

지도 시각화 II

(GeoJson과 Choropleth 활용)

최수연 교수

mibm400@hanmail.net

공공데이터 파일 읽어오기

- 변수명 = `pd.read_excel('파일경로명', 속성들)`
- ✓ `header=[행번호]` :
 - 위에서 몇번째 행부터 읽어올지 지정(행번호는 0부터 시작)
 - 열이름(변수이름)으로 설정할 index 번호 기술
- DataFrame에서 문자열 데이터의 일부분을 변경하고자 할때
- ✓ `df['열이름'] = df['열이름'].str.replace('찾을 문자열', '변경할 문자열')`
- ✓ 예) `df['단과대'] = df['단과대'].str.replace('인문', '인문사회')`
 - 열이름: 부분 변경할 문자열이 있는 열이름
 - 인문대학 -> 인문사회대학으로 변경하고자 할때 찾을 문자열은 인문, 변경할 문자열은 인문사회

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)

지도에 단계구분도 추가하기

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

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

조회조건

부가기능

Q 조회

항목

행정구역별

검색하기

검색

하위레벨 전체선택

?

레벨선택

전체해제

접기

☐

전국

☒

서울특별시

☐

부산광역시

☐

대구광역시

☐

인천광역시

☐

광주광역시

☐

대전광역시

☐

울산광역시

☐

세종특별자치시

☐

경기도

2) 행정구역별	2022		
	1인당 자동차등록 대수 (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

```
1 #자동차_등록대수(2022년_서울).xlsx 파일을 읽고 상위 5개 데이터 확인하기
2 #변수명 = pd.read_excel( '파일경로명, header=[행번호])
3 #header=[행번호]: 열이름으로 설정할 행번호 기술
4 #변수명.head()
5
6 import pandas as pd
7
8 carData = pd.read_excel('/content/drive/MyDrive/Colab Notebooks/10주차/자동차_등록대수(2022년_서울).xlsx',
9                          header=[1])
10 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

```
1 #json 라이브러리 선언하기
2 #import json
3 import json
4
5 #GeoJson화일을 열고 데이터 읽어오기
6 #변수명 = json.load((open('json파일경로명')))
7 geoSeoul = json.load(open('/content/drive/MyDrive/Colab Notebooks/10주차/서울시군구.geojson'))
8 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],
 [127.00916588888747, 37.58038371854548],
 [127.00916588888747, 37.58038371854548],
 [127.00913781377908, 37.58039352939572],
 [127.00890785297045, 37.580424231608646],
 [127.00876564011087, 37.580443107078565],
 [127.00871274905404, 37.58045116513156],
 [127.00864326221884, 37.580468252047105]]]]]]]
```

지도 데이터 읽고 확인하기

- GeoJson 파일에서 시군구 이름을 확인할 수 있는 키워드 확인하기

```
1 #변수명[''][0]['']
```

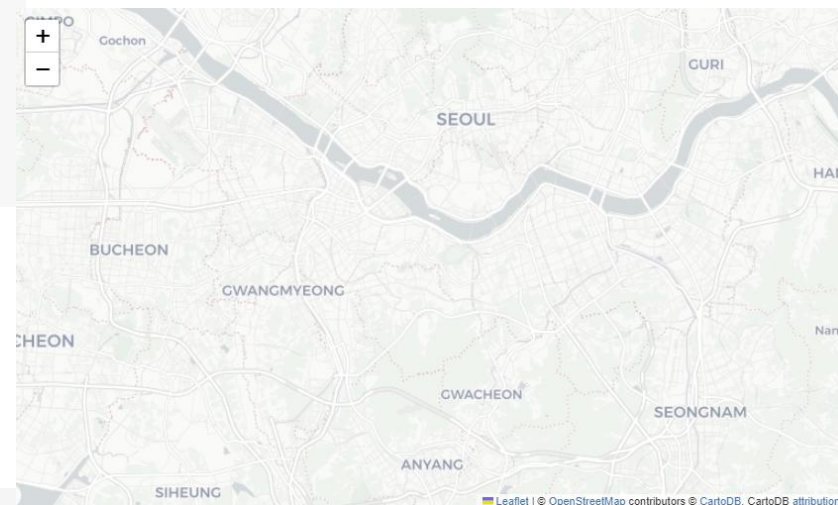
```
2
```

```
3 geoSeoul['features'][0]['properties']
```

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

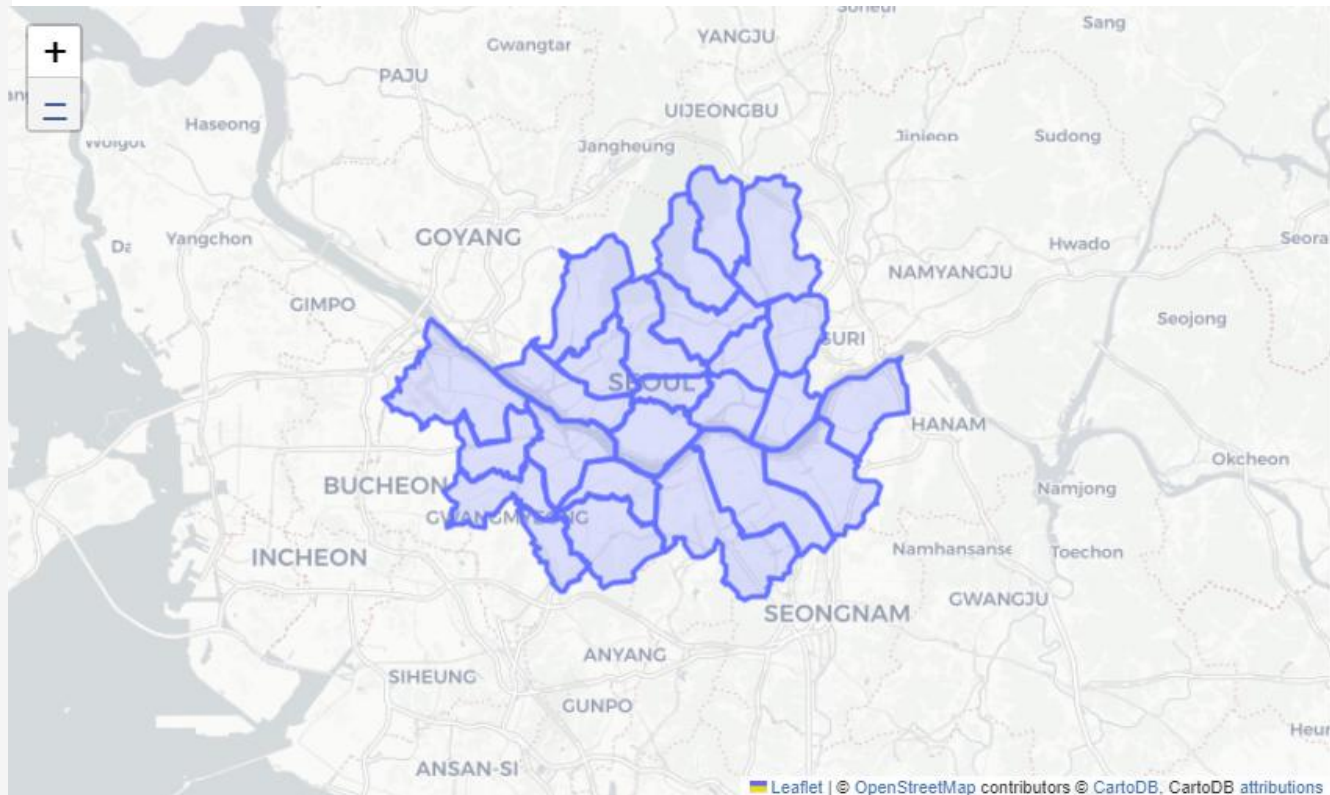
지도 생성하기

```
1 #지도를 생성하기 위한 folium 라이브러리 선언하기
2 #import folium
3 import folium
4
5
6 #folium.Map()을 사용하여 지도 생성하고 출력하기
7 #변수명 = folium.Map(속성들)
8 #location=[위도, 경도] : 지도의 중심 좌표를 [위도, 경도] 또는(위도, 경도)로 나타냄
9 #zoom_start = 정수 : 지도를 처음 그릴때 확대 정도
10 #zoom_control=True : zoom in/out 버튼 표시 여부(True/False)
11 #control_scales=False : 스케일 컨트롤 버튼 표시 여부(True/False)
12 #tiles= '스타일' : 지도 스타일을 지정(openstreetmap,stamenterrain,cartodbpositron)
13 map = folium.Map(location=[37.4965,126.9572], #송실대학교 중심좌표
14                  zoom_start=11, #초기 확대 정도
15                  tiles='cartodbpositron') #지도 스타일
16 map
```



GeoJson을 지도에 추가하여 확인하기

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

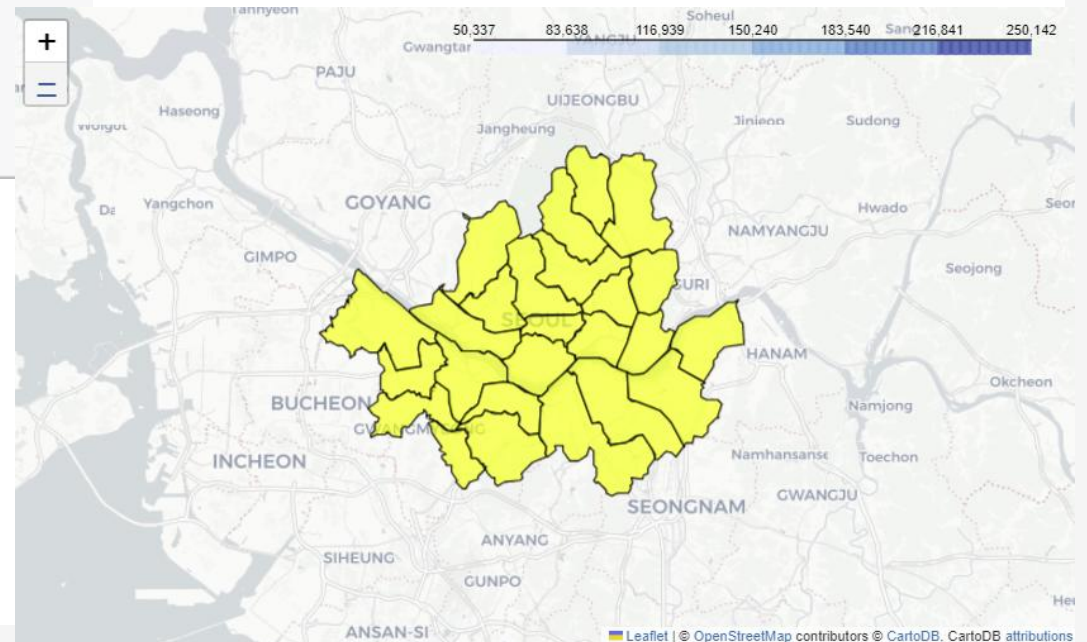


단계구분도를 생성하기

- 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

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

'₩u3000₩u3000₩u3000종로구'

불필요한 문자가 삽입됨

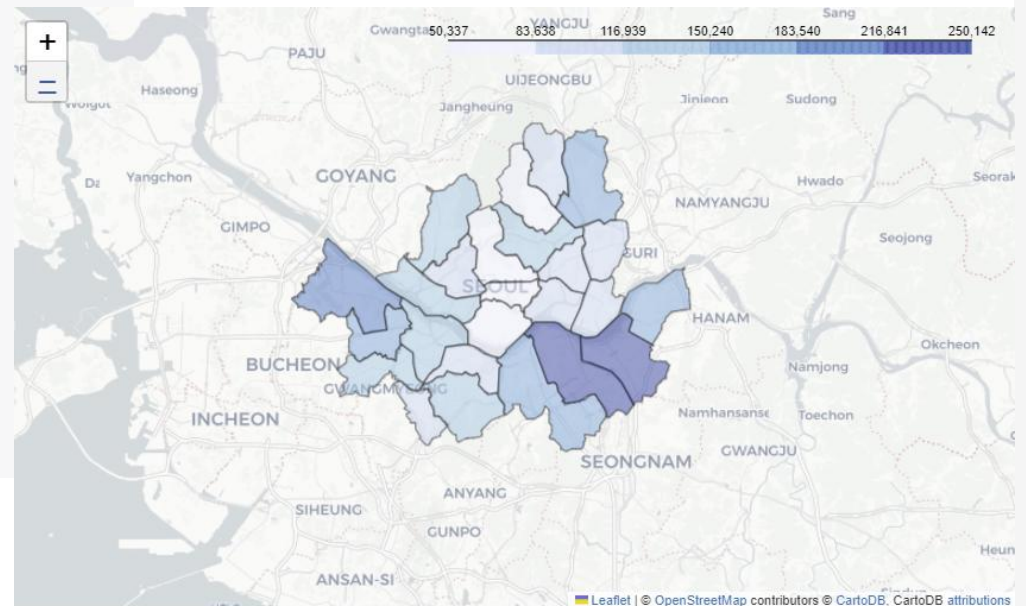
```
1 #불필요한 일부분자열 삭제하기
2 #df['열이름'] = df['열이름'].str.replace('찾을 문자열', '변경할 문자열')
3 # 열이름: 부분 변경할 문자열이 있는 열이름
4
5 carData['행정구역별'] = carData['행정구역별'].str.replace('₩u3000₩u3000₩u3000', '')
6 carData['행정구역별'][1]
```

'종로구'

단계구분도를 생성하기

- 단계구분도 재실행하고 추가 옵션 설정하기

```
1 map = folium.Map(location=[37.4965,126.9572],      #송실대학교 중심좌표
2                 zoom_start=11,                    #초기 확대 정도
3                 tiles='cartodbpositron')          #지도 스타일
4
5 folium.Choropleth(geo_data = geoSeoul,
6                  data=carData[1:],
7                  columns=['행정구역별', '자동차등록대수(대)'],
8                  key_on = 'feature.properties.SIG_KOR_NM',
9                  fill_color='Blues',
10                 nan_fill_color = 'yellow',
11                 fill_opacity=0.5,
12                 line_opacity=0.5
13                 ).add_to(map)
14 map
```



수고하셨습니다.