작성일: 2023-03-16

중요! 자전거 정류장 및 집계구의 좌표계를 (EPSG 5179)로 바꿔 사용하는 것이 중요함

1. 목적: 인구통계 정보를 전처리 하는 전반적인 과정에 대해서 설명하고자함

2. 데이터 다운로드

1) 인구통계 데이터 및 집계구 데이터의 경우, 아래 sgis에 가입후 데이터 다운로드 신청을 해서 받음

- https://kosis.kr/oneid/cmmn/login/LoginView.do

- 집계구 데이터의 경우, shp 파일이 존재함. 좌표계가 맞지 않아서, QGIS에 업로드 하고, 좌표 바꾼 후 내보내기함

- 인구 통계 데이터의 경우, txt 파일이며, 칼럼이 따로 존재하지 않음. 그래서 그냥 txt 파일 켜서, 칼럼 넣어줌

2) 타슈 위치 정보

- 나스 서버에서 정주현 학생이 업로드 해둔 정보를 활용함

-ftp://dmfile.ipdisk.co.kr:2222/HDD2/Data/Project/%BF%AC%B1%B8%C0%E7%B4%DC/Bike%20Sharing/Data/1\_Bike%20Sharing/Daejeon%20Bike/

3. 주요 프로세스

1) 데이터 로드

- 집계구 경계선 (shp) 파일을 로드함 (EPSG 5179)

- 각 통계자료를 로드함

2) 집계구 경계선 정보에서 polygon 객체만 추출함

3) 각 정류소를 나타내는 point에 버퍼를 먹이고(500m), (a. 영역이 겹치는 집계구 인덱스), (b.겹치는 영역의 비중(해당 집계구내)), (c.인덱스와 집계구 pair 정보)를 산출함

- 집계구 인덱스와 집계구 번호는 다름

- 버퍼를 먹일때는 projection을 해야함

- a. b. c. 는 딕셔너리 형태 >> b.의 경우 집계구 인덱스: 영역의 비중, c. 의 경우 집계구 인덱스:집계구번호

4) 정보 추출 (get\_pop\_info, get\_pop\_info1)

5) 추출된 정보를 나타내는 데이터 프레임의 칼럼명은 따릉이 데이터의 칼럼명과 동일하게 설정함

4. 정보추출과정에 대한 설명

1) (자전거 정류소별 진행) 통계자료[데이터프레임]에서, 겹치는 집계구 번호를 기준으로 인덱싱함

- 이때 집계구 번호와 통계값 칼럼만 사용

2) 인덱싱된 데이터프레임을 대상으로, 집계구 번호를 기준 groupby 집계구 번호를 하고 sum 연산을 진행

- 같은 집계구에 대한 정보가 여러개 나타날 수 있기 때문임

3) 이렇게 되면, 집계구번호 별 sum된 통계값을 알 수 있음 >> 해당 정보를 가지고 딕셔너리 생성

- 집계구 번호를 키, 통계값을 value로

4) 이후 집계구별 sum된 통계값 value를 겹치는 영역의 비중과 곱해주고 더함

- <b의 키로 인덱싱한 c의 value>를 3)에서 생성한 딕셔너리의 key로 사용 >> value는 sum한 통계값, 이를 b.의 value와 곱하는 것이 weighted summation

- 경우에 따라, 통계값이 존재하지 않을 수 있는데, 이 경우는 0을 더함

5. 주요 코드

1) 정류소별 반경내에 위치하는 집계구 관련 정보 산출

bin\_dict = {}

intersect\_dict = {}

new\_bin\_bin\_dict = {}

for xidx, (xx, yy) in enumerate(station\_address[['Yn','Xn']].values):

uni\_buffered\_point = Point((yy,xx)).buffer(500)

uni\_buffered\_point()

its\_idx = [xx for xx,yy in enumerate(shp\_list) if yy.intersects(uni\_buffered\_point)]

bin\_dict[xidx+1] = its\_idx

new\_bin\_list = {}

new\_bin\_dict = {}

for uni\_its in its\_idx:

new\_bin\_list[uni\_its] = (shp\_list[uni\_its].buffer(0).intersection(uni\_buffered\_point)).area / (shp\_list[uni\_its].area)

new\_bin\_dict[uni\_its] = int(temp\_shp[uni\_its]['properties']['TOT\_REG\_CD'])

new\_bin\_bin\_dict[xidx+1] = new\_bin\_dict

intersect\_dict[xidx+1] = new\_bin\_list

2) 통계정보 산출

def get\_pop\_info(idx\_dic, census\_weight\_dic, census\_idx\_dic, static\_df):

base\_dict = {}

for (uni\_key1, uni\_val1), (uni\_key2, uni\_val2), (uni\_key3, uni\_val3) in zip(idx\_dic.items(), census\_weight\_dic.items(),

census\_idx\_dic.items()):

newa1= (static\_df.loc[static\_df.num\_idx.isin(uni\_val3.values())])[['num\_idx','target']]

newa2 = newa1.groupby('num\_idx').sum()

temp\_dict1 = dict(zip(newa2.index.values, newa2['target']))

temp\_zero = 0

for u\_uni\_key, u\_uni\_val in uni\_val2.items():

temp\_zero+=temp\_dict1[uni\_val3[u\_uni\_key]] \* u\_uni\_val

base\_dict[uni\_key1] = temp\_zero

return base\_dict