AI와 데이터 기초 II

지도 시각화

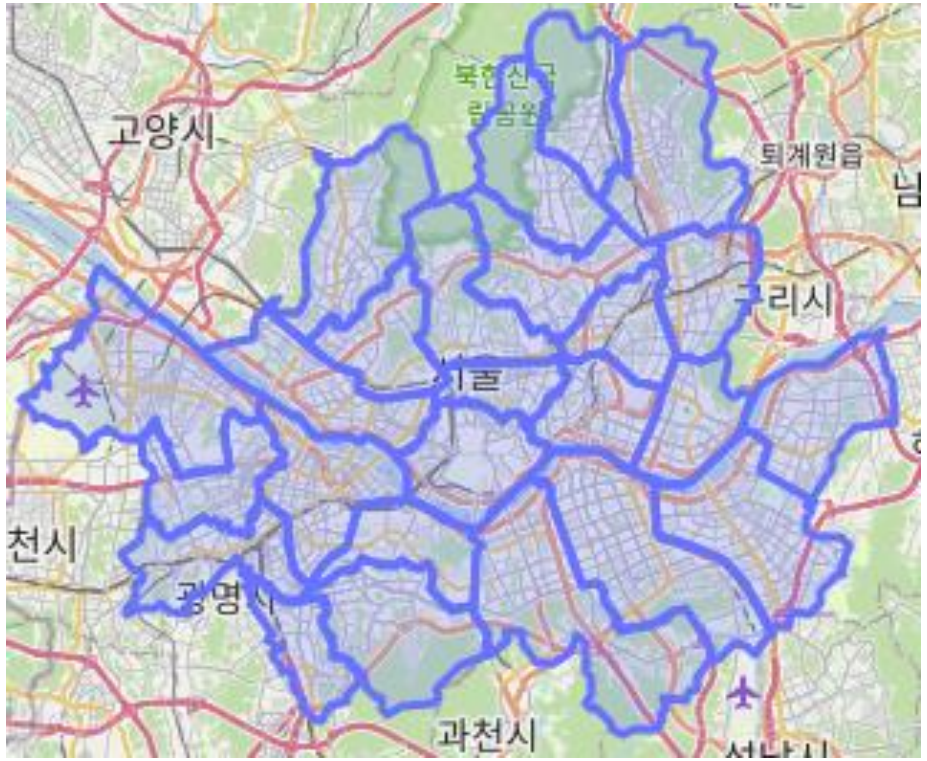
오늘 수업은

- GeoJson과 단계구분도 이해하기
- 지도에 경계선 표시하기
- 데이터에 따른 단계구분도 추가하기
- [실습 내용]
  - 서울시 자동차 등록대수 단계구분도로 표시하기
- SHP 파일과 Geojson 이해하기
- [실습 내용]
  - SHP 파일에서 원하는 지역을 Geojson으로 바꾸기

GeoJson

출처 : https://ko.wikipedia.org/wiki/GeoJSON

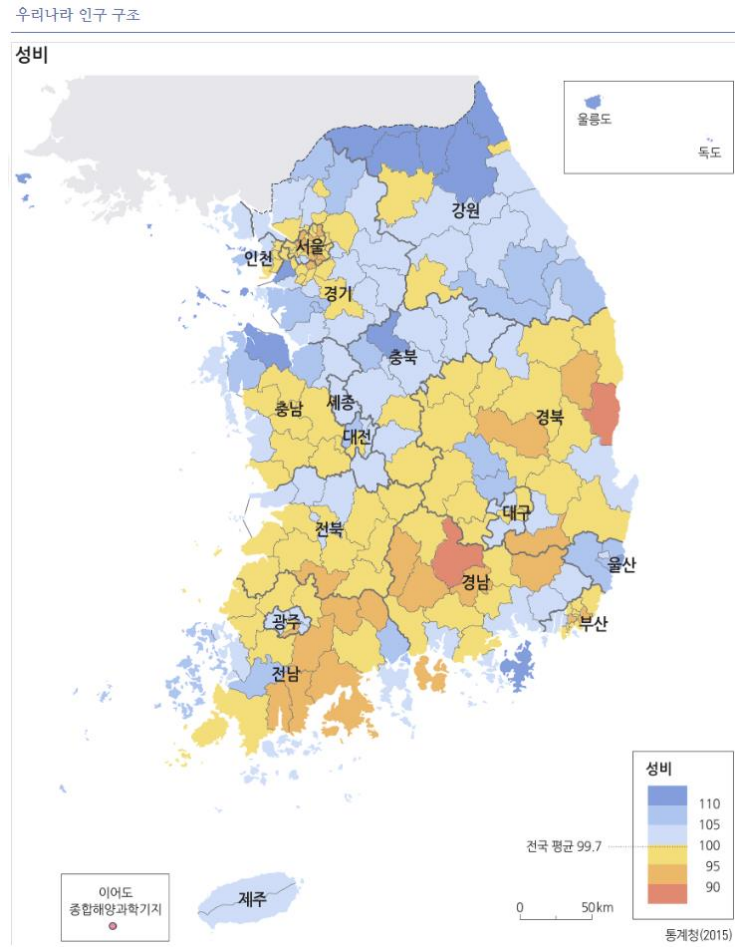
- 위치 정보를 기반으로 지형을 표현하기 위해 설계된 개방형 공개 표준 형식
- 행정구역과 같은 경계선이나 경로를 표현하는데 활용
- 위치 정보가 (경도, 위도, (고도)) 순서로 저장됨
  - 구글앱이나 OSM에서는 (위도, 경도) 순으로 저장됨
- 확장자 : *.json, *.geojson
- 웹 또는 모바일에서 데이터 시각화 가능
- 다른 포맷에 비해 위치 데이터 용량이 적음



단계 구분도

❖단계 구분도(Choropleth Map)

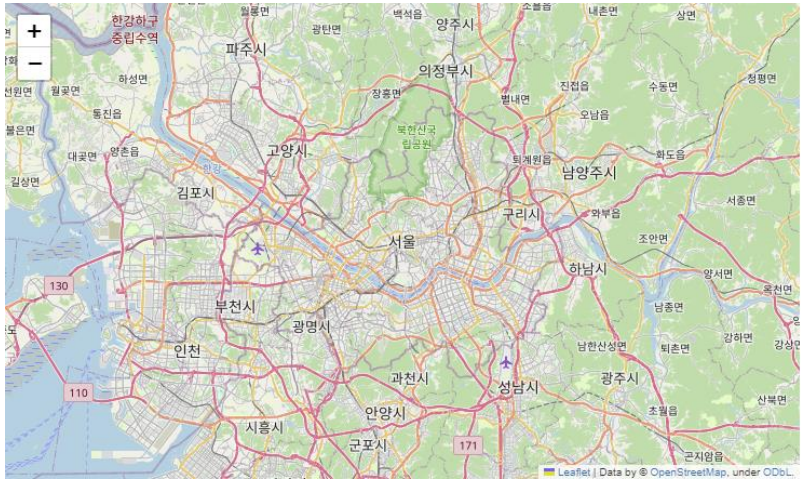
- 지역별간의 분포 차이 및 통계치를 색깔이나 그라데이션으로 구분하여 표현한 지도
- 수치적인 데이터를 색상으로 단계적으로 표현한 지도



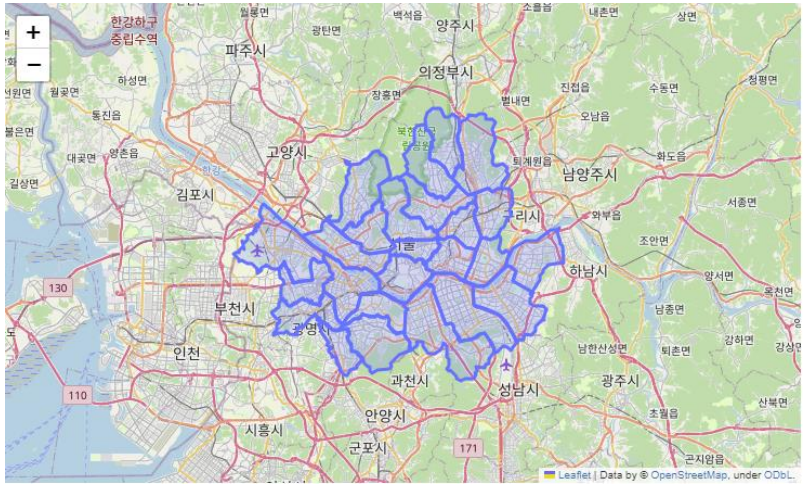
사진출처: http://nationalatlas.ngii.go.kr/pages/page_1361.php

GeoJson과 단계구분도를 이용한 시각화 순서

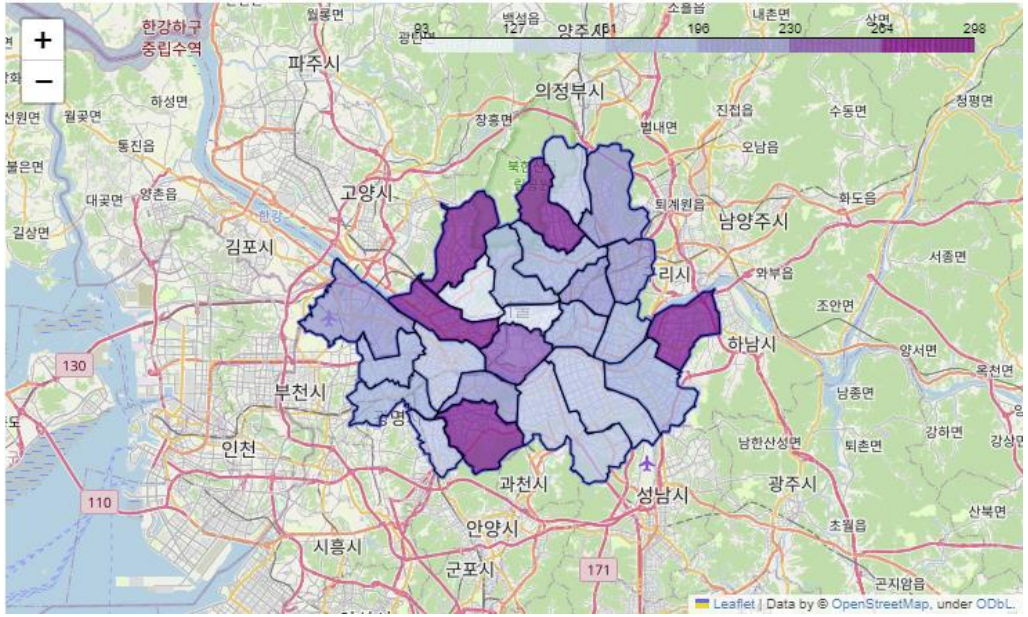
① folium을 활용하여 Map 그리기



② GeoJson으로 행정구역 경계 확인하기



③ Choropleth를 지도에 추가하기



GeoJson 파일 읽어오기

❖GeoJson 파일을 읽어들이기 위한 라이브러리 선언

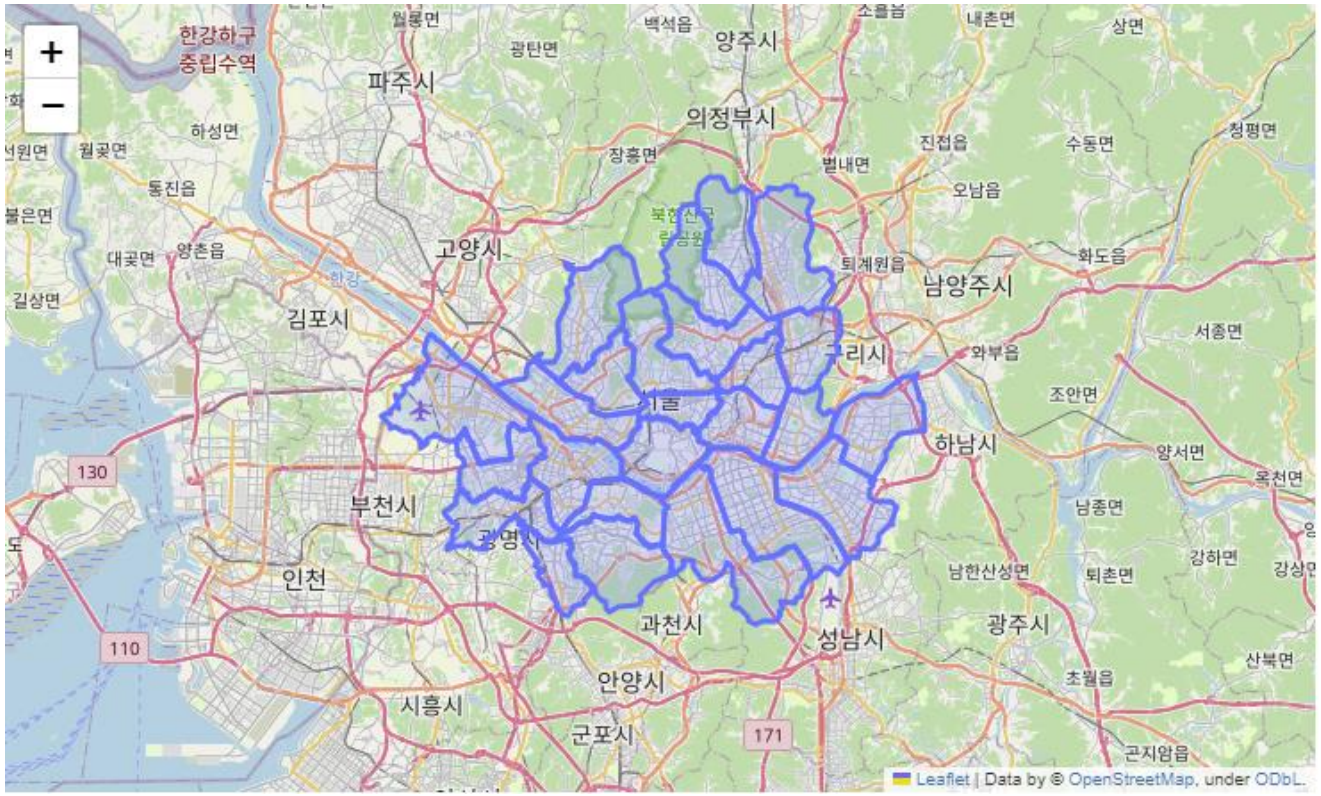
```
import json
```

❖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

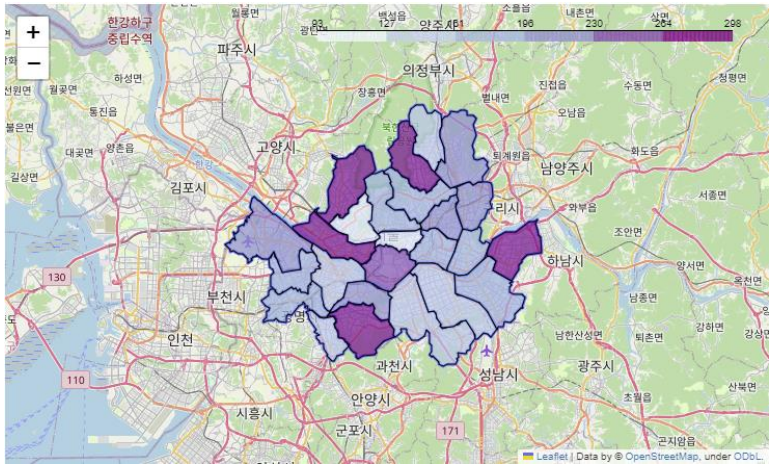
- GeoJson 좌표를 지도에 추가하기
  - folium.GeoJson(Json변수명).add_to(지도변수명)
    - GeoJson(Json변수명) : GeoJson 데이터에서 좌표들을 지도에 표현



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

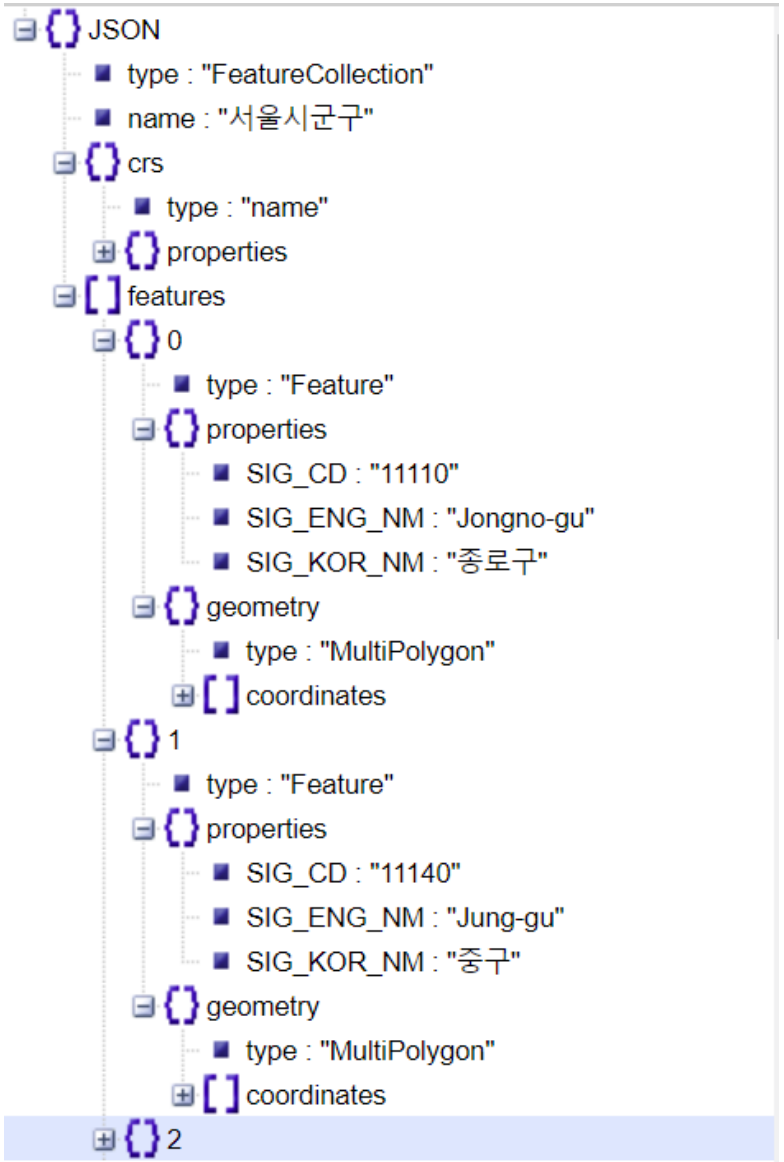


단계 구분도를 표현하기 위한 준비물

- 서울시 시군구별 자동차 등록대수에 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)
    - 자동차_등록대수(2022년_서울).xlsx

행정구역 경계를 나타내는 지도 데이터

❖서울시군구.geojson

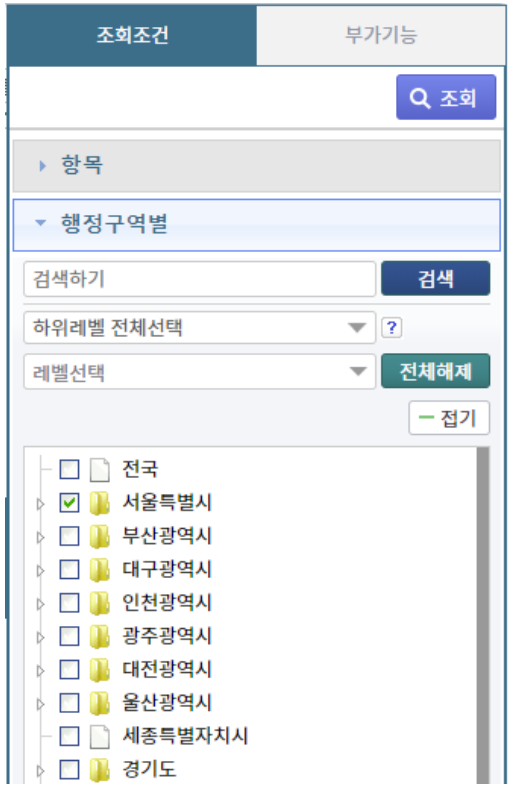


서울시 자동차 등록대수
단계구분도로 표시하기

서울시 1인당 자동차 등록대수

❖2022년 서울시 자동차 등록대수

- https://kosis.kr/statHtml/statHtml.do?orgId=101&tblId=DT_1YL20731&conn_path=I2
- 조회설정에서 서울시 시군구만 선택
- 자동차_등록대수(2022년_서울).xlsx



2) 행정구역별	2022		
	1인당 자동차등록대수 (A÷B) (대)	자동차등록대수 (A) (대)	주민등록인구 (B) (명)
서울특별시	0.3	3,193,351	9,428,372
종로구	0.4	50,337	141,379
중구	0.5	58,836	120,437
용산구	0.3	75,505	218,650
성동구	0.4	104,434	281,000
광진구	0.3	98,535	337,416
동대문구	0.3	99,447	336,644
중랑구	0.3	116,094	385,318
성북구	0.3	122,644	430,397
강북구	0.3	75,180	293,660
도봉구	0.3	95,690	311,694
노원구	0.3	152,529	503,734
은평구	0.3	133,758	466,746
서대문구	0.3	90,932	306,337
마포구	0.3	122,018	364,638
양천구	0.3	151,846	440,881
강서구	0.4	207,536	569,166
구로구	0.4	148,659	395,315
금천구	0.4	91,676	229,642
영등포구	0.4	144,596	375,675
동작구	0.3	106,037	380,596
관악구	0.2	118,615	486,752
서초구	0.4	176,799	404,325
강남구	0.5	248,320	529,102
송파구	0.4	250,142	658,801
강동구	0.3	153,186	460,067

자동차 등록 데이터 읽고 확인하기

```
#자동차_등록대수(2022년_서울).xlsx 화일을 읽고 상위 5개 데이터 확인하기
#변수명 = pd.read_excel( '파일경로명, header=[행번호])
#header=[행번호]: 열이름으로 설정할 행번호 기술
#변수명.head()

import pandas as pd

carData = pd.read_excel('/content/drive/MyDrive/Colab Notebooks/1일차/자동차_등록대수(2022년_서울).xlsx',
                        header=[1])
carData.head()
```

	행정구역별	1인당 자동차등록대수 (대)	자동차등록대수(대)	주민등록인구(명)
0	서울특별시	0.3	3193351	9428372
1	종로구	0.4	50337	141379
2	중구	0.5	58836	120437
3	용산구	0.3	75505	218650
4	성동구	0.4	104434	281000

자동차 등록 데이터 읽고 확인하기

```
1 ##행열개수, 데이터 타입, 열의 개수 등 데이터 정보 확인하고 folium.Choropleth의 속성중 columns=[] 선택하기
2 #변수명.info()
3
4 carData.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26 entries, 0 to 25
Data columns (total 4 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   행정구역별            26 non-null     object
 1   1인당 자동차등록대수 (대)  26 non-null     float64
 2   자동차등록대수(대)       26 non-null     int64
 3   주민등록인구(명)        26 non-null     int64
dtypes: float64(1), int64(2), object(1)
memory usage: 960.0+ bytes
```

지도 데이터 읽고 확인하기

❖GeoJson 파일 읽어오기

```
#json 라이브러리 선언하기
#import json
import json

#GeoJson화일을 열고 데이터 읽어오기
#변수명 = json.load((open('json파일경로명')))
geoSeoul = json.load(open('/content/drive/MyDrive/Colab Notebooks/1일차/서울시군구.geojson'))
geoSeoul
```


```
{'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],
```

지도 데이터 읽고 확인하기

❖GeoJson 파일에서 시군구 이름을 확인할 수 있는 키워드 확인하기

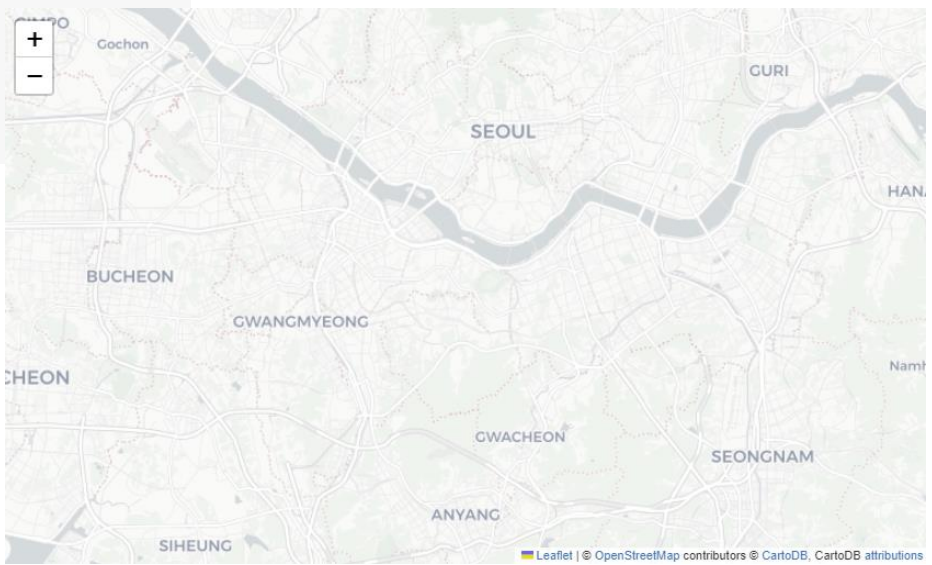
```
1 #변수명[''][0]['']
2
3 geoSeoul['features'][0]['properties']
```

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

지도 생성하기

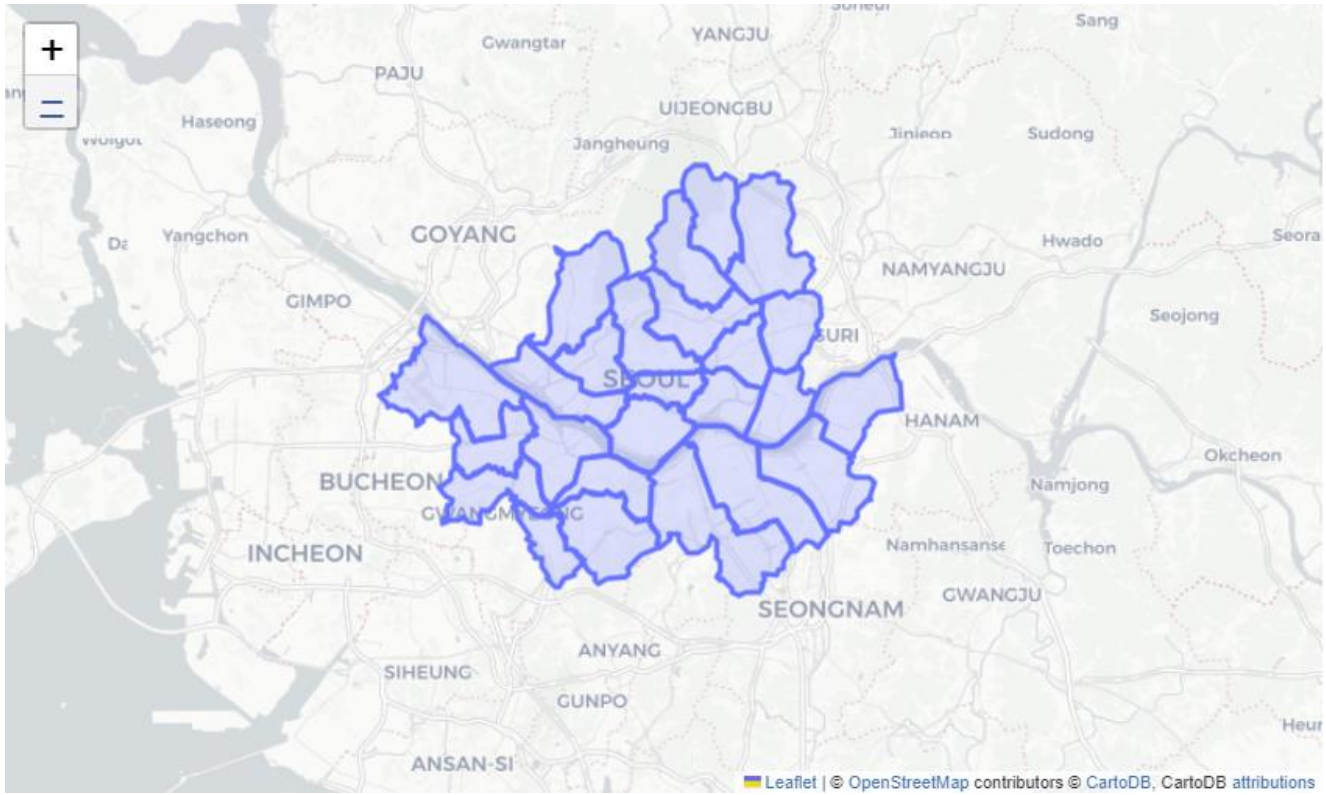
```
#지도를 생성하기 위한 folium 라이브러리 선언하기
#import folium
import folium


#folium.Map()을 사용하여 지도 생성하고 출력하기
#변수명 = folium.Map(속성들)
#location=[위도, 경도] : 지도의 중심 좌표를 [위도, 경도] 또는(위도, 경도)로 나타냄
#zoom_start = 정수 : 지도를 처음 그릴때 확대 정도
#zoom_control=True : zoom in/out 버튼 표시 여부(True/False)
#control_scales=False : 스케일 컨트롤 버튼 표시 여부(True/False)
#tiles= '스타일' : 지도 스타일을 지정(openstreetmap,stamenterrain,cartodbpositron)
map = folium.Map(location=[37.607,127.0424],    #동덕여자대학교 중심좌표
                 zoom_start=11,                 #초기 확대 정도
                 tiles='cartodbpositron')       #지도 스타일
map
```



GeoJson을 지도에 추가하여 확인하기

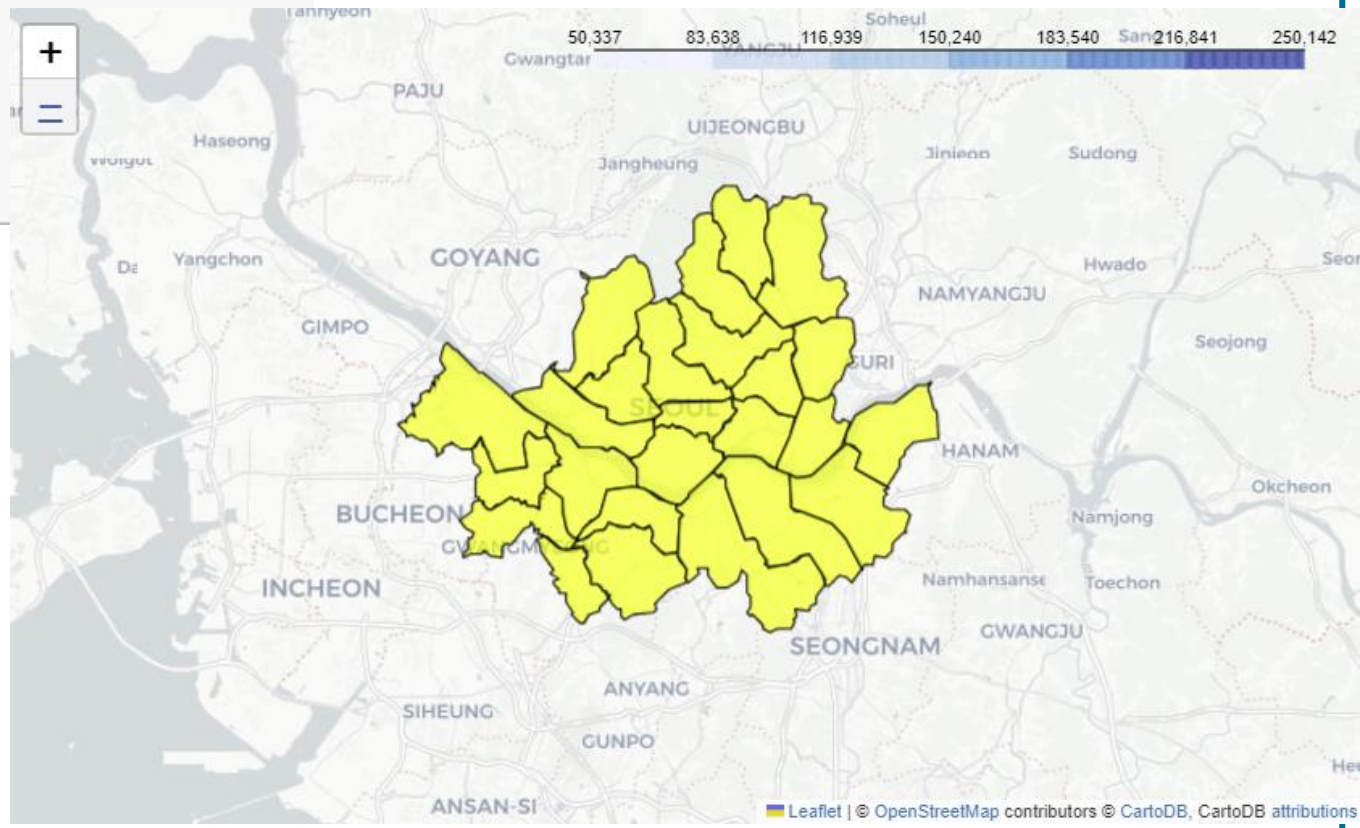
```
1 #folium.GeoJson(변수명).add_to(지도변수명)
2 folium.GeoJson(geoSeoul).add_to(map)
3 map
```



단계구분도를 생성하기

❖단계구분도를 생성하고 지도에 추가하기

```
folium.Choropleth(geo_data = geoSeoul,
                  data=carData[1:],
                  columns=['행정구역별', '자동차등록대수(대)'],
                  key_on = 'feature.properties.SIG_KOR_NM',
                  fill_color='Blues',
                  nan_fill_color = 'yellow'
                  ).add_to(map)
map
```



단계구분도를 생성하기

❖문제 확인하고 해결하기

- columns=[´행정구역별´, ´자동차등록대수(대)´]와 key_on = ´feature.properties.SIG_KOR_NM´,에 일치하는 데이터가 없음으로 모든 색상이 동일한 색상(nan_fill_color)로 표시됨

```
#carData['행정구역별']의 첫번째 인덱스 데이터 확인하기
#변수명['열이름'][index번호]
carData['행정구역별'][1]
```

'₩u3000₩u3000₩u3000종로구'

불필요한 문자가 삽입됨

```
#불필요한 일부문자열 삭제하기
#df[ '열이름' ] = df[ '열이름' ].str.replace( '찾을 문자열' , '변경할 문자열' )
#   열이름: 부분 변경할 문자열이 있는 열이름

carData['행정구역별']= carData['행정구역별'].str.replace('₩u3000₩u3000₩u3000', '')
carData['행정구역별'][1]
```

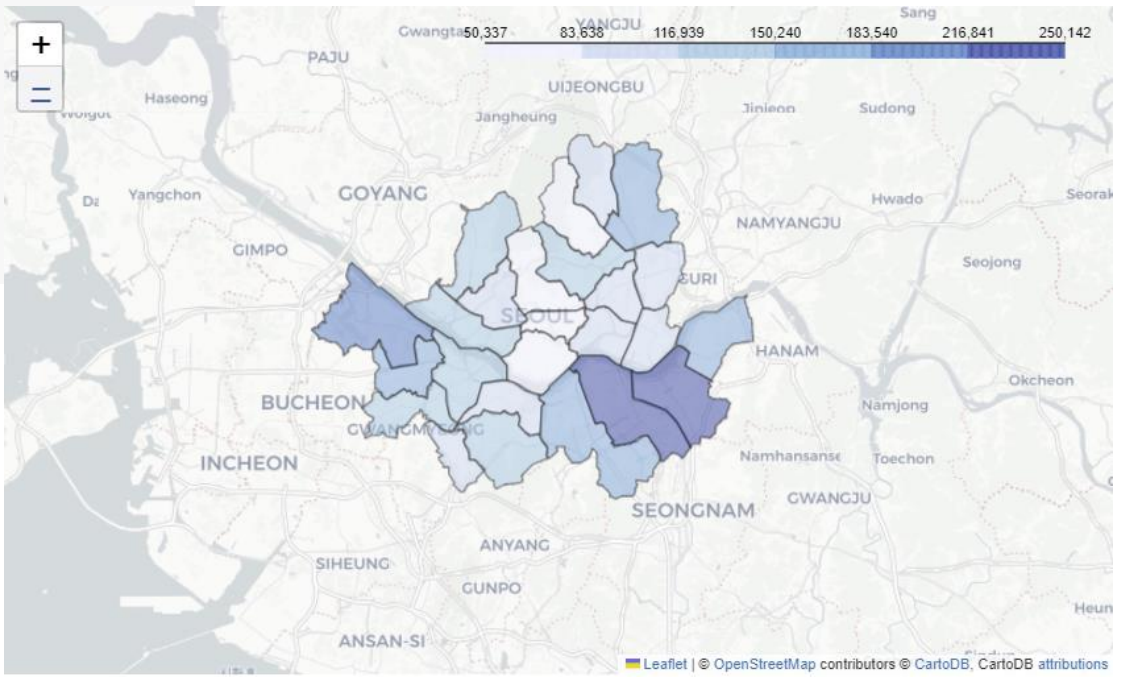
'종로구'

단계구분도를 생성하기

❖단계구분도 재실행하고 추가 옵션 설정하기

```
map = folium.Map(location=[37.607,127.0424],    #동덕여자대학교 중심좌표
                 zoom_start=11,                 #초기 확대 정도
                 tiles='cartodbpositron')       #지도 스타일

folium.Choropleth(geo_data = geoSeoul,
                  data=carData[1:],
                  columns=['행정구역별', '자동차등록대수(대)'],
                  key_on = 'feature.properties.SIG_KOR_NM',
                  fill_color='Blues',
                  nan_fill_color = 'yellow',
                  fill_opacity=0.5,
                  line_opacity=0.5
                  ).add_to(map)
map
```



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 ㈜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

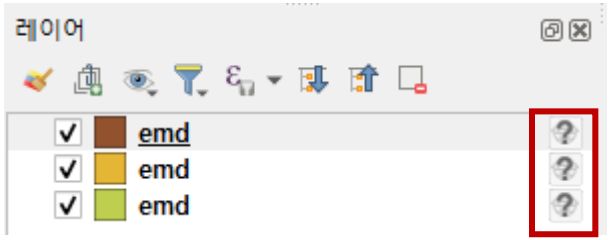
SHP를 Geojson 으로 변경하는 순서

- 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

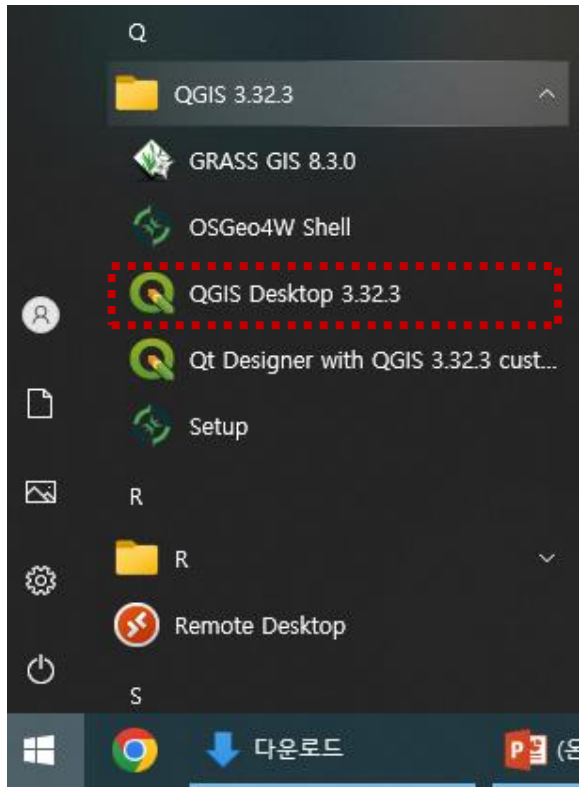
2023년 7월 업데이트 다운로드
2023년 2월 업데이트 다운로드
2022년 11월 업데이트 다운로드
2022년 3월 업데이트 다운로드
2021년 1월 업데이트 다운로드
2020년 5월 업데이트 다운로드
2019년 5월 업데이트 다운로드
2019년 2월 업데이트 다운로드
2018년 4월 업데이트 다운로드
2017년 3월 업데이트 다운로드
2016년 2월 업데이트 다운로드
2015년 6월 업데이트 다운로드
2014년 7월 업데이트 다운로드
2014년 5월 업데이트 다운로드
2013년 11월 업데이트 다운로드

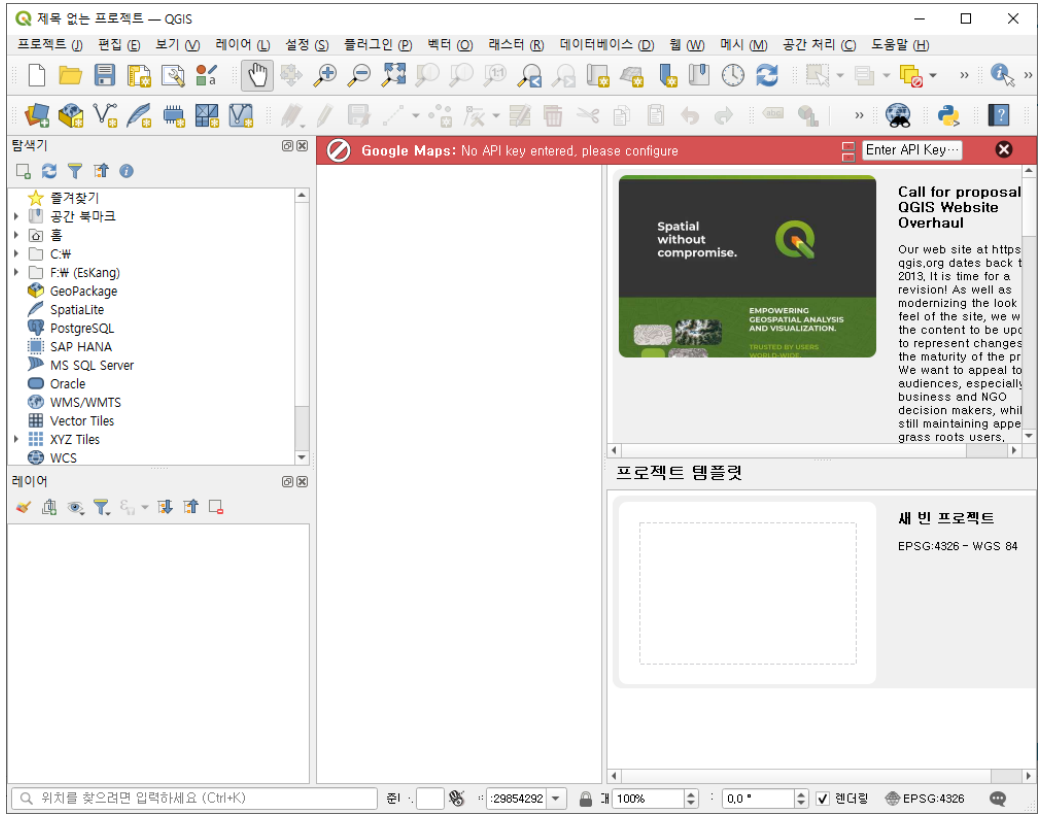


벡터레이어 추가시 '?'가 보이면 좌표계 변경이 안됨

QGIS 설치하기

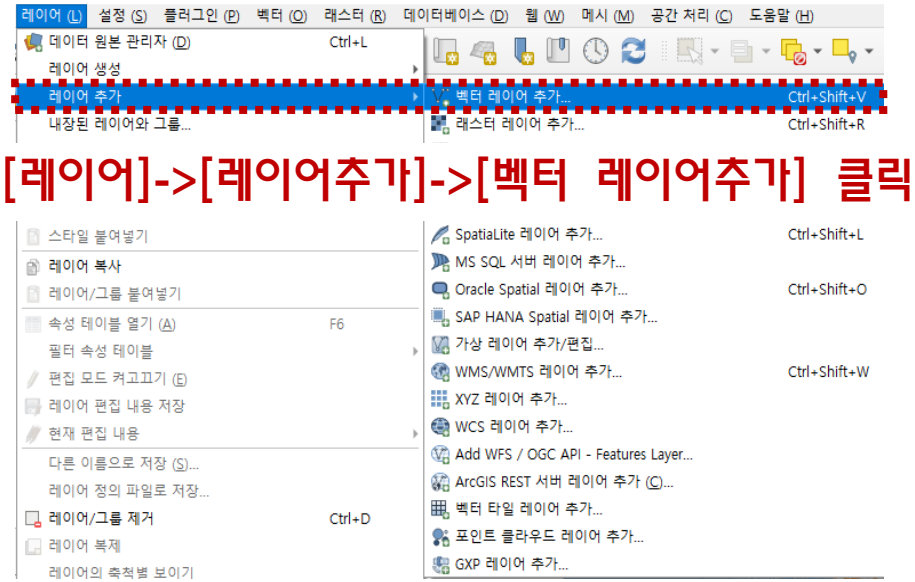
1. 본인의 PC 운영체제에 맞는 설치 파일 다운로드하기
   https://www.qgis.org/ko/site/forusers/download.html#
2. 설치하기
3. QGIS 실행하기



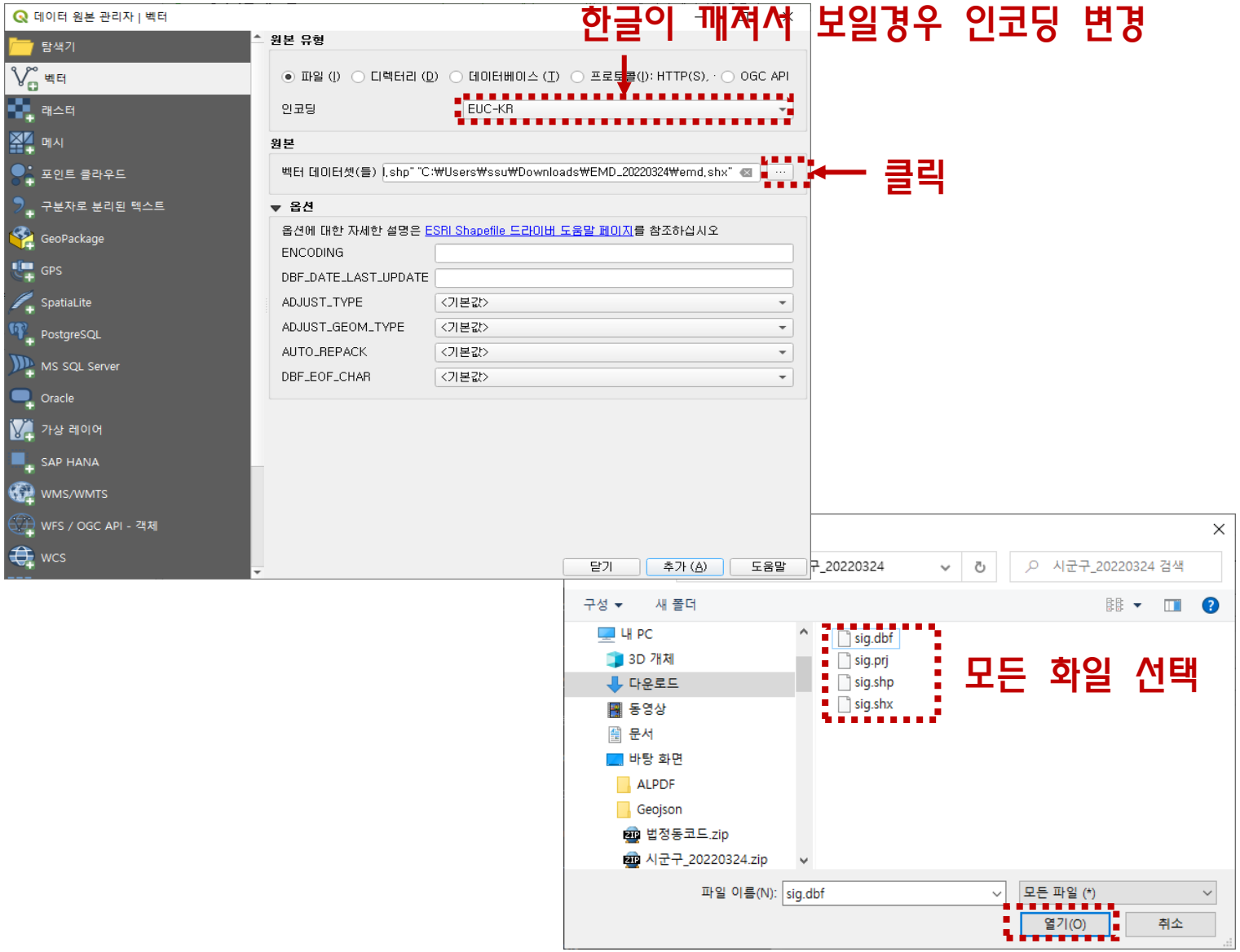


QGIS에서 SHP 파일 읽어오기

❖QGIS에서 SHP 파일 읽어오기



[레이어]->[레이어추가]->[벡터 레이어추가] 클릭

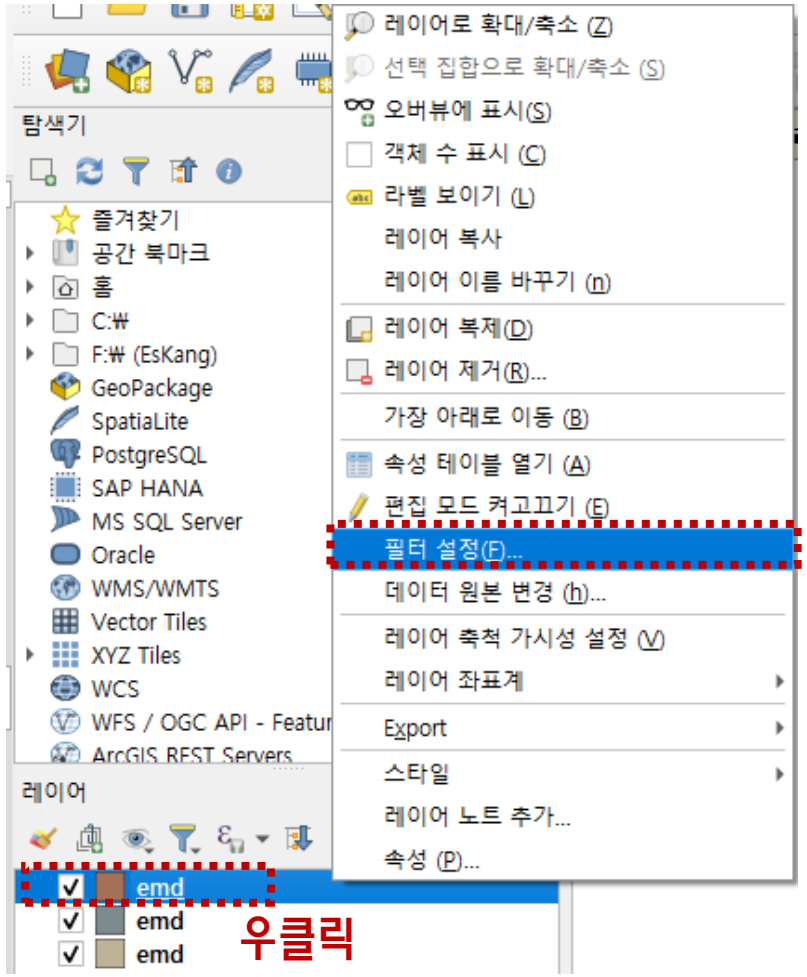


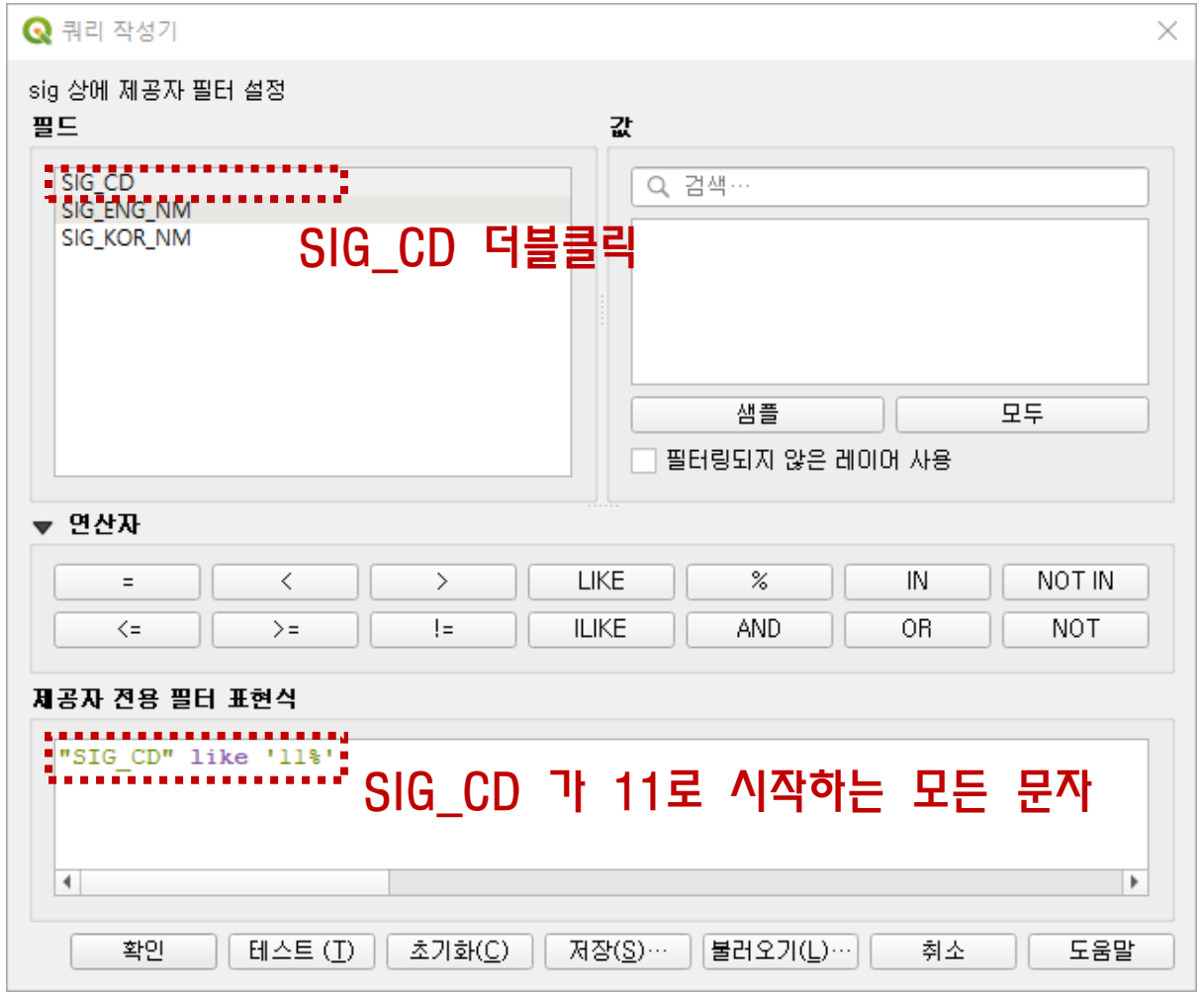
한글이 깨져서 보일경우 인코딩 변경

← 클릭

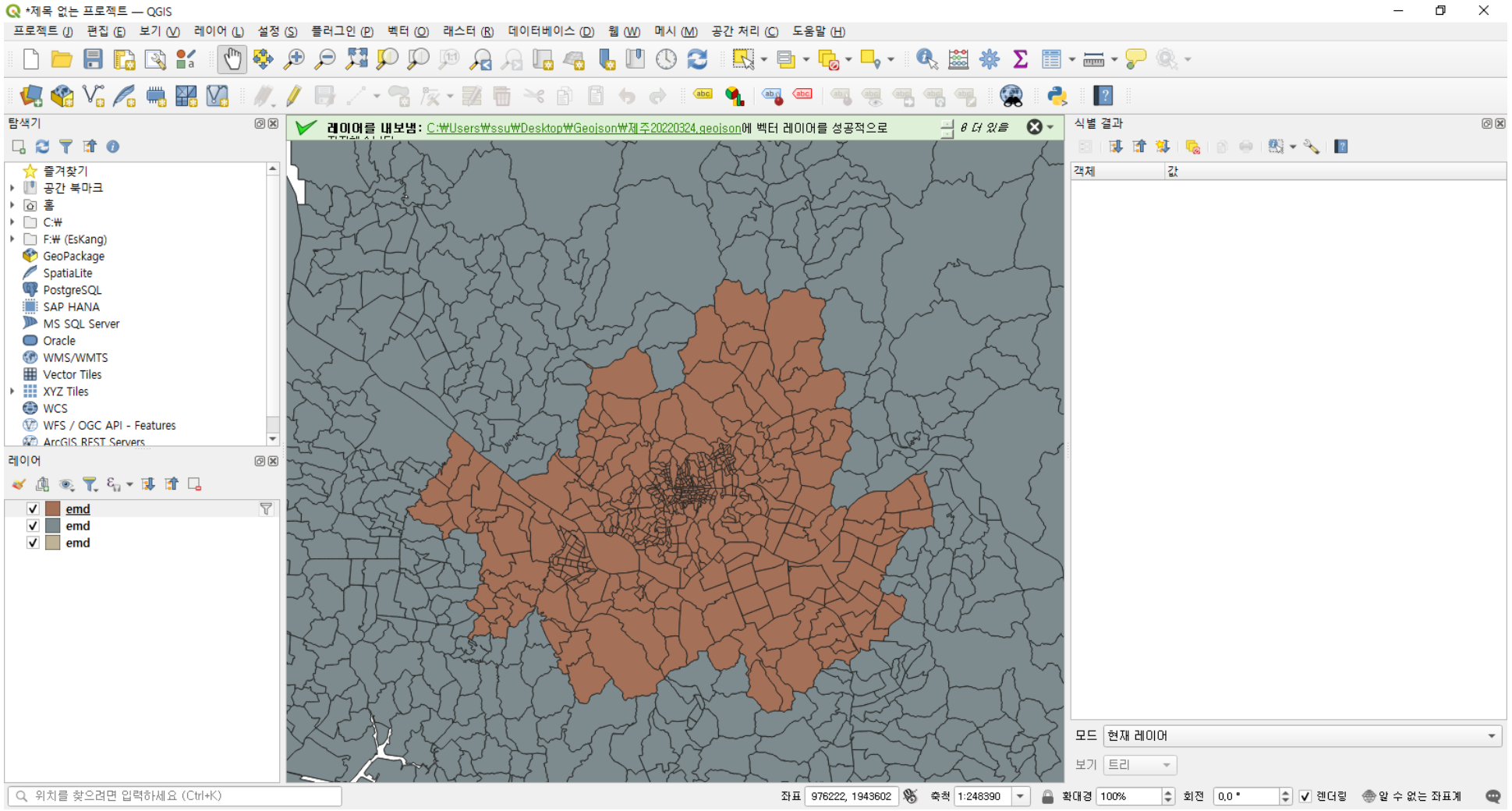
모든 화일 선택

원하는 지역만 추출하기


탐색기
즐겨찾기
공간 북마크
홈
C:₩
F:₩ (EsKang)
GeoPackage
SpatiaLite
PostgreSQL
SAP HANA
MS SQL Server
Oracle
WMS/WMTS
Vector Tiles
XYZ Tiles
WCS
WFS / OGC API - Featur
ArcGIS REST Servers
레이어
emd
emd
emd
우클릭
레이어로 확대/축소 (Z)
선택 집합으로 확대/축소 (S)
오버뷰에 표시(S)
객체 수 표시 (C)
라벨 보이기 (L)
레이어 복사
레이어 이름 바꾸기 (n)
레이어 복제(D)
레이어 제거(R)...
가장 아래로 이동 (B)
속성 테이블 열기 (A)
편집 모드 켜고끄기 (E)
필터 설정(F)...
데이터 원본 변경 (h)...
레이어 축척 가시성 설정 (V)
레이어 좌표계
Export
스타일
레이어 노트 추가...
속성 (P)...

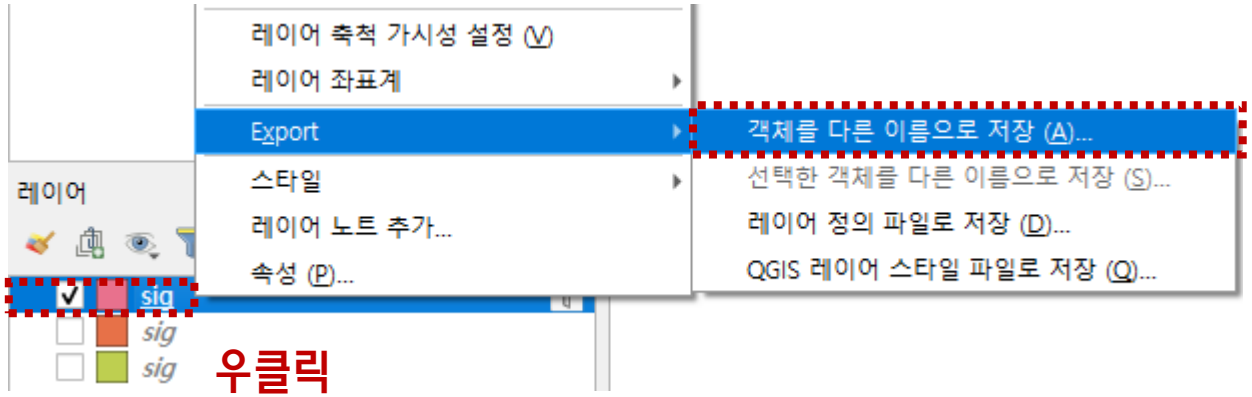

쿼리 작성기
sig 상에 제공자 필터 설정
필드
값
SIG_CD
SIG_ENG_NM
SIG_KOR_NM
SIG_CD 더블클릭
검색…
샘플
모두
필터링되지 않은 레이어 사용
연산자
=
<
>
LIKE
%
IN
NOT IN
<=
>=
!=
ILIKE
AND
OR
NOT
제공자 전용 필터 표현식
"SIG_CD" like '11%'
SIG_CD 가 11로 시작하는 모든 문자
확인
테스트 (T)
초기화(C)
저장(S)…
불러오기(L)…
취소
도움말

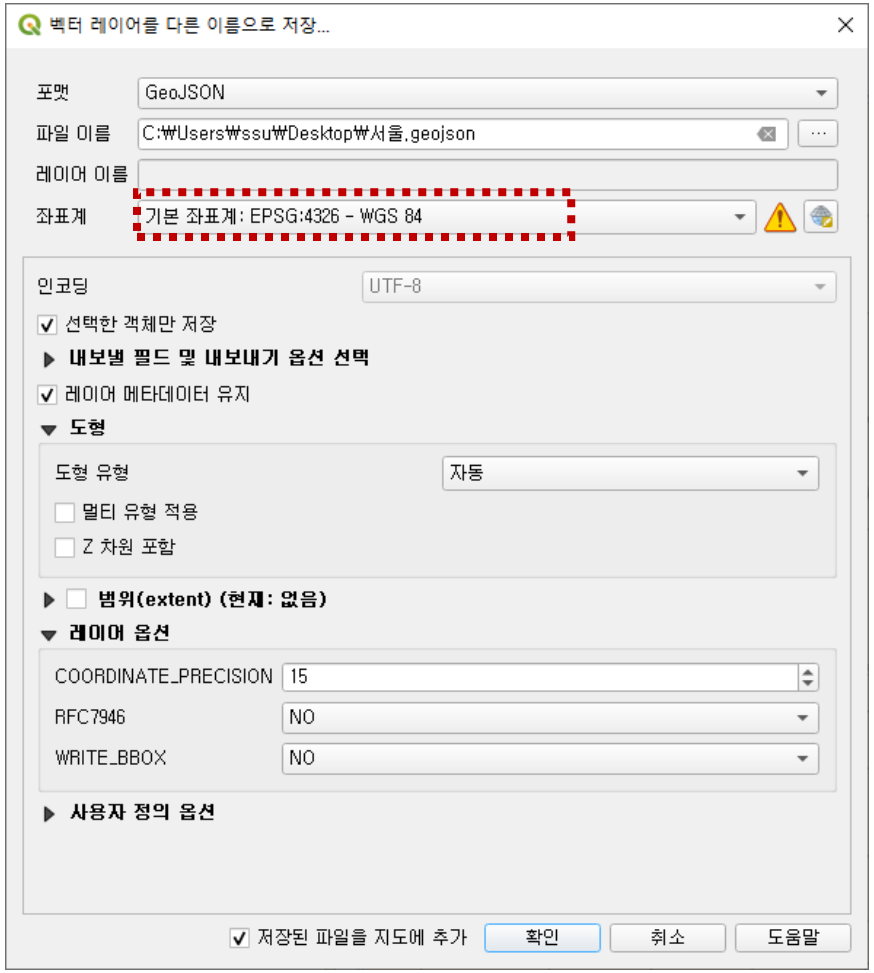
추출된 지역 확인하기


*제목 없는 프로젝트 — QGIS
프로젝트 (J) 편집 (E) 보기 (V) 레이어 (L) 설정 (S) 플러그인 (P) 벡터 (O) 래스터 (R) 데이터베이스 (D) 웹 (W) 메시 (M) 공간 처리 (C) 도움말 (H)
탐색기
즐겨찾기
공간 북마크
홈
C:₩
F:₩ (EsKang)
GeoPackage
SpatiaLite
PostgreSQL
SAP HANA
MS SQL Server
Oracle
WMS/WMTS
Vector Tiles
XYZ Tiles
WCS
WFS / OGC API - Features
ArcGIS REST Servers
레이어
emd
emd
emd
레이어를 내보냄: C:₩Users₩ssu₩Desktop₩Geojson₩제주20220324.geojson에 벡터 레이어를 성공적으로
식별 결과
객체
값
모드 현재 레이어
보기 트리
위치를 찾으려면 입력하세요 (Ctrl+K)
좌표 976222, 1943602
축척 1:248390
확대경 100%
회전 0,0 °
렌더링
알 수 없는 좌표계

GeoJson으로 저장하기

❖Geojson 으로 저장하기





저장된 GeoJson 확인하기

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