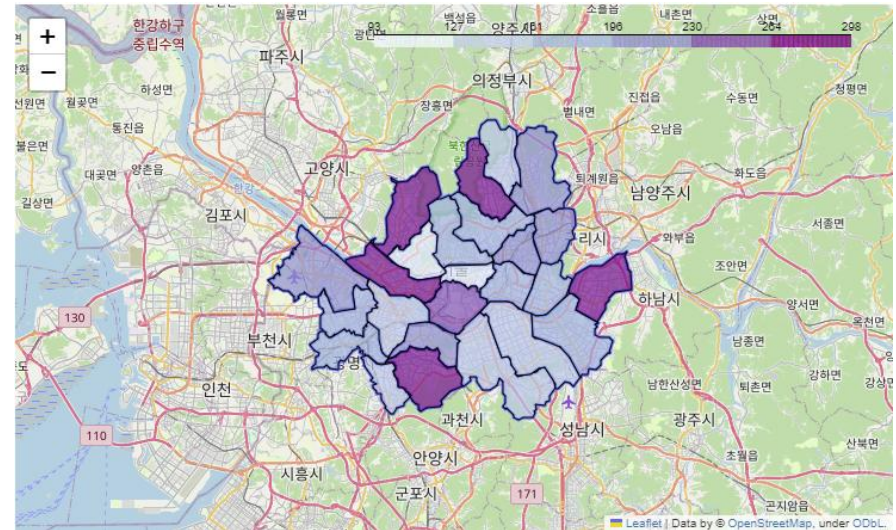
① folium을 활용하여 Map 그리기



② GeoJson으로 행정구역 경계 확인하기



③ Choropleth를 지도에 추가하기



GeoJson 파일 읽어오기

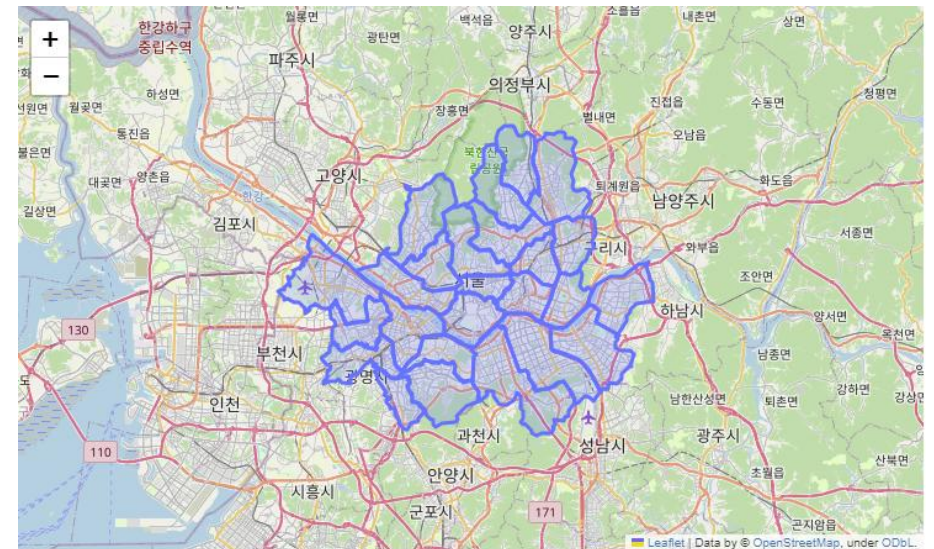
- GeoJson 파일을 읽어들이기 위한 라이브러리 선언

```
import json
```

- GeoJson 파일 읽어오기
- ✓ 변수명 = json.load(open('파일경로명', encoding='인코딩방식'))
- load() : JSON 문자열을 파이썬 객체로 변경
 - open() : “파일경로명”의 파일을 읽을 수 있도록 열어줌
 - 파일경로명 : GeoJson 또는 Json 파일의 경로명
 - encoding='인코딩방식': 한글이 깨져서 보일 경우 인코딩 방식 설정
 - 인코딩방식 : EUC-KR(2byte 한글인코딩), cp949(MS office), utf-8(유니코드 인코딩)

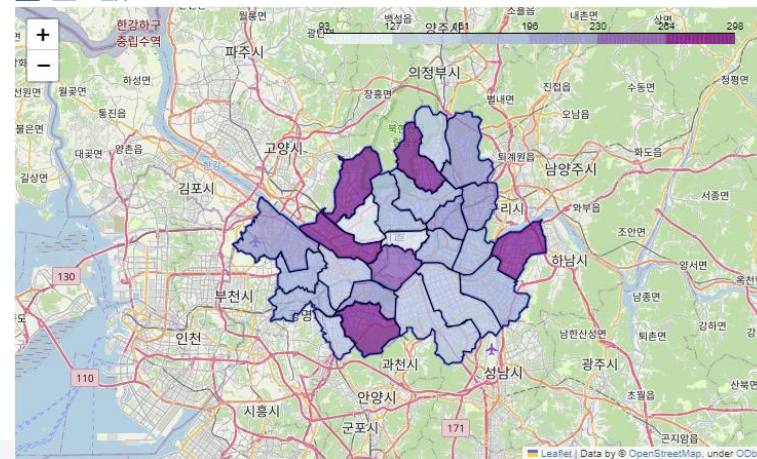
GeoJson으로 지도에 행정구역 경계선 추가하기

- GeoJson 좌표를 지도에 추가하기
- ✓ folium.**GeoJson**(**Json변수명**).**add_to**(**지도변수명**)
 - **GeoJson(Json변수명)** : GeoJson 데이터에서 좌표들을 지도에 표현



지도에 단계구분도 추가하기

- 단계구분도를 지도에 추가하기
 - ✓ `folium.Choropleth(속성들).add_to(지도변수명)`
 - `geo_data = "지도 데이터": *.geojson, *.json의 지도 데이터`
 - `data = "시각화 하고자 하는 데이터": DataFrame 형식의 데이터`
 - `columns = (열이름1, 열이름2)`
 - 열이름1 : 지도 데이터와 매핑할 문자열 데이터의 열이름으로 지도 데이터와 동일한 값이 있는 열이름
 - 열이름2 : 색상으로 나누어질 수치 데이터의 열이름으로 시각화 하고자 하는 열이름
 - `key_on = "매핑할 geo데이터": geojson 데이터에서 시각화하고자 하는 데이터와 동일한 값`
 - `fill_color = "색상": 시각화에 사용될 색상`
 - `fill_opacity = 실수 : 채워진 색상의 투명도(0~1사이의 실수)`
 - `line_weight = 정수 : 경계선 두께`
 - `line_opacity = 실수 : 경계선 투명도 (0~1사이의 실수)`
 - `nan_fill_color = "색상": 통계수치가 없는 구역의 색상`



단계 구분도를 표현하기 위한 준비물

- 서울시 시군구의 유기동물보호현황에 대한 단계 구분도를 그리기 위한 준비물
 - ✓ 서울시 지도
 - folium.Map()을 사용해서 생성
 - ✓ 행정구역 경계를 나타내는 지도 데이터(GeoJson)
 - 행정구역 코드, 행정구역 이름(영문/국문), 행정구역 경계의 경도와 위도 좌표
 - 서울시군구.geojson
 - ✓ 지도상에 단계별로 색상을 나타내기 위한 수치 데이터(공공데이터)
 - 2022년유기동물보호현황.xlsx

행정구역 경계를 나타내는 지도 데이터

- 온라인 실습자료에 제공

 서울시군구.geojson

```
JSON
{
  type: "FeatureCollection",
  name: "서울시군구",
  crs: {
    type: "name"
  },
  properties: {},
  features: [
    {
      type: "Feature",
      properties: {
        SIG_CD: "11110",
        SIG_ENG_NM: "Jongno-gu",
        SIG_KOR_NM: "종로구"
      },
      geometry: {
        type: "MultiPolygon",
        coordinates: []
      }
    },
    {
      type: "Feature",
      properties: {
        SIG_CD: "11140",
        SIG_ENG_NM: "Jung-gu",
        SIG_KOR_NM: "중구"
      },
      geometry: {
        type: "MultiPolygon",
        coordinates: []
      }
    },
    {
      type: "Feature",
      properties: {
        SIG_CD: "11150",
        SIG_ENG_NM: "Jongno-gu",
        SIG_KOR_NM: "종로구"
      },
      geometry: {
        type: "MultiPolygon",
        coordinates: []
      }
    }
  ]
}
```

2022년 서울시 유기동물 보호 현황

• 2022년 서울시 유기동물 보호 현황

✓ <https://data.seoul.go.kr/dataList/369/S/2/datasetView.do>

✓ 2022년유기동물보호현황.xlsx



통계

활용사례(갤러리) 등록 URL 복사 목록 이동

서울시 유기동물보호 현황 통계

○ 통계개요

- * 통계명 : 유기동물보호현황
- * 통계종류 : 서울시 자치구별 유기동물보호현황을 제공하는 일반 · 보고통계
- * 작성목적 : 유기동물보호현황의 객관적인 통계수치를 파악하여 동물유기행위에 대한

[전체 설명보기](#)

유기동물보호 현황

자료갱신일 : 2023-04-13 / 수록기간 : 년 2003 ~ 2022
출처 : 서울특별시, 시정통계



항목[1/1]

자치구별[26/26]

현황별[16/16]

시점[1/20]

(단위 : 마리)

새창보기

주소보기

행렬전환

부가기능설정

분석

차트

다운로드

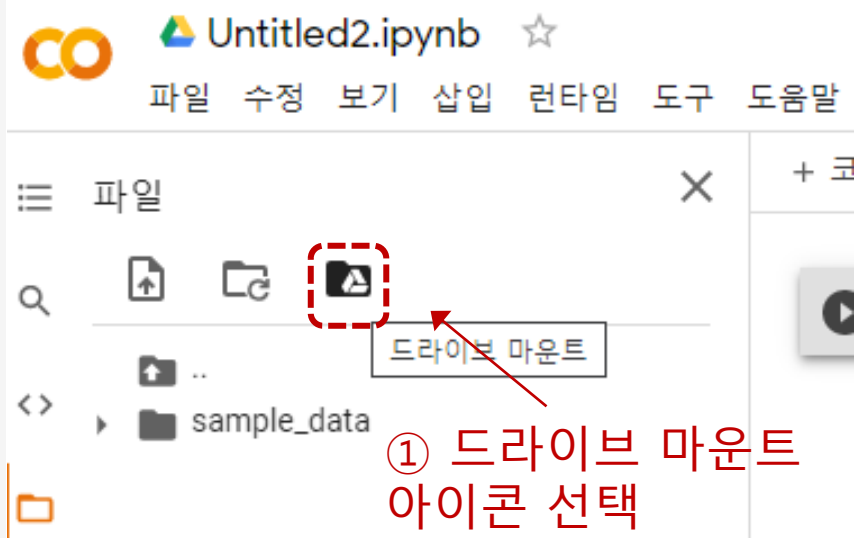
인쇄

도움말

자치구별(1)	자치구별(2)	2022					
		합계					
		소계	개	인도(주인)		입양분양	폐사안락사
		소계	소계	인도(주인)	입양분양	폐사안락사	
	용산구	238	118	79	33		
	성동구	165	70	34	17		
	광진구	167	67	33	27		
	동대문구	201	94	34	41		
	종로구	199	102	48	35		
	성북구	168	115	48	35		
	강북구	270	162	56	63		
	도봉구	137	82	45	24		
	노원구	215	137	48	54		
	은평구	273	152	62	39		
	서대문구	120	80	30	23		
	마포구	286	82	45	26		
	양천구	189	93	54	22		
	강서구	222	119	49	42		
	구로구	180	94	43	32		
	금천구	148	100	33	38		
	영등포구	152	72	37	23		
	동작구	224	68	41	17		
	관악구	298	199	74	62		
	서초구	130	77	23	26		
	강남구	147	92	41	37		
	송파구	173	69	35	17		
	강동구	284	132	71	49		

Colab Drive Mount하여 파일 업로드

- GeoJson 파일 사이즈가 크므로 Google Drive에 업로드



+ 코드 + 텍스트

[] 1 ② 셀에 아래의 코드가 나타나면
Shift + Enter로 실행

```
1 from google.colab import drive
2 drive.mount('/content/drive')
```

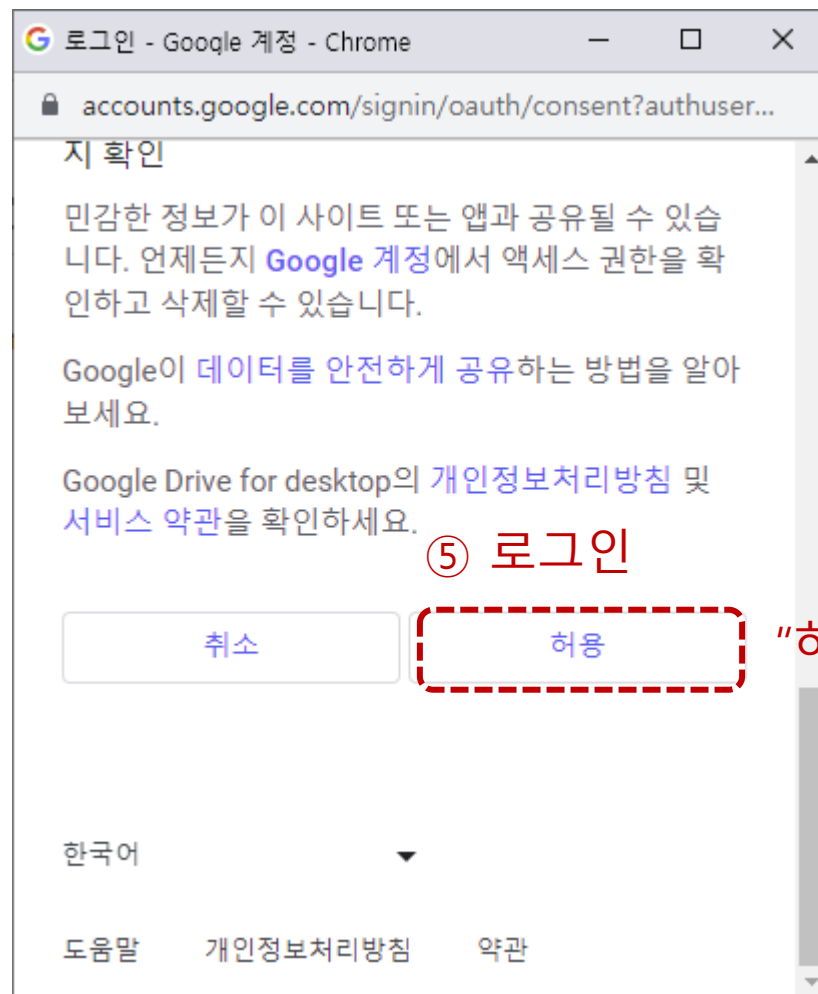
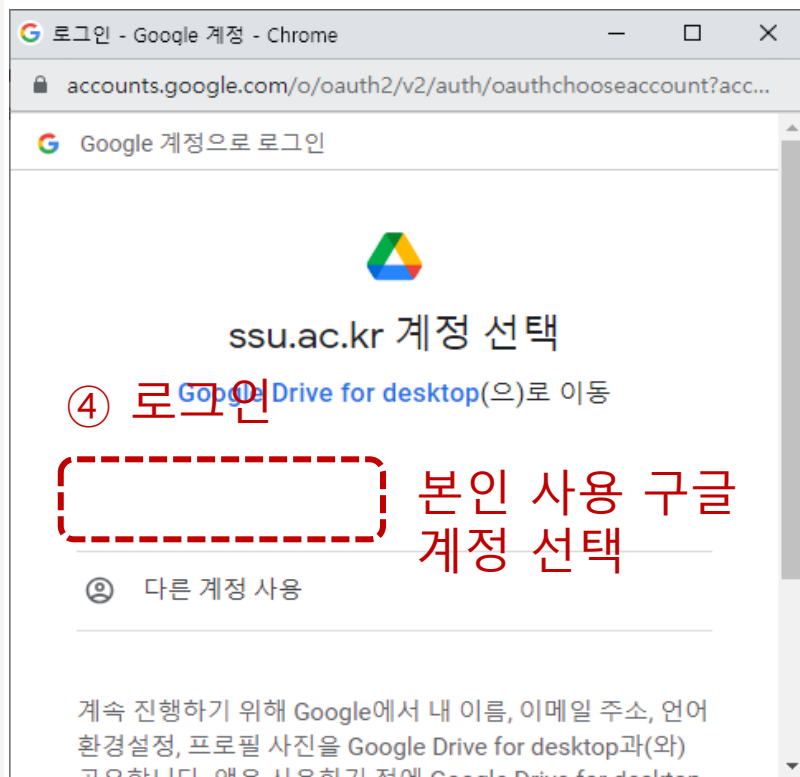
노트북이 **Google Drive** 파일에 액세스하도록 허용하시겠습니까?

Google Drive에 연결하면 액세스 권한이 취소될 때까지 이 노트북에서 실행된 코드가 Google Drive의 파일을 수정할 수 있습니다.

아니요

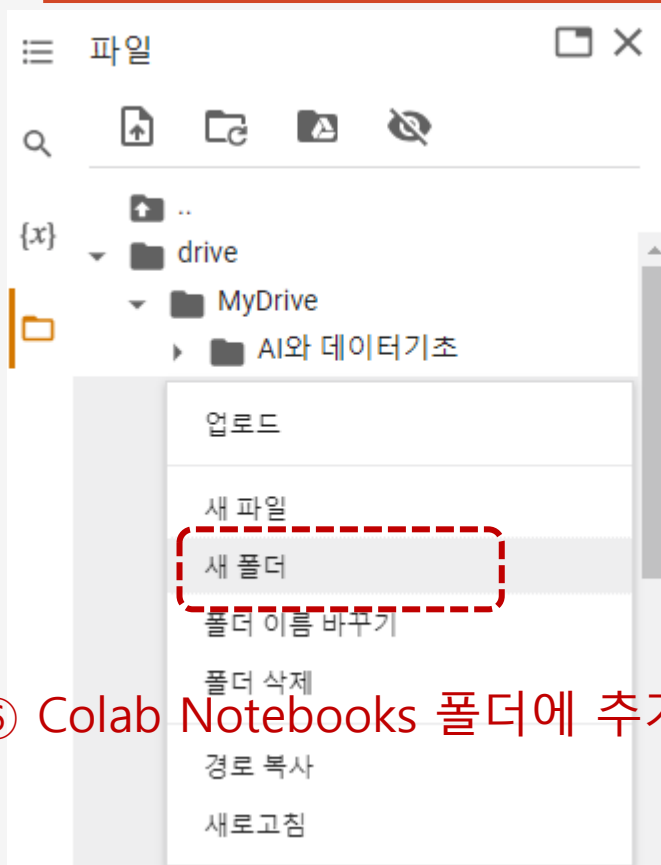
③ 선택
Google Drive에 연결

Colab Drive Mount하여 파일 업로드

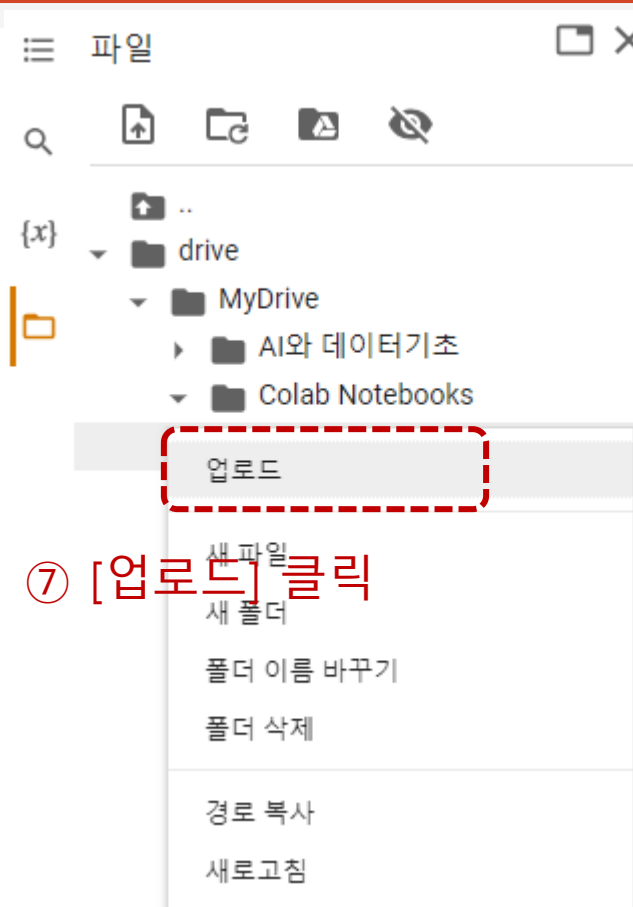


"허용" 클릭

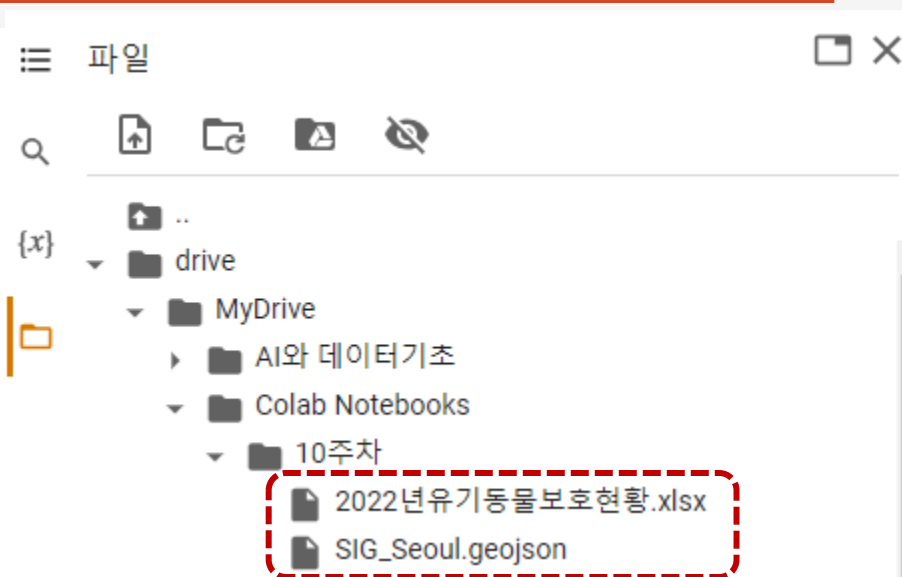
Colab Drive Mount하여 파일 업로드



⑥ Colab Notebooks 폴더에 추가



⑦ [업로드] 클릭



⑧ 2022년유기동물보호현황.xlsx 파일과
SIG_Seoul.geojson 파일 업로드

준비파일 읽고 확인하기(유기동물보호현황)

```
1 import pandas as pd
2 adData = pd.read_excel('/content/drive/MyDrive/Colab Notebooks/10주차/2022년유기동물보호현황.xlsx',
3                         header=[2,3])
4 adData.head()
```

자치구별 (1)	자치구별 (2)	소계	개	고양이										기타
자치구별 (1)	자치구별 (2)	소계	소계	인도 (주 인)	입양분 양	폐사안락 사	계류기 증	소계	인도 (주 인)	입양분 양	폐사안락 사	계류기 증	소 계	
0	합계	소계	4870	2491	1110	814	536	31	2134	56	621	1393	64	24
1	합계	종로구	191	79	28	22	29	-	103	1	15	87	-	
2	합계	중구	93	36	19	10	7	-	47	1	13	33	-	1
3	합계	용산구	238	118	79	33	5	1	115	11	48	56	-	
4	합계	성동구	165	70	34	17	19	-	91	3	11	77	-	

준비파일 읽고 확인하기(유기동물보호현황)

• 유기동물보호현황 데이터의 정보 확인하기

#	Column	Non-Null Count	Dtype
0	(자치구별(1), 자치구별(1))	26 non-null	object
1	(자치구별(2), 자치구별(2))	26 non-null	object
2	(소계, 소계)	26 non-null	int64
3	(개, 소계)	26 non-null	int64
4	(개, 인도(주인))	26 non-null	int64
5	(개, 입양분양)	26 non-null	int64
6	(개, 폐사안락사)	26 non-null	int64
7	(개, 계류기종)	26 non-null	object
8	(고양이, 소계)	26 non-null	int64
9	(고양이, 인도(주인))	26 non-null	object
10	(고양이, 입양분양)	26 non-null	int64
11	(고양이, 폐사안락사)	26 non-null	int64
12	(고양이, 계류기종)	26 non-null	object
13	(기타, 소계)	26 non-null	int64
14	(기타, 인도(주인))	26 non-null	object
15	(기타, 입양분양)	26 non-null	object
16	(기타, 폐사안락사)	26 non-null	object
17	(기타, 계류기종)	26 non-null	object

* (자치구별(2), 자치구별(2)) 열

- GeoJson 데이터 중 동일한 값으로 매칭되는 열
- object(문자열 데이터)

* (소계, 소계) 열

- 단계별로 색상을 구분하기 위한 기준열
- int64(수치 데이터)

```
adData.info()
```

준비파일 읽고 확인하기(GeoJson)

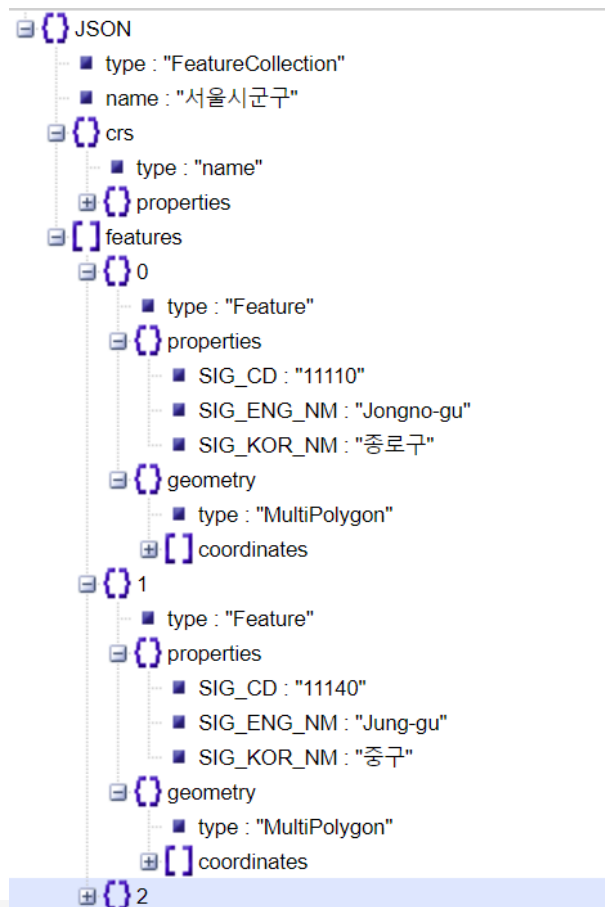
- GeoJson 파일 읽어오기

```
1 import json
2
3 geo_json = json.load(open('/content/drive/MyDrive/Colab Notebooks/10주차/서울시군구.geojson'))
4 geo_json
```

```
{'type': 'FeatureCollection',
 'name': '서울시군구',
 'crs': {'type': 'name',
 'properties': {'name': 'urn:ogc:def:crs:OGC:1.3:CRS84'}},
 'features': [{'type': 'Feature',
 'properties': {'SIG_CD': '11110',
 'SIG_ENG_NM': 'Jongno-gu',
 'SIG_KOR_NM': '종로구'},
 'geometry': {'type': 'MultiPolygon',
 'coordinates': [[[[[127.00864326221884, 37.580468252047105],
 [127.00871274905404, 37.58045116513156],
 [127.00876564011087, 37.580443107078565],
 [127.00890785297045, 37.580424231608646],
 [127.00913781377908, 37.58039352939572],
 [127.00916523299747, 37.58039071654549],
 [127.00923792440939, 37.58038254887839],
 [127.00926870590294, 37.58037916816284],
 [127.0092779097711, 37.58037803231545],
 [127.0094301670040, 37.580356622257295],
 [127.00864326221884, 37.580468252047105]]]]]]]
```

준비파일 읽고 확인하기(GeoJson)

• GeoJson 데이터 확인하기



`geo_json['features'][0]['properties']` [properties 내용 확인]

`{'SIG_CD': '11110', 'SIG_ENG_NM': 'Jongno-gu', 'SIG_KOR_NM': '종로구'}`

공공데이터에서 매칭되는 부분

[geometry 내용 확인]

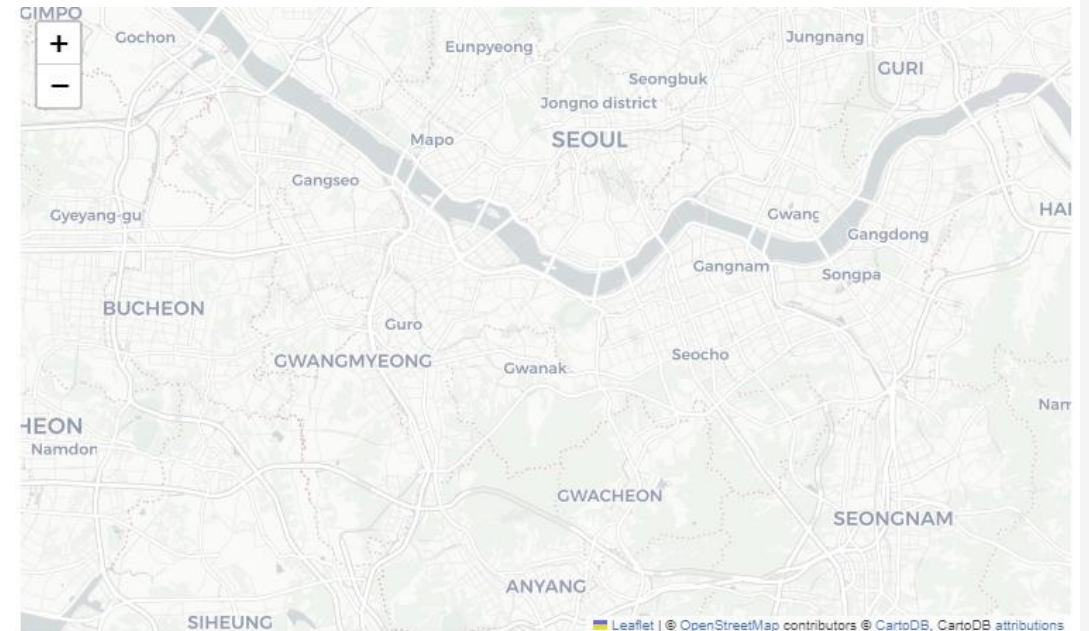
`geo_json['features'][0]['geometry']`

```
{'type': 'MultiPolygon',
 'coordinates': [[[[[127.00864326221884, 37.580468252047105],
 [127.00871274905404, 37.58045116513156],
 [127.00876564011087, 37.580443107078565],
 [127.00890785297045, 37.580424231608646],
 [127.00913781377908, 37.58039352939572],
 [127.00916523299747, 37.58039071654549],
 [127.00923792440939, 37.58038254887839],
 [127.00926870590294, 37.58037916816284],
 [127.0092779097711, 37.58037803231545],
 [127.0094381678049, 37.580356632257235],
 [127.0100319799304, 37.58027803392697],
 [127.01003587429, 37.58027746602466]
```

folium을 활용하여 Map 그리기

- **folium.Map()**을 활용하여 지도 생성

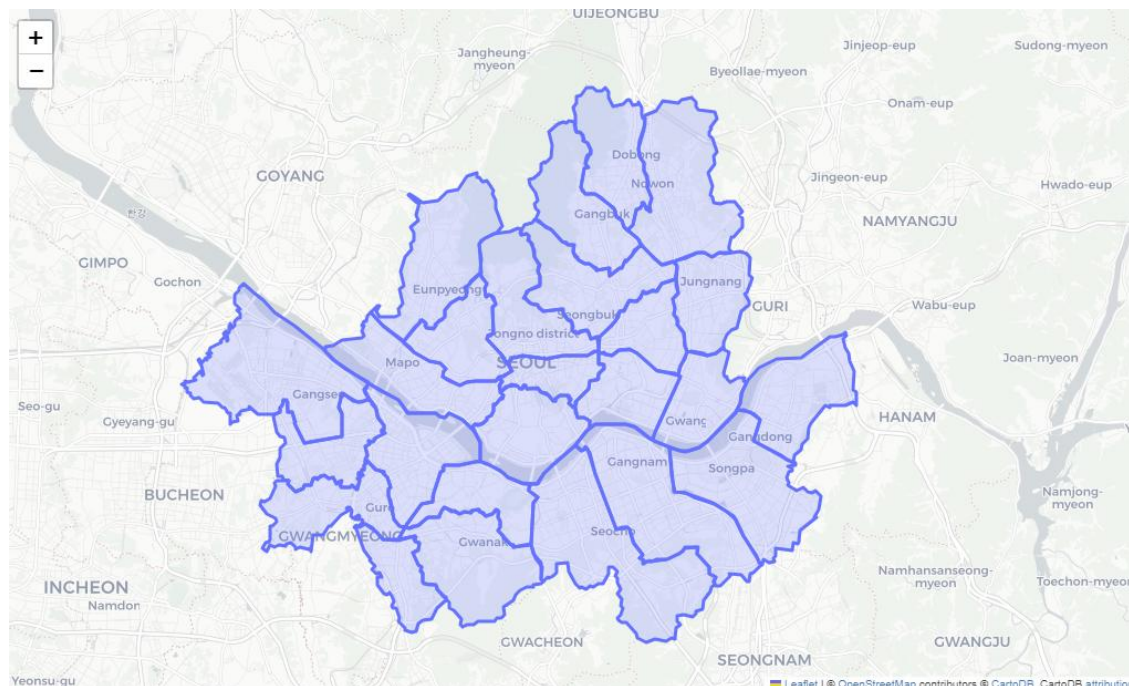
```
1 import folium
2
3 map = folium.Map(location=[37.4965,126.9572],
4                       zoom_start=11,
5                       tiles='cartodbpositron')
6 map
```



GeoJson으로 행정구역 경계 확인하기

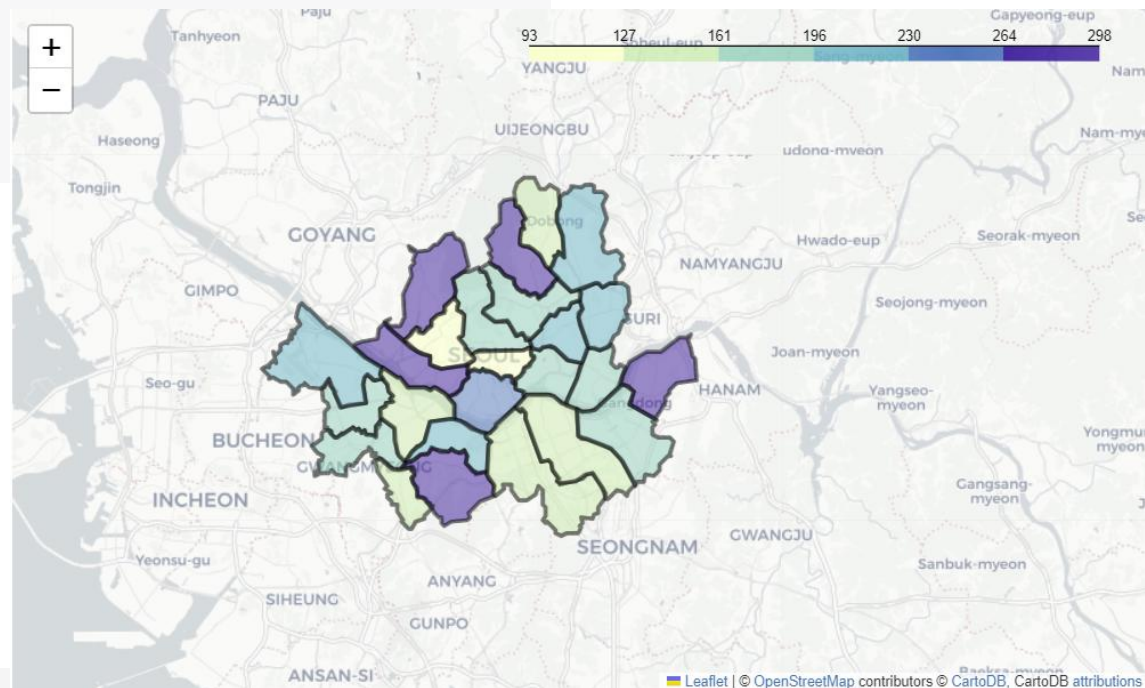
- GeoJson 좌표를 지도에 추가하기
 - ✓ folium.**GeoJson**(**Json변수명**).**add_to**(**지도변수명**)
 - **GeoJson(Json변수명)** : GeoJson 데이터에서 좌표들을 지도에 표현

```
1 folium.GeoJson(geo_json).add_to(map)
2 map
```



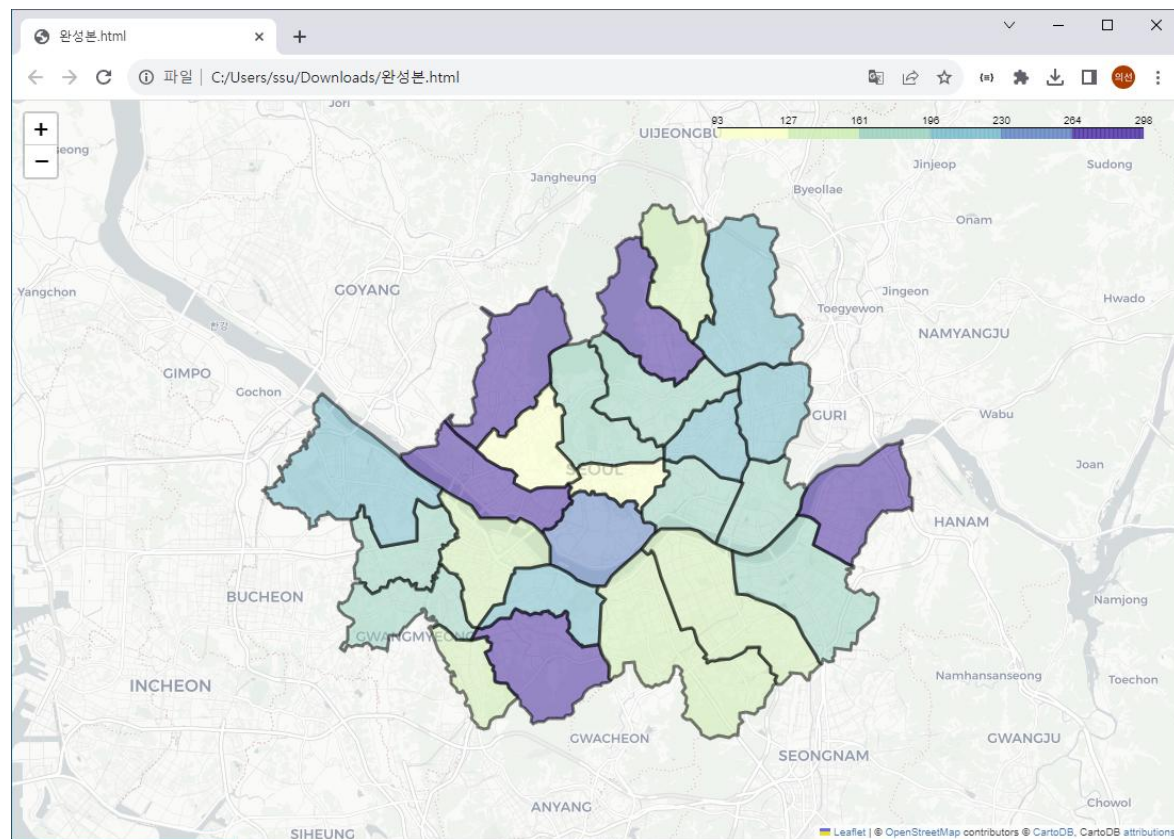
Choropleth를 지도에 추가하기

```
1 folium.Choropleth(geo_data = geo_json,  
2                   data = adData[1:],  
3                   columns=[('자치구별(2)', '자치구별(2)'), ('소계', '소계')],  
4                   key_on='feature.properties.SIG_KOR_NM',  
5                   fill_color = 'YlGnBu',  
6                   fill_opacity=0.5,  
7                   line_weight=3,  
8                   line_opacity=0.5).add_to(map)  
9 map
```



완성된 지도를 HTML로 저장하기

- 지도변수명.**save**('html파일명')
- ✓ **html파일명** : *.html 형식의 파일명



SHP 파일에서 원하는 지역을 Geojson으로 바꾸기

SHP와 Geojson 의 차이점

- SHP(shape 파일)
 - ✓ GIS(Geographical Information System)에서 사용
 - ✓ 선, 점, 다각형(Polygon)으로 벡터의 형태로 지형을 표현하는 텍스트 파일
 - ✓ 4개의 파일로 구성 : dbf, shp, shx, prj
 - ✓ 지도 분석 혹은 특정위치(공간) 편집으로 활용
 - ✓ 국가공간정보포털 오픈마켓과 (주)지오서비스(<http://www.gisdeveloper.co.kr/?p=2332>)에서 제공
- GeoJson
 - ✓ 위치 정보를 점으로 지형을 표현한 개방형 공개 표준 형식
 - ✓ *.json 파일
 - ✓ shp 파일에 비해 파일 사이즈가 작고 처리 속도가 빠름
 - ✓ 웹 또는 모바일에서 데이터 시각화 가능

SHP를 Geojson 으로 변경하는 순서

- SHP에서 원하는 지역만 추출하고 geojson으로 변경하기

- ① SHP파일 다운로드

- <http://www.gisdeveloper.co.kr/?p=2332>

- <http://data.nsdi.go.kr/dataset>

- ② QGIS에서 원하는 지역만 추출하고 GeoJson으로 저장하기

- <https://www.qgis.org/ko/site/forusers/download.html#> 에서 설치파일 다운로드 및 설치

- ③ GeoJson 확인하기

- <https://geojson.io/#map=2/0/20>

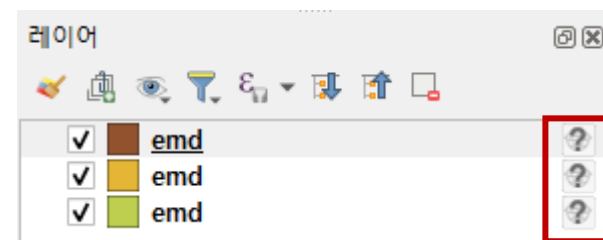
SHP 파일 다운로드하기

- <http://www.gisdeveloper.co.kr/?p=2332>
- 시군구에 해당하는 2022년 3월 자료 다운로드하기



시군구

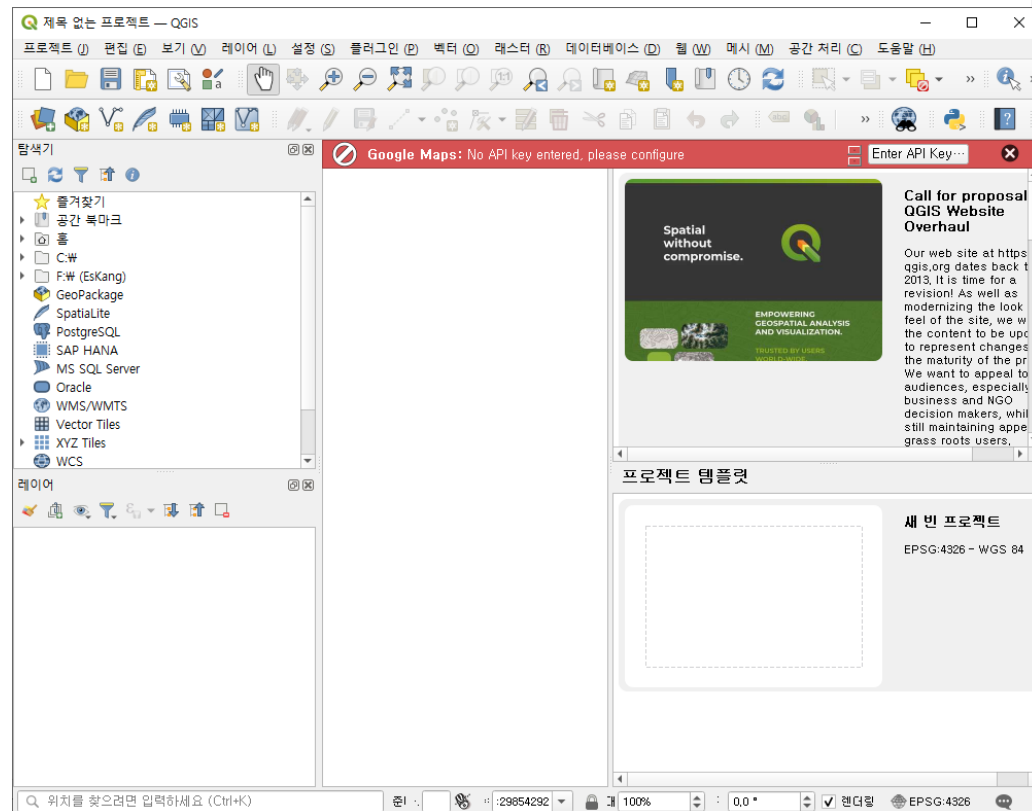
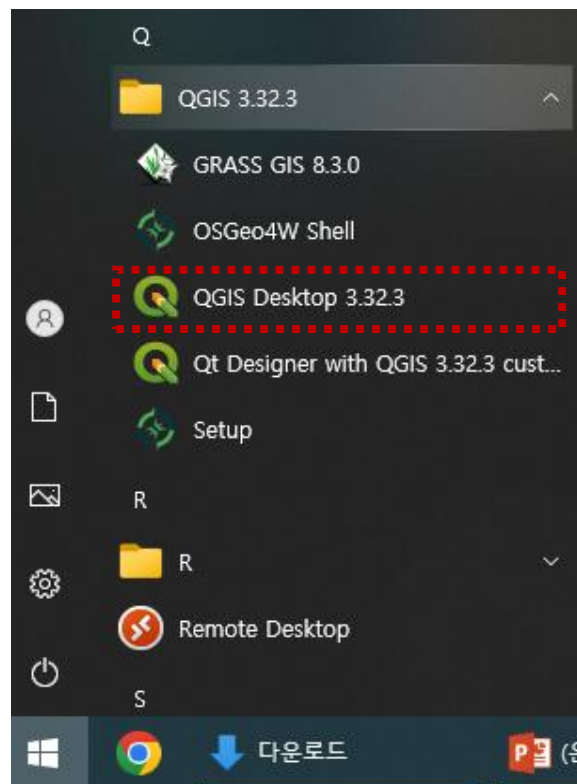
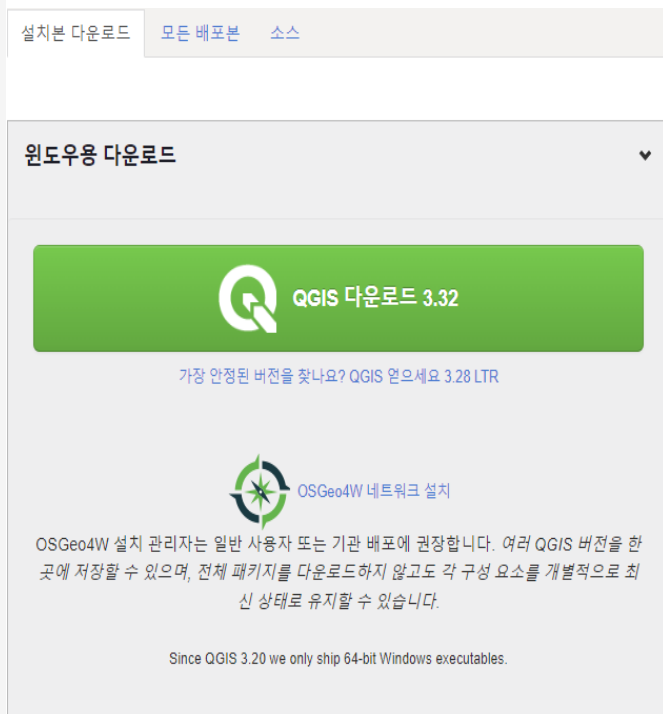
[2023년 7월 업데이트 다운로드](#)
[2023년 2월 업데이트 다운로드](#)
[2022년 11월 업데이트 다운로드](#)
[2022년 3월 업데이트 다운로드](#)
[2021년 1월 업데이트 다운로드](#)
[2020년 5월 업데이트 다운로드](#)
[2019년 5월 업데이트 다운로드](#)
[2019년 2월 업데이트 다운로드](#)
[2018년 4월 업데이트 다운로드](#)
[2017년 3월 업데이트 다운로드](#)
[2016년 2월 업데이트 다운로드](#)
[2015년 6월 업데이트 다운로드](#)
[2014년 7월 업데이트 다운로드](#)
[2014년 5월 업데이트 다운로드](#)
[2013년 11월 업데이트 다운로드](#)



벡터레이어 추가시 '?'가 보이면 좌표계 변경이 안됨

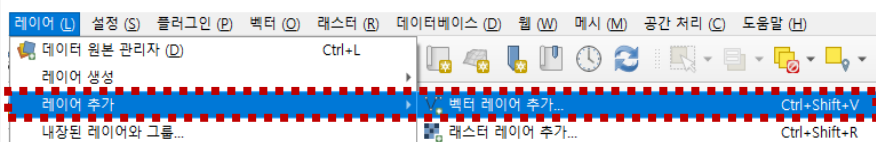
QGIS 설치하기

1. 본인의 PC 운영체제에 맞는 설치 파일 다운로드하기
<https://www.qgis.org/ko/site/forusers/download.html#>
2. 설치하기
3. QGIS 실행하기

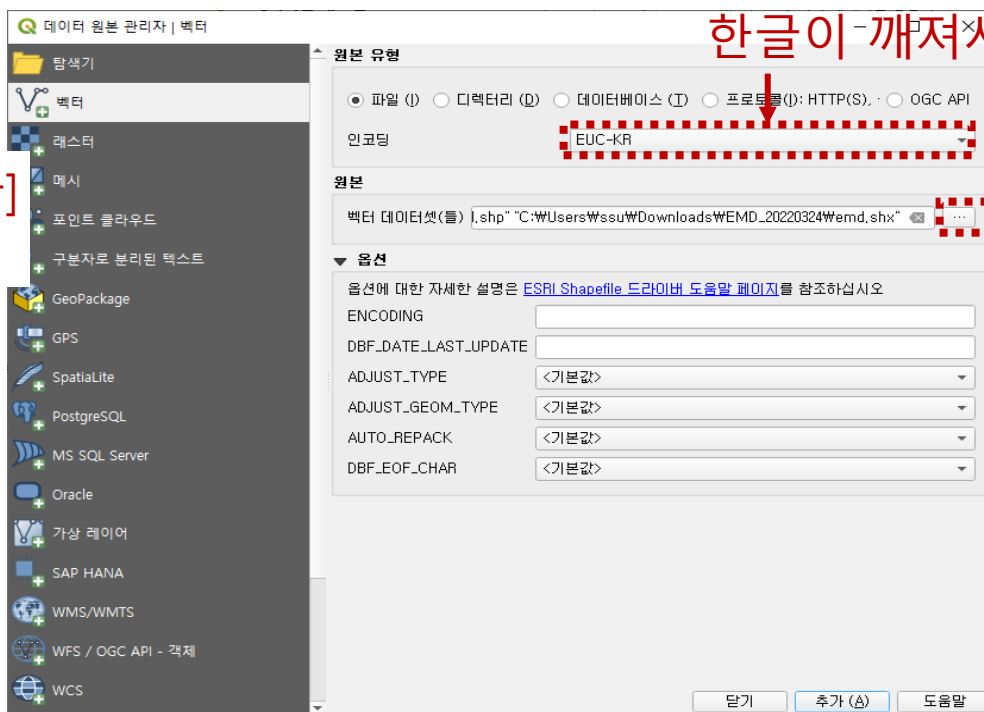
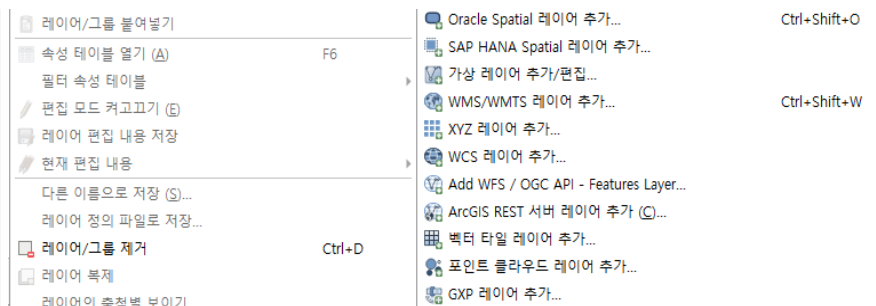


QGIS에서 SHP 파일 읽어오기

• QGIS에서 SHP 파일 읽어오기

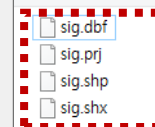


[레이어]->[레이어추가]->[벡터 레이어추가]
클릭

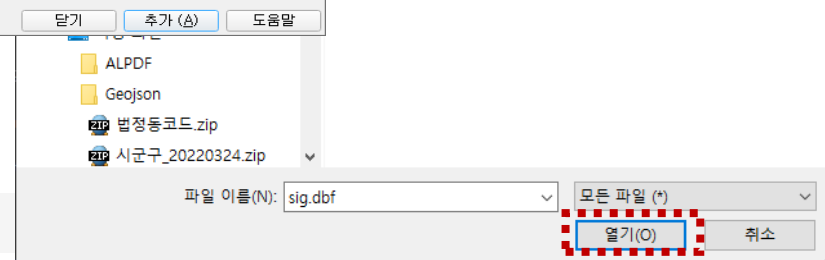


한글이 깨져서 보일경우 인코딩 변경

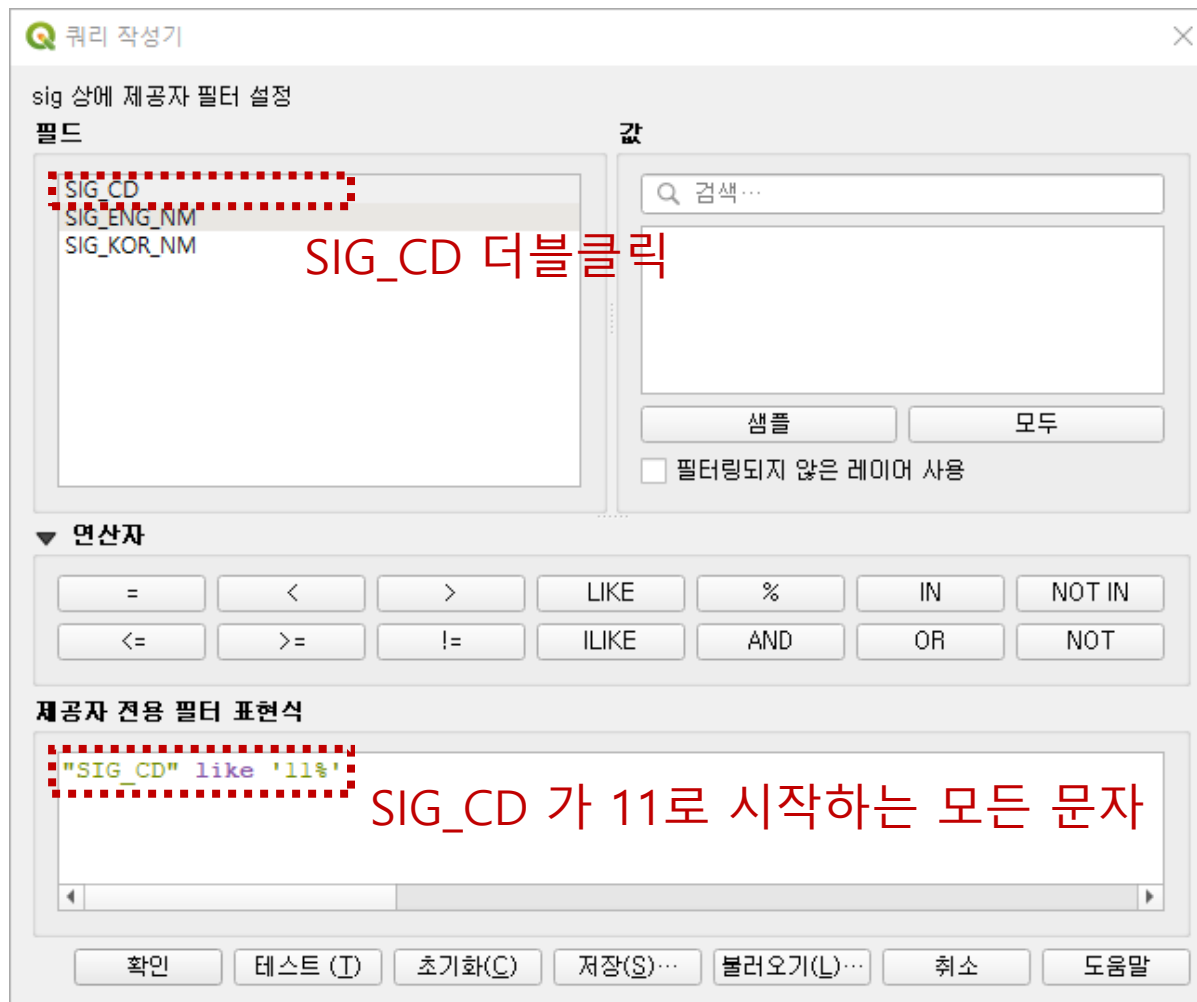
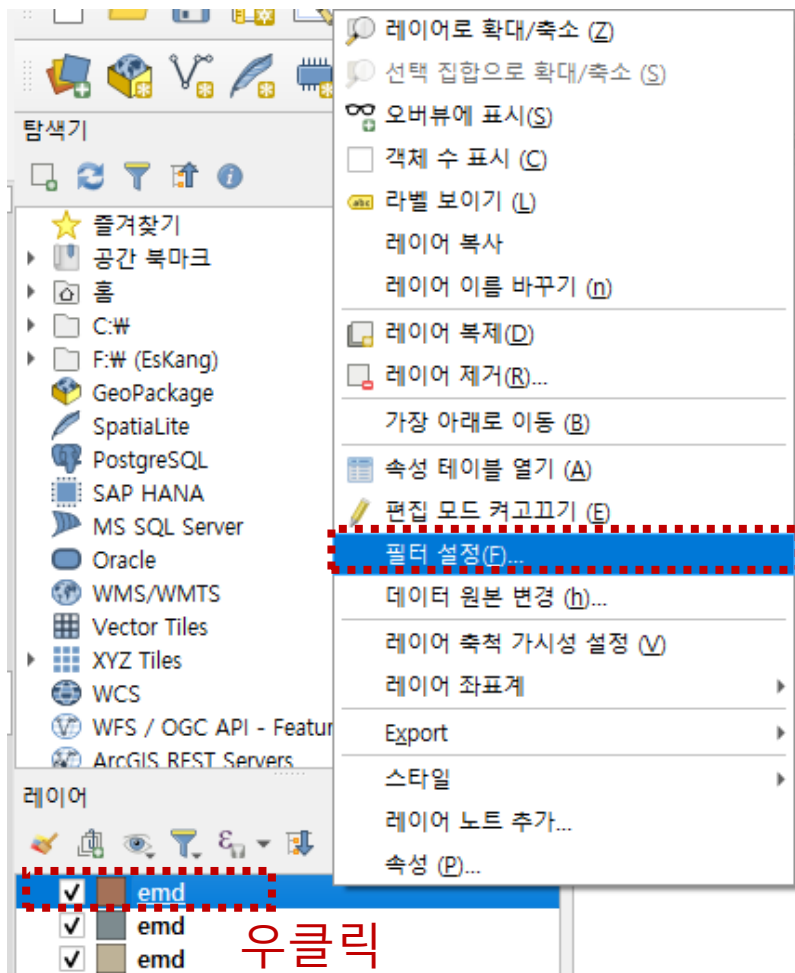
클릭



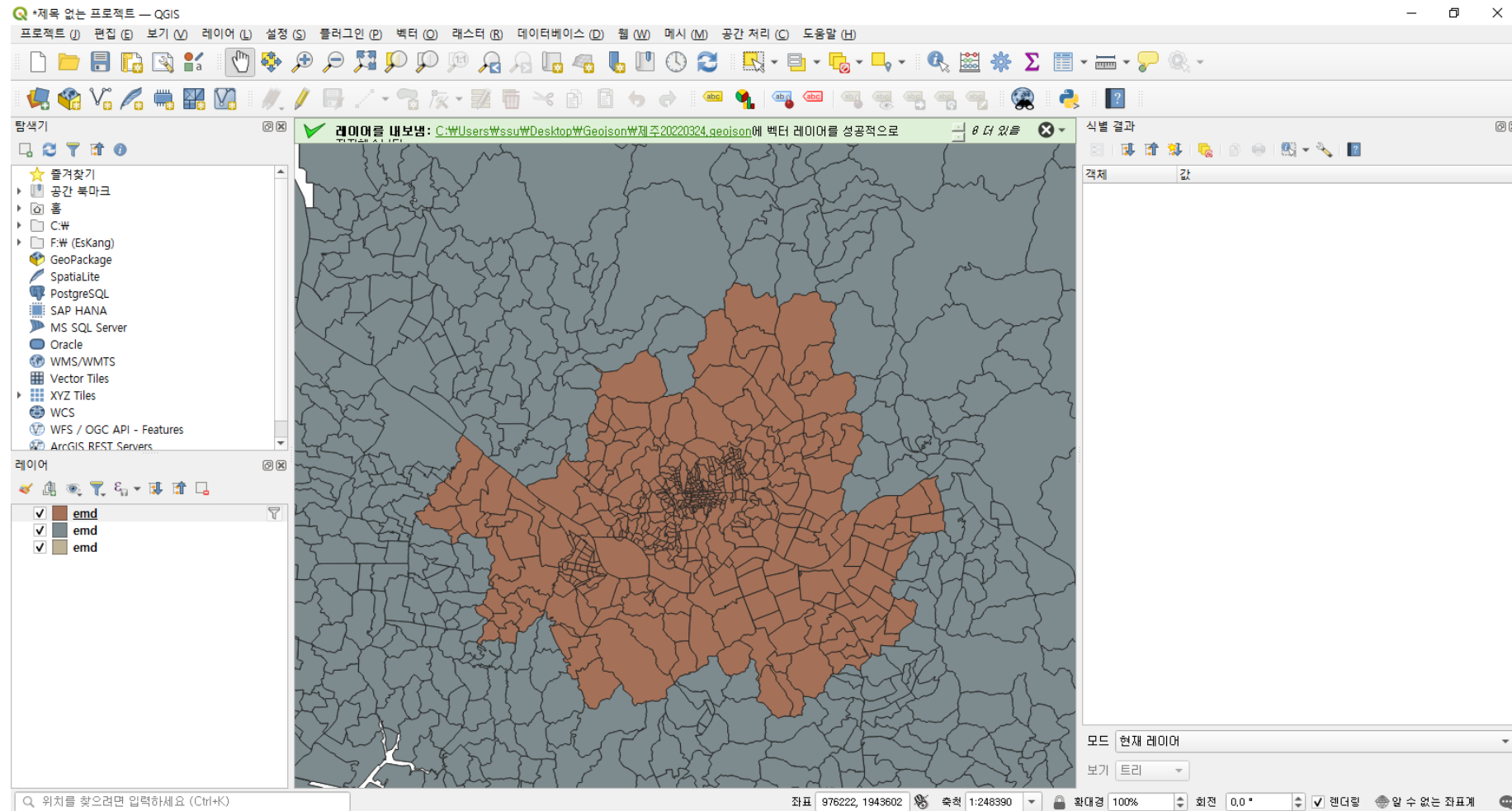
모든 파일 선택



원하는 지역만 추출하기

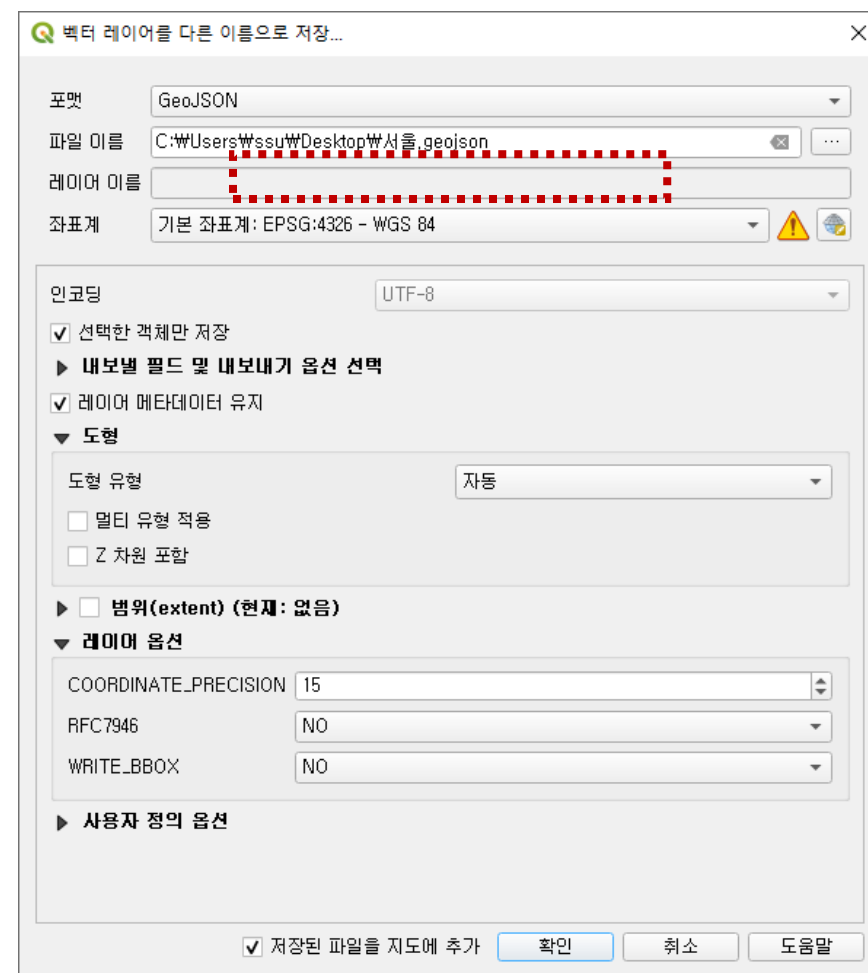
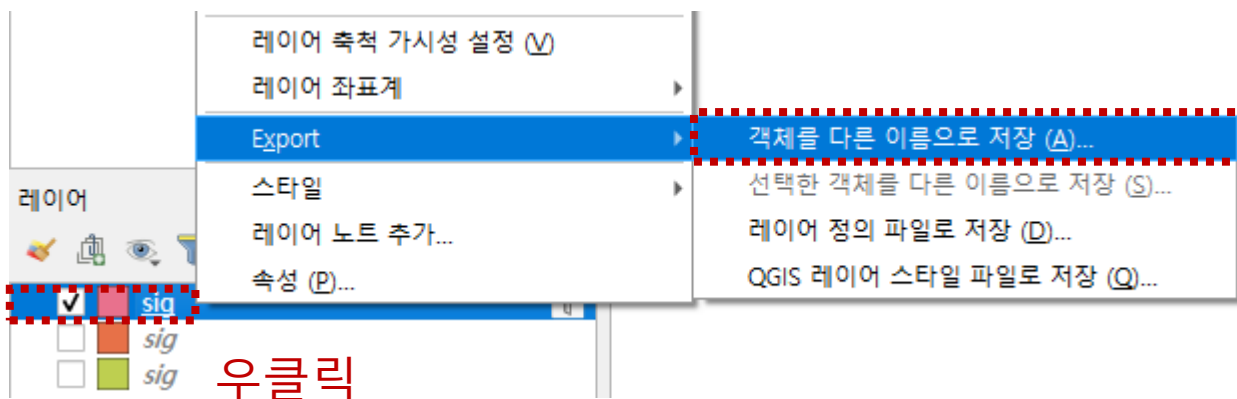


추출된 지역 확인하기



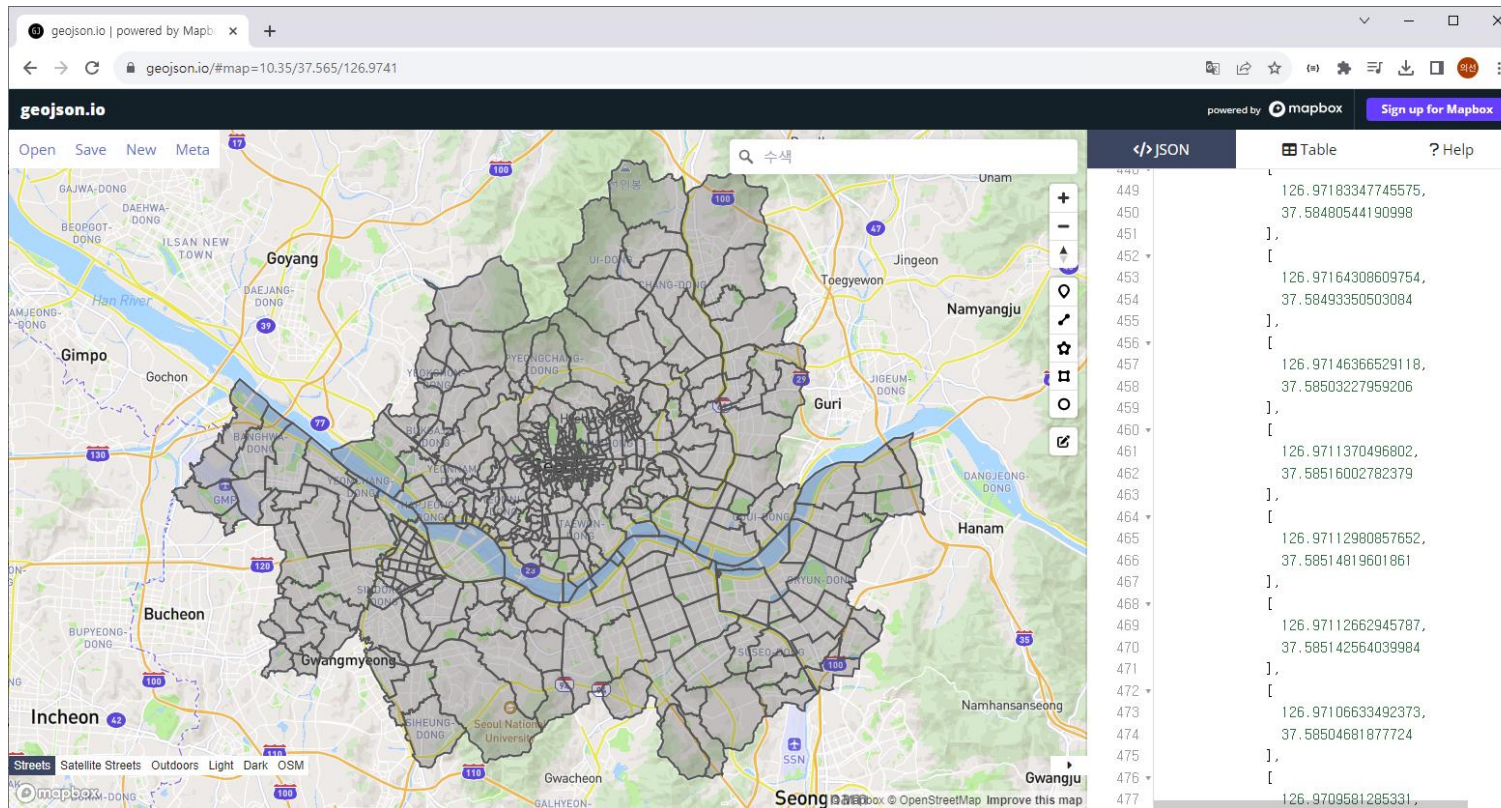
GeoJson으로 저장하기

• Geojson 으로 저장하기



저장된 GeoJson 확인하기

- <https://geojson.io/#map=2/0/20>



수고하셨습니다.