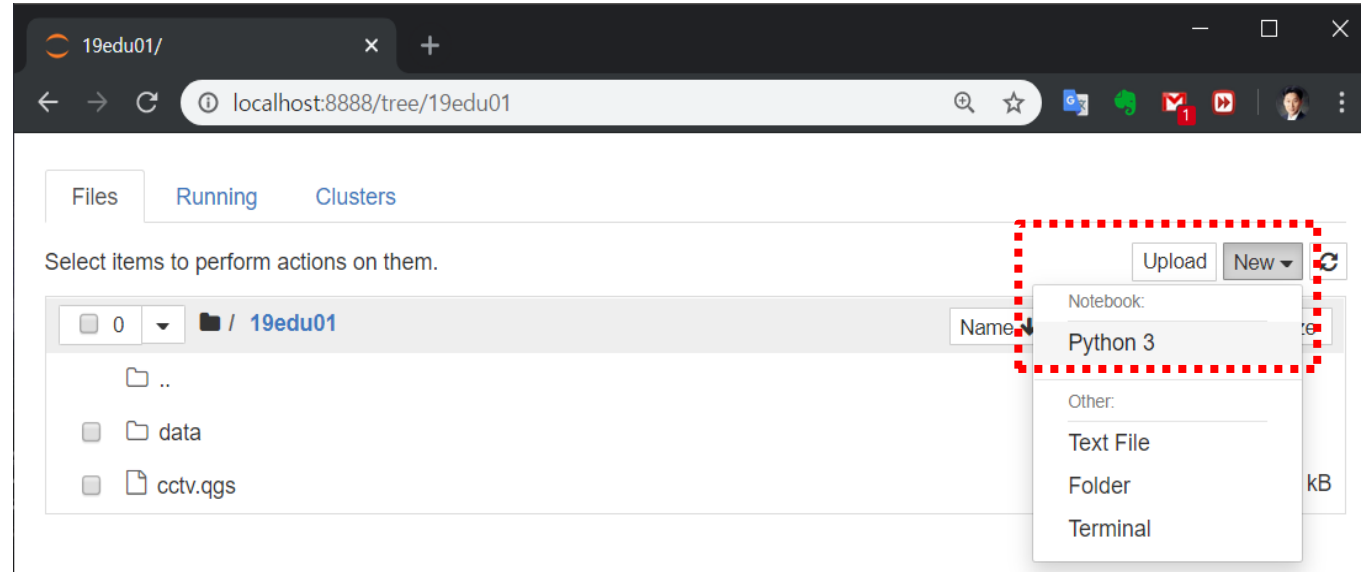


4차시 - 파이썬 기반 공간분석 및 시각화

Jupyter Notebook에서 새 파이썬 코드 생성

- Jupyter Notebook을 실행하면 웹브라우저 상에 로컬 웹서비스로 접속됨 (크롬/파이어폭스 권장)
 - 상단 오른쪽의 New 버튼을 눌러 Python3를 클릭하면 새 탭에 새로운 파이썬 코드를 입력할 수 있는 화면이 나타남



- Cell 단위로 파이썬 코드를 입력할 수 있는 구조로 되어 있어 여기에 일정 단락(단위 결과를 확인할 수 있는)별로 코드를 입력
 - 먼저 Pandas, Geopandas 등 분석에 필요한 패키지(라이브러리)를 호출

In [2]:

```
import pandas as pd
import geopandas as gpd # Import geopandas
import fiona
```

CCTV 데이터 로딩 및 속성 추출

- 다운로드 받은 CCTV CSV 파일을 pandas 데이터프레임으로 로딩
 - 필요한 컬럼만 적용하면 데이터량을 줄일 수 있음

CCTV CSV 로딩

```
df_cctv = pd.read_csv('./data/org/전국cctv표준데이터.csv', index_col=None, header=0,
                      names=['admin', 'addr1', 'addr2', 'usetype', 'cnt', 'pixel', 'azimuth', 'keep',
                              'inst_date', 'tel', 'lat', 'lon', 'up_date', 'offer_cd', 'offer', 'id2'],
                      usecols=['admin', 'addr1', 'lat', 'lon'],
                      dtype={'admin':object, 'addr1':object, 'lat':float, 'lon':float},
                      encoding="EUC-KR" )
df_cctv.head()
```

Out[21]:

	admin	addr1	lat	lon
0	청도공영사업공사	경상북도 청도군 화양읍 남성현로 348	35.687293	128.724165
1	청도공영사업공사	경상북도 청도군 화양읍 남성현로 348	35.687293	128.724165
2	청도공영사업공사	경상북도 청도군 화양읍 남성현로 348	35.687293	128.724165
3	청도공영사업공사	경상북도 청도군 화양읍 남성현로 348	35.687293	128.724165
4	청도공영사업공사	경상북도 청도군 화양읍 남성현로 348	35.687293	128.724165

- 서울시 데이터만 추출하고, 기본정보 확인

CCTV CSV 에서 서울시 데이터만 추출

df_cctv[['lat', 'lon']].dropna() #좌표값 없는 데이터 제거

```
df_cctv2 = df_cctv[df_cctv['addr1'].str.contains('서울특별시')==True]
df_cctv2.head()
```

Out[122]:

	admin	addr1	lat	lon
8089	서울특별시 송파구청	서울특별시 송파구 바람드리길 2	37.538288	127.115963
8090	서울특별시 송파구청	서울특별시 송파구 바람드리길 14-1	37.536698	127.118098
8091	서울특별시 송파구청	서울특별시 송파구 천호대로 996	37.538177	127.122626
8092	서울특별시 송파구청	서울특별시 송파구 올림픽로57길 2	37.534559	127.121416
8093	서울특별시 송파구청	서울특별시 송파구 강동대로 76	37.526222	127.115766

추출한 CCTV의 기본정보 및 기초 통계 확인

df_cctv2.info()

#df_cctv2.describe()

<class 'pandas.core.frame.DataFrame'>

Int64Index: 24199 entries, 8089 to 160379

Data columns (total 4 columns):

admin 24199 non-null object

addr1 24199 non-null object

lat 24199 non-null float64

lon 24199 non-null float64

dtypes: float64(2), object(2)

memory usage: 945.3+ KB

CCTV 공간데이터화

- Geopandas를 이용하여 좌표를 포인트 공간데이터화

```
# 강남구 CCTV 공간데이터화
```

```
geom_cctv = gpd.points_from_xy(df_cctv2.lon, df_cctv2.lat)  
geom_cctv[0:5]
```

Out[35]:

```
[<shapely.geometry.point.Point at 0x16326aacc88>,  
<shapely.geometry.point.Point at 0x16326aacc08>,  
<shapely.geometry.point.Point at 0x1630aba5f08>,  
<shapely.geometry.point.Point at 0x16326b2bf08>,  
<shapely.geometry.point.Point at 0x16326b2b308>]
```

- 포인트 좌표를 EPSG 4326로 정의하여 공간 데이터프레임으로 생성

```
# 좌표계 정의/변환용 라이브러리
```

```
import pyproj  
from fiona.crs import from_epsg  
columns = ['admin', 'addr1']  
gdf_cctv = gpd.GeoDataFrame(df_cctv2[columns], geometry=geom_cctv, crs=from_epsg(4326))  
#df_cctv_gdf.info()  
gdf_cctv.head()
```

Out[125]:

	admin	addr1	geometry
8089	서울특별시 송파구청	서울특별시 송파구 바람드리길 2	POINT (127.11596 37.53829)
8090	서울특별시 송파구청	서울특별시 송파구 바람드리12길 14-1	POINT (127.11810 37.53670)
8091	서울특별시 송파구청	서울특별시 송파구 천호대로 996	POINT (127.12263 37.53818)
8092	서울특별시 송파구청	서울특별시 송파구 올림픽로57길 2	POINT (127.12142 37.53456)
8093	서울특별시 송파구청	서울특별시 송파구 강동대로 76	POINT (127.11577 37.52622)

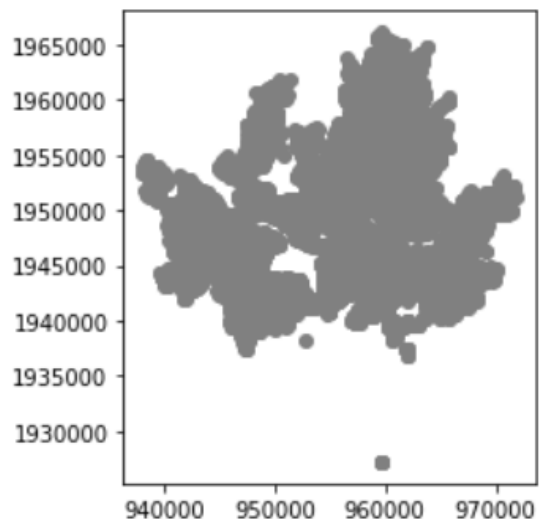
CCTV 좌표계 변환 및 geojson 저장

- EPSG 5179 좌표계로 변환하고, 차트로 확인

```
gdf_cctv2 = df_cctv_gdf.to_crs(epsg=5179) #좌표계를 5179로 변환  
gdf_cctv2.plot(color='gray')
```

Out[126]:

<matplotlib.axes._subplots.AxesSubplot at 0x16336ed4348>



- 강남구만 재추출

```
#강남구 CCTV만 한번 더 추출  
gdf_cctv3 = gdf_cctv2[gdf_cctv2['admin'].isin(['서울특별시 강남구청'])]  
gdf_cctv3.head()
```

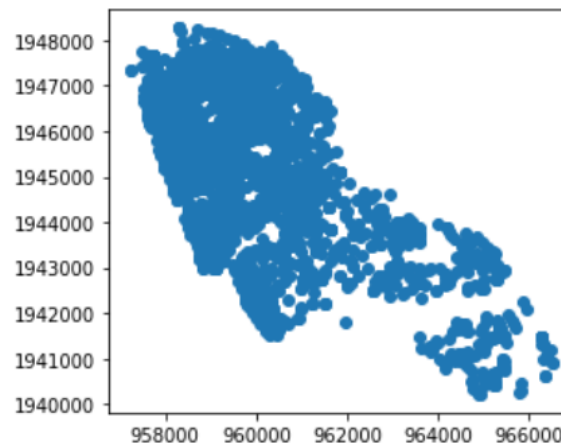
Out[145]:

	admin	addr1	geometry
77730	서울특별시 강남구청	서울특별시 강남구 헌릉로622길 27	POINT (965231.546 1940473.890)

```
gdf_cctv3.plot()
```

Out[146]:

<matplotlib.axes._subplots.AxesSubplot at 0x163379d7588>





CCTV geojson 저장 및 확인

- Geojson으로 저장

```
#CCTV 공간데이터를 geojson으로 저장
gdf_cctv2.to_file('./data/python/cctv.geojson', driver='GeoJSON')
gdf_cctv3.to_file('./data/python/cctv_gn.geojson', driver='GeoJSON')
```

원도우탐색기

이름	수정한 날짜	유형	크기
 cctv.geojson	2019-12-20 오후 1:21	GEOJSON 파일	5,221KB
 cctv_gn.geojson	2019-12-20 오후 1:21	GEOJSON 파일	437KB



텍스트 에디터로 확인

- Notepad++
- 대용량은 gVim 등

```
{
  "type": "FeatureCollection",
  "crs": { "type": "name", "properties": { "name": "urn:ogc:def:crs:EPSG::5179" } },
  "features": [
    { "type": "Feature", "properties": { "admin": "서울특별시 강남구청", "addr1": "서울특별시 강남구 현릉로622길 27" }, "geometry": { "type": "Point", "coordinates": [ 965231.545, 125.0667 ] } },
    { "type": "Feature", "properties": { "admin": "서울특별시 강남구청", "addr1": "서울특별시 강남구 방고개로36길 6" }, "geometry": { "type": "Point", "coordinates": [ 965272.212, 125.0667 ] } },
    { "type": "Feature", "properties": { "admin": "서울특별시 강남구청", "addr1": "서울특별시 강남구 방고개로34길 23 (세곡동)" }, "geometry": { "type": "Point", "coordinates": [ 965272.212, 125.0667 ] } },
    { "type": "Feature", "properties": { "admin": "서울특별시 강남구청", "addr1": "서울특별시 강남구 현릉로590길 11 (세곡동, 리엔파크3단지)" }, "geometry": { "type": "Point", "coordinates": [ 965272.212, 125.0667 ] } },
    { "type": "Feature", "properties": { "admin": "서울특별시 강남구청", "addr1": "서울특별시 강남구 현릉로590길 11 (세곡동, 리엔파크3단지)" }, "geometry": { "type": "Point", "coordinates": [ 965272.212, 125.0667 ] } },
    { "type": "Feature", "properties": { "admin": "서울특별시 강남구청", "addr1": "서울특별시 강남구 현릉로618길 8" }, "geometry": { "type": "Point", "coordinates": [ 965125.0667, 125.0667 ] } }
  ]
}
```

도서관 CSV 로딩

• 도서관 CSV 로딩

```
#도서관 csv 로딩
df_library = pd.read_csv('./data/org/전국도서관표준데이터.csv', index_col=None, header=0,
                        names=['name', 'do', 'gu', 'type', 'offday', 'time1', 'time2', 'time3', 'time4',
                                'time5', 'time6', 'seat', 'datacnt1', 'datacnt2', 'datacnt3', 'rent1', 'rentdat', 'addr', 'admin',
                                'tel', 'land_area', 'bd_area', 'url', 'lat', 'lon', 'update', 'offer_cd', 'offer', 'blank'],
                        usecols=['name', 'do', 'gu', 'lat', 'lon'],
                        dtype={'name':object, 'do':object, 'gu':object, 'lat':float, 'lon':float},
                        encoding="EUC-KR" )
df_library.head()
```

Out[128]:

	name	do	gu	lat	lon
0	육과공공도서관	전라남도	곡성군	35.274540	127.13557
1	도봉어린이문화정보도서관(디지털자료실)	서울특별시	서울특별시 도봉구	37.659007	127.04947
2	도봉어린이문화정보센터(자료열람실, 잉글리시 아일 랜드)	서울특별시	서울특별시 도봉구	37.659007	127.04947
3	학마을도서관 (종합자료실)	서울특별시	서울특별시 도봉구	37.662108	127.02782
4	학마을도서관 (어린이 자료실)	서울특별시	서울특별시 도봉구	37.662110	127.02782

• 도서관에서 서울시 데이터만 속성 추출

```
# 도서관 CSV 에서 서울시 데이터만 추출
df_library[['lat', 'lon']].dropna() #좌표값 없는 데이터 제거
df_library2 = df_library[df_library['do'] == '서울특별시'] #시도 항목 값이 서울인 것만 추출
df_library2.head()
```

Out[129]:

	name	do	gu	lat	lon
1	도봉어린이문화정보도서관(디지털자료실)	서울특별시	서울특별시 도봉구	37.659007	127.04947
2	도봉어린이문화정보센터(자료열람실, 잉글리시 아일 랜드)	서울특별시	서울특별시 도봉구	37.659007	127.04947
3	학마을도서관 (종합자료실)	서울특별시	서울특별시 도봉구	37.662108	127.02782
4	학마을도서관 (어린이 자료실)	서울특별시	서울특별시 도봉구	37.662110	127.02782
5	방학동영유아플라자	서울특별시	서울특별시 도봉구	37.663230	127.03033

도서관 공간데이터화

- 도서관 - 좌표로 공간데이터화

```
# 서울시 도서관 공간데이터화
geom_lib = gpd.points_from_xy(df_library2.lon, df_library2.lat)
geom_lib[0:3]
```

Out[131]:

```
[<shapely.geometry.point.Point at 0x163371e2288>,
 <shapely.geometry.point.Point at 0x163371e2b88>,
 <shapely.geometry.point.Point at 0x163371e2888>]
```

```
columns = ['name', 'gu']
gdf_library = gpd.GeoDataFrame(df_library2[columns], geometry=geom_lib, crs=from_epsg(4326))
#df_library_gdf.info()
gdf_library.head()
```

Out[132]:

	name	gu	geometry
1	도봉어린이문화정보도서관(디지털자료실)	서울특별시 도봉구	POINT (127.04947 37.65901)
2	도봉어린이문화정보센터(자료열람실, 잉글리시 아일랜드)	서울특별시 도봉구	POINT (127.04947 37.65901)
3	학마을도서관 (종합자료실)	서울특별시 도봉구	POINT (127.02782 37.66211)
4	학마을도서관 (어린이 자료실)	서울특별시 도봉구	POINT (127.02782 37.66211)
5	방학동영유아플라자	서울특별시 도봉구	POINT (127.03033 37.66323)

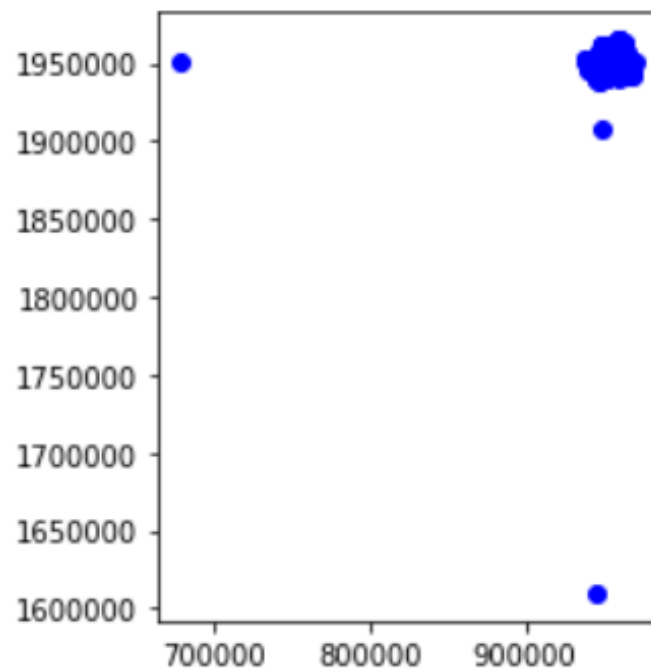
도서관 공간데이터 좌표계 변환

- 5179로 좌표계 변환 – 이상치가 존재

```
gdf_library2 = df_library_gdf.to_crs(epsg=5179) #좌표계를 5179로 변환  
gdf_library2.plot(color='blue')
```

Out[133]:

<matplotlib.axes._subplots.AxesSubplot at 0x163370adf08>



행정구역 SHP 로딩 및 좌표계 정의

#좌표 범위를 벗어나는 데이터를 제외하기 위해 행정구역 공간데이터 로딩

```
gdf_admin_gu_pg = gpd.read_file('./data/org/SIG_201905/TL_SCCO_SIG.shp', encoding='euc-kr')
```

#euc-kr, euckr, utf-8, cp949 중 선택

```
gdf_admin_gu_pg.crs
```

Out[134]:

```
{'proj': 'tmerc',  
 'lat_0': 38,  
 'lon_0': 127.5,  
 'k': 0.9996,  
 'x_0': 1000000,  
 'y_0': 2000000,  
 'ellps': 'GRS80',  
 'units': 'm',  
 'no_defs': True}
```

내용상으로는 epsg 5179이나 EPSG 좌표계로 정의되어 있지 않음. 정의 필요

In [135]:

```
gdf_admin_gu_pg.crs = "epsg:5179"
```

```
gdf_admin_gu_pg.crs
```

Out[135]:

```
'epsg:5179'
```

서울시 구별 행정구역만 속성 추출

```
#서울지역 행정구역만 추출
```

```
gdf_admin_gu_pg['DO'] = gdf_admin_gu_pg['SIG_CD'].str.slice(start=0, stop=2)
gdf_admin_gu_pg2 = gdf_admin_gu_pg[gdf_admin_gu_pg['DO'].str.contains('11')==True]
gdf_admin_gu_pg2.info()
gdf_admin_gu_pg2.head()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
```

```
Int64Index: 25 entries, 0 to 24
```

```
Data columns (total 5 columns):
```

```
SIG_CD      25 non-null object
```

```
SIG_ENG_NM  25 non-null object
```

```
SIG_KOR_NM  25 non-null object
```

```
geometry     25 non-null geometry
```

```
DO           25 non-null object
```

```
dtypes: geometry(1), object(4)
```

```
memory usage: 1.2+ KB
```

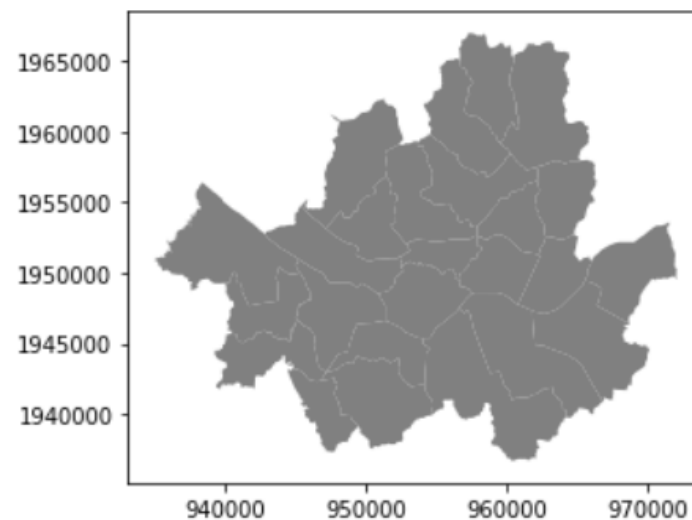
Out[139]:

	SIG_CD	SIG_ENG_NM	SIG_KOR_NM	geometry	DO
0	11110	Jongno-gu	종로구	POLYGON ((956615.453 1953567.199, 956621.579 1...	11
1	11140	Jung-gu	중구	POLYGON ((957890.386 1952616.746, 957909.908 1...	11
2	11170	Yongsan-gu	용산구	POLYGON ((953115.761 1950834.084, 953114.206 1...	11

```
gdf_admin_gu_pg2.plot(color='grey')
```

Out[140]:

<matplotlib.axes._subplots.AxesSubplot at 0x1632d15ce88>



서울시 행정구역 내의 도서관 포인트만 공간 추출

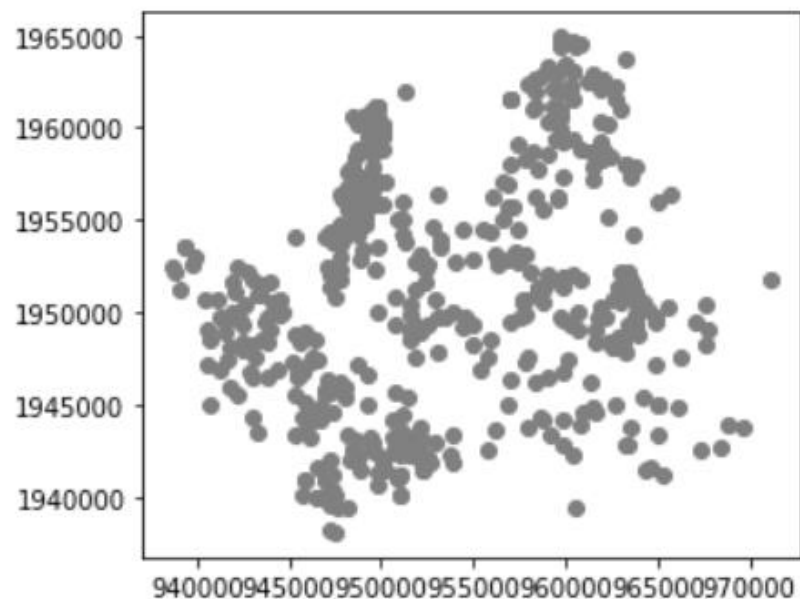
```
#서울시 안에 있는 도서관만 공간 선택
```

```
gdf_admin_gu_pg2['dummy'] = 'dummy' #서울시 구별 폴리곤에 더미 컬럼 추가
```

```
geom = gdf_admin_gu_pg2.dissolve(by='dummy').geometry[0] # '더미' 값으로 디졸브한 서울시 영역
```

```
gdf_library3 = df_library_gdf2[df_library_gdf2.within(geom)] #서울시 영역 내 도서관 포인트 선택
```

```
gdf_library3.plot(color='grey')
```



```
gdf_library3.head()
```

Out[151]:

	name	gu	geometry
1	도봉어린이문화정보도서관(디지털자료실)	서울특별시 도봉구	POINT (960262.103 1962262.693)
2	도봉어린이문화정보센터(자료열람실, 잉글리시 아일랜드)	서울특별시 도봉구	POINT (960262.103 1962262.693)
3	학마을도서관 (종합자료실)	서울특별시 도봉구	POINT (958354.222 1962616.089)
4	학마을도서관 (어린이 자료실)	서울특별시 도봉구	POINT (958354.249 1962616.333)
5	방학동영유아플라자	서울특별시 도봉구	POINT (958576.252 1962739.481)

강남구 도서관만 속성 추출

#강남구 도서관만 한번 더 추출

```
gdf_library4 = gdf_library3[gdf_library3['gu'].str.contains('강남구')==True]  
gdf_library4.head()
```

Out[152]:

	name	gu	geometry
1433	세곡도서관	강남구	POINT (965240.472 1941165.141)
1434	대치1작은도서관	강남구	POINT (960819.672 1943869.225)
1435	역삼2동작은도서관	강남구	POINT (959910.788 1944167.146)
1436	신사동작은도서관	강남구	POINT (957833.364 1947293.047)
1437	압구정동작은도서관	강남구	POINT (957981.364 1947604.146)

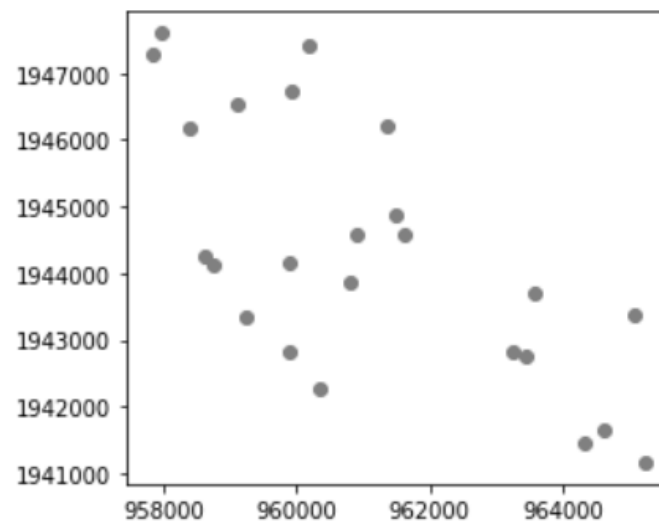
#도서관 공간데이터를 geojson으로 저장

```
gdf_library3.to_file('./data/python/library.geojson', driver='GeoJSON')  
gdf_library4.to_file('./data/python/library_gn.geojson', driver='GeoJSON')
```

```
gdf_library4.plot(color='grey')
```

Out[153]:

<matplotlib.axes._subplots.AxesSubplot at 0x1633728ce48>



보안등 로딩 및 공간데이터화

```
# 보안등 csv 로딩
df_slight = pd.read_csv('./data/org/전국보안등정보표준데이터.csv', index_col=None, header=0,
                        names=['slight_id', 'cnt', 'addr1', 'addr2', 'lat', 'lon', 'inst_year', 'inst_ty',
                              'tel', 'admin', 'up_date', 'offer_cd', 'offer', 'blank'],
                        usecols=['slight_id', 'lat', 'lon', 'inst_year', 'offer'],
                        dtype={'slight_id':object, 'lat':float, 'lon':float, 'inst_year':object,
                              'offer':object, },
                        encoding="EUC-KR" )
df_slight.head()
```

Out[163]:

	slight_id	lat	lon	inst_year	offer
0	11-14-10	37.633862	127.042875	2007	서울특별시 도봉구
1	11-17-47	37.633861	127.041317	2013	서울특별시 도봉구
2	11-16-01	37.633853	127.040014	2013	서울특별시 도봉구
3	11-17-10	37.633847	127.039282	2007	서울특별시 도봉구
4	11-14-09	37.633821	127.042452	2008	서울특별시 도봉구

```
len(df_slight)
```

Out[157]:

1590819

서울시 강남구 보안등만 속성 추출

```
# 보안등 csv 에서 서울시 데이터만 추출
df_slight[['lat', 'lon']].dropna() #좌표값 없는 데이터 제거
df_slight2 = df_slight[df_slight['offer'].str.contains('서울특별시 강남구')==True] #서울인 것만 추출
df_slight2.head()
```

Out[164]:

	slight_id	lat	lon	inst_year	offer
35953	개포동660-37	37.480474	127.062079	NaN	서울특별시 강남구
35954	개포동660-37	37.480474	127.062079	NaN	서울특별시 강남구
35955	개포동660-37	37.480474	127.062079	NaN	서울특별시 강남구
35956	개포동660-37	37.480474	127.062079	NaN	서울특별시 강남구
35957	개포동660-37	37.480474	127.062079	NaN	서울특별시 강남구

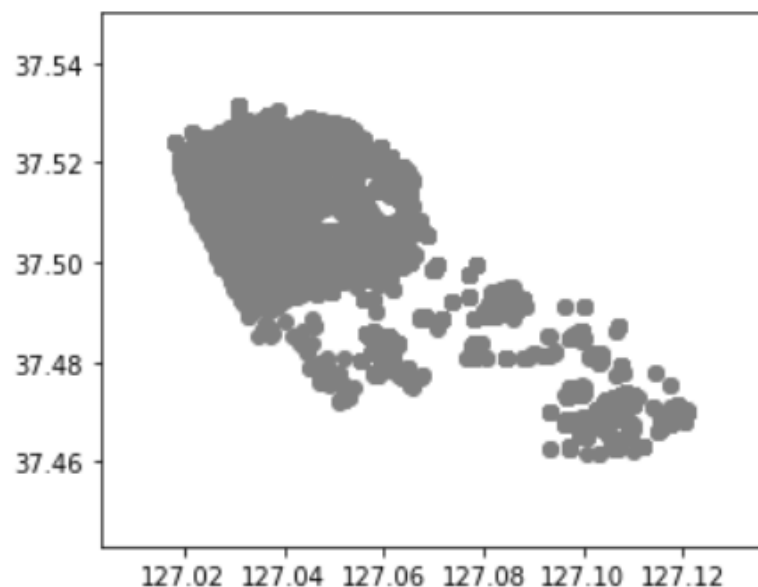
```
len(df_slight2)
```

Out[160]:

11498

보안등 공간데이터화, 좌표계 변환 및 파일 저장

```
# 강남구 보안등 공간데이터화
geom_slight = gpd.points_from_xy(df_slight2.lon, df_slight2.lat)
coulumns = ['slight_id', 'inst_year']
gdf_slight = gpd.GeoDataFrame(df_slight2[coulumns], geometry=geom_slight, crs=from_epsg(4326))
gdf_slight.plot(color='grey')
```



```
gdf_slight2 = gdf_slight.to_crs(epsg=5179) #좌표계를 5179로 변환
# 강남구 보안등 공간데이터를 geojson으로 저장
gdf_slight2.to_file('./data/python/slight.geojson', driver='GeoJSON')
```


학령인구 로딩

```
#학령인구 로딩
df_spop = pd.read_excel('./data/org/sl_teen_pop2018.xls', sheet_name = 'Sheet1',
                        index_col=None, header=2,
                        names=['year', 'gu', 'sum', 'year9_cnt', 'year9_ratio', 'year0_cnt',
                              'year0_ratio', 'spop_cnt', 'spop_ratio' ],
                        dtype={'gu':object, 'spop_ratio':float},
                        encoding="utf-8" )
df_spop.head()
```

Out[181]:

	year	gu	sum	year9_cnt	year9_ratio	year0_cnt	year0_ratio	spop_cnt	spop_ratio
0	2018	종로구	153065	24475	16.0	19179	12.5	20289	13.3
1	2018	중구	125725	16223	12.9	14145	11.3	13310	10.6
2	2018	용산구	228999	32494	14.2	29557	12.9	28164	12.3

```
df_spop.drop(['year', 'sum', 'year9_cnt', 'year9_ratio', 'year0_cnt', 'year0_ratio', 'spop_cnt'],
axis=1, inplace=True) #불필요한 컬럼 삭제
df_spop.head()
```

Out[187]:

	gu	spop_ratio
0	종로구	13.3
1	중구	10.6

범죄정보 로딩

범죄율 정보 로딩

```
df_crime = pd.read_excel('./data/org/sl_crime_2018.xls', sheet_name = 'Sheet1',
                        index_col=None, header=2,
                        names=['year', 'gu', 'crime_sum', 'etc1', 'etc2', 'etc3', 'etc4', 'etc5',
                              'etc6', 'etc7', 'etc8', 'etc9', 'etc10', 'etc11' ],
                        dtype={'gu':object, 'crime_sum':int},
                        encoding="utf-8" )

df_crime.head()
```

Out[190]:

	year	gu	crime_sum	etc1	etc2	etc3	etc4	etc5	etc6	etc7	etc8	etc9	etc10	etc11
0	2018	종로구	3690	3913	6	7	3	7	236	1100	1483	969	1962	1830
1	2018	중구	4030	2679	2	2	11	11	207	115	1855	832	1955	1719
2	2018	용산구	3411	2543	1	1	3	2	331	285	1096	522	1980	1733

```
df_crime.drop(['year', 'etc1', 'etc2', 'etc3', 'etc4', 'etc5', 'etc6', 'etc7', 'etc8', 'etc9',
               'etc10', 'etc11'], axis=1, inplace=True) #불필요한 컬럼 삭제
df_crime.head()
```

Out[191]:

	gu	crime_sum
0	종로구	3690
1	중구	4030
2	용산구	3411

학령인구를 행정구역에 속성 조인

```
# 시군구 행정구역에 학령인구 속성 조인
```

```
gdf_gu = pd.merge(gdf_admin_gu_pg2, df_spop, how='left', left_on='SIG_KOR_NM', right_on='gu')  
gdf_gu.head()
```

Out[192]:

	SIG_CD	SIG_ENG_NM	SIG_KOR_NM	geometry	DO	dummy	gu	spop_ratio
0	11110	Jongno-gu	종로구	POLYGON ((956615.453 1953567.199, 956621.579 1...	11	dummy	종로구	13.3
1	11140	Jung-gu	중구	POLYGON ((957890.386 1952616.746, 957909.908 1...	11	dummy	중구	10.6
2	11170	Yongsan-gu	용산구	POLYGON ((953115.761 1950834.084, 953114.206 1...	11	dummy	용산구	12.3
3	11200	Seongdong-gu	성동구	POLYGON ((959681.109 1952649.605, 959842.412 1...	11	dummy	성동구	12.6
4	11215	Gwangjin-gu	광진구	POLYGON ((964825.082 1952633.250, 964875.590 1...	11	dummy	광진구	13.5

범죄 정보를 행정구역에 속성 조인

```
# 시군구 행정구역에 범죄 통계 속성 조인
```

```
gdf_gu2 = pd.merge(gdf_gu, df_crime, how='left', left_on='SIG_KOR_NM', right_on='gu')  
gdf_gu2.head()
```

Out[193]:

G_CD	SIG_ENG_NM	SIG_KOR_NM	geometry	DO	dummy	gu_x	spop_ratio	gu_y	crime_sum
11110	Jongno-gu	종로구	POLYGON ((956615.453 1953567.199, 956621.579 1...	11	dummy	종로 구	13.3	종로 구	3690
11140	Jung-gu	중구	POLYGON ((957890.386 1952616.746, 957909.908 1...	11	dummy	중구	10.6	중구	4030
11170	Yongsan-gu	용산구	POLYGON ((953115.761 1950834.084, 953114.206 1...	11	dummy	용산 구	12.3	용산 구	3411

시군구 행정구역의 불필요 항목 정리 및 범죄율 항목 생성

```
columns2=['DO', 'SIG_CD', 'SIG_KOR_NM', 'spop_ratio', 'crime_sum' ]
gdf_gu3 = gpd.GeoDataFrame(gdf_gu2[columns2], geometry=gdf_gu2.geometry, crs=from_epsg(5179))

crime_sum2 = gdf_gu3['crime_sum'].sum(axis=0) #서울시 전체 범죄 발생건수 합계
print(crime_sum2)

gdf_gu3['crime_ratio'] = round(gdf_gu3['crime_sum'] / crime_sum2 * 100, 1) #소수 첫째자리 범죄율 산출
gdf_gu3.head()
```

101948

Out[234]:

	DO	SIG_CD	SIG_KOR_NM	spop_ratio	crime_sum	geometry	crime_ratio
0	11	11110	종로구	13.3	3690	POLYGON ((956615.453 1953567.199, 956621.579 1...	3.6
1	11	11140	중구	10.6	4030	POLYGON ((957890.386 1952616.746, 957909.908 1...	4.0
2	11	11170	용산구	12.3	3411	POLYGON ((953115.761 1950834.084, 953114.206 1...	3.3
3	11	11200	성동구	12.6	2457	POLYGON ((959681.109 1952649.605, 959842.412 1...	2.4
4	11	11215	광진구	13.5	3915	POLYGON ((964825.082 1952633.250, 964875.590 1...	3.8

차트 패키지 호출 및 학령인구 단계구분도 시각화

```
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
from mpl_toolkits.axes_grid1 import make_axes_locatable
%matplotlib inline
mpl.rcParams['font', family='NanumGothic'] #한글 폰트 적용시
plt.rcParams["figure.figsize"] = (20,10)
```

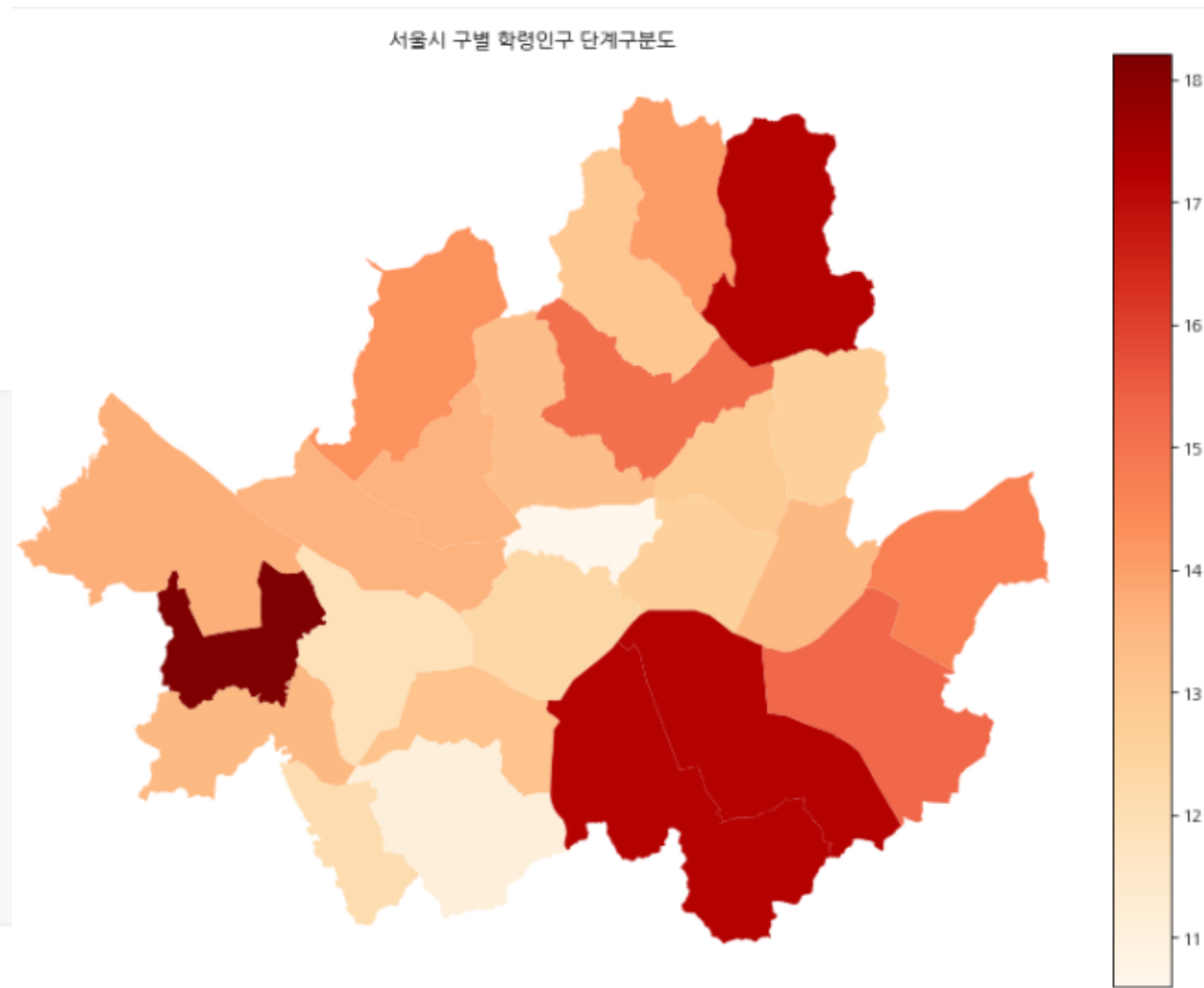
```
# 서울시 시군구 학령인구 단계구분도(Choropleth Map) 시각화
fig, ax = plt.subplots(1, 1)
# 범례
divider = make_axes_locatable(ax)
cax = divider.append_axes("right", size="5%", pad=0.1)
# 시각화
column = 'spop_ratio'
gdf_gu3.plot(column, ax=ax, legend=True, cax=cax, cmap='OrRd')

ax.set_title("서울시 구별 학령인구 단계구분도")
ax.set_axis_off()
plt.show()
```

참조 URL

https://matplotlib.org/mpl_toolkits/axes_grid/users/overview.html

https://matplotlib.org/3.1.1/gallery/axes_grid1/demo_colorbar_with_axes_divider.html

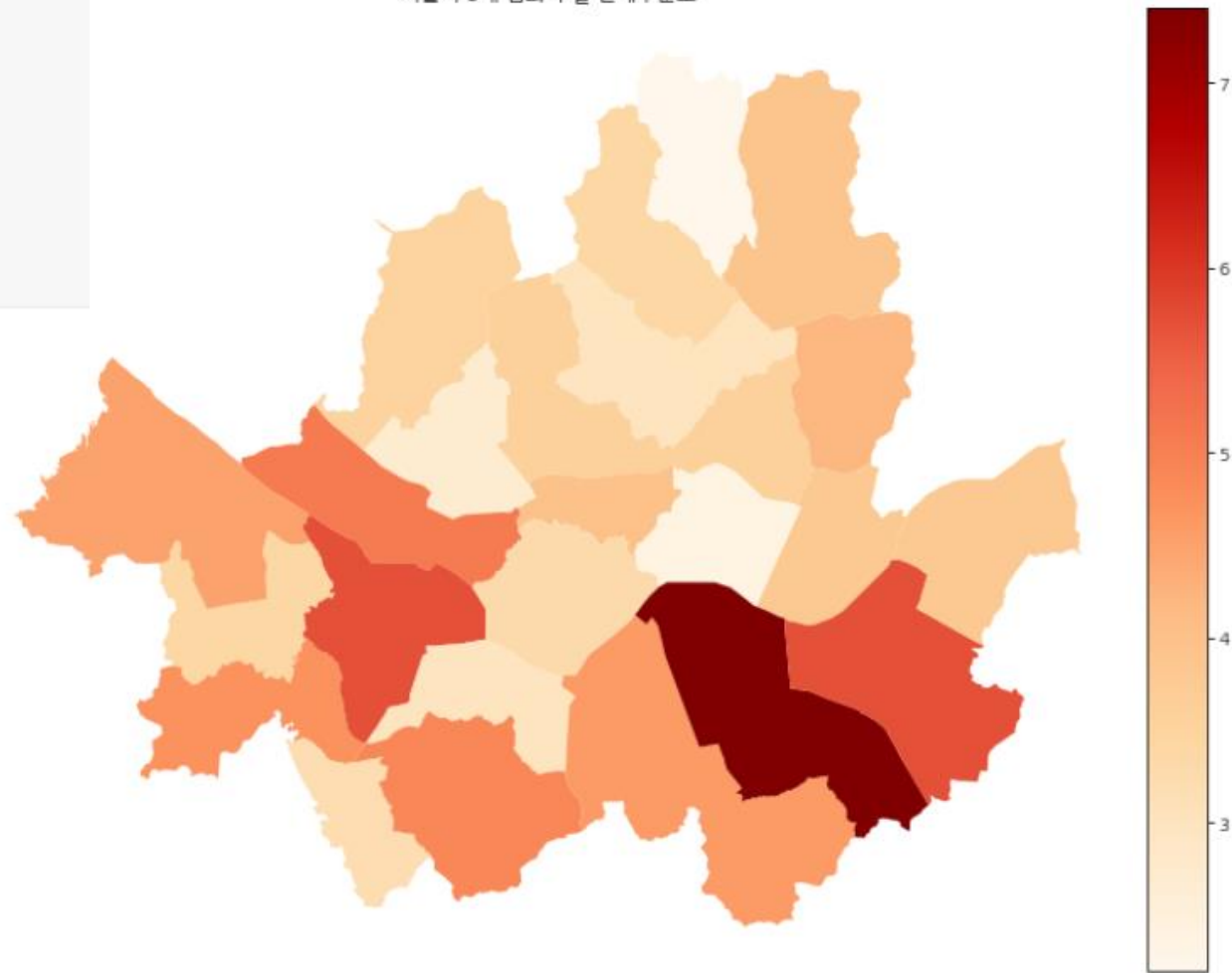


서울시 5대 범죄 구별 단계구분도 시각화

```
# 서울시 시군구 5대 범죄 발생률 단계구분도(Choropleth Map) 시각화
fig, ax = plt.subplots(1, 1)
# 범례
divider = make_axes_locatable(ax)
cax = divider.append_axes("right", size="5%", pad=0.1)
# 시각화
column = 'crime_ratio'
gdf_gu3.plot(column, ax=ax, legend=True, cax=cax, cmap='OrRd')

ax.set_title("서울시 5대 범죄 구별 단계구분도")
ax.set_axis_off()
plt.show()
```

서울시 5대 범죄 구별 단계구분도



참조 URL

<http://geopandas.org/mapping.html>

서울시 5대 범죄 구별 버블맵 시각화 (1)

서울시 5대 범죄 버블맵

```
import geoplot as gplt #conda install -c conda-forge geoplot
import geoplot.crs as gcrs
import mapclassify as mc
plt.rcParams["figure.figsize"] = (15,15)
```

```
scheme = mc.Quantiles(gdf_gu_pt['crime_sum'], k=5)
proj = gcrs.WebMercator()
```

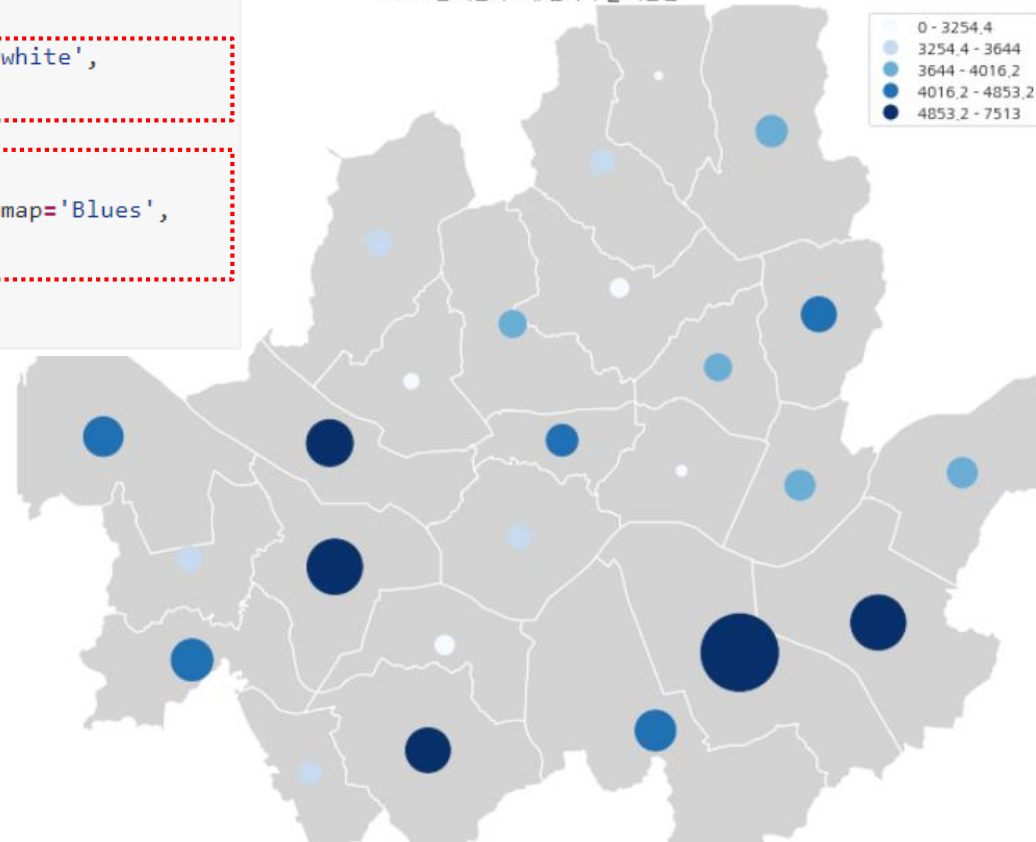
```
ax = gplt.polyplot(gdf_gu4326, zorder=-1, linewidth=1, projection=proj, edgecolor='white',
                  facecolor='lightgray', figsize=(12, 12) )
```

```
gplt.pointplot(gdf_gu_pt, projection=gcrs.WebMercator(),
               scale='crime_sum', limits=(5, 50), hue='crime_sum', scheme=scheme, cmap='Blues',
               legend=True, legend_var='hue', ax=ax)
```

```
plt.title("2018년 서울시 5대 범죄 구별 버블맵")
```

geoplot은 EPSG 4326으로 변환해야 함

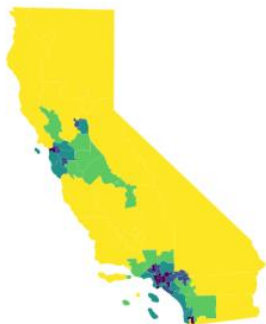
2018년 서울시 5대 범죄 구별 버블맵



scheme=None



scheme="Quantiles"



scheme="EqualInterval"



scheme="FisherJenks"



서울시 5대 범죄 구별 버블맵 시각화 (2)

```
# 서울시 5대 범죄 버블맵 2
```

```
import geoplot as gplt #conda install -c conda-forge geoplot
```

```
import geoplot.crs as gcrs
```

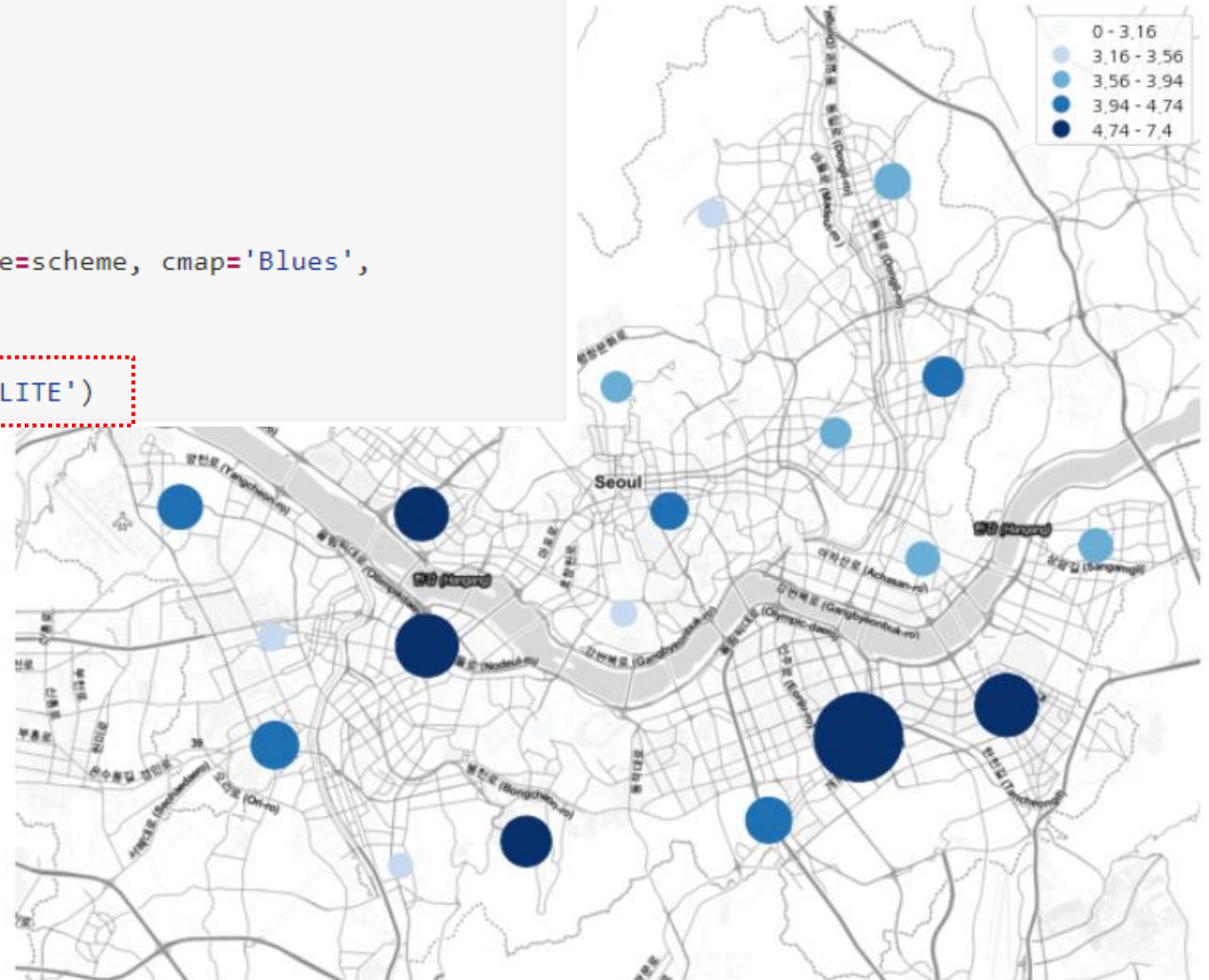
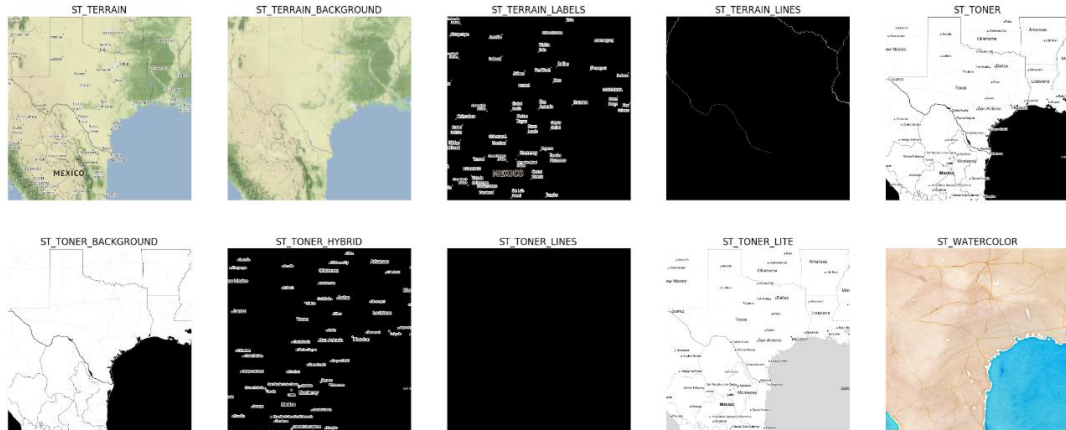
```
import mapclassify as mc
```

```
scheme = mc.Quantiles(gdf_gu_pt['crime_ratio'], k=5)
```

```
extent = gdf_gu4326.total_bounds
```

```
ax = gplt.pointplot(gdf_gu_pt, projection=gcrs.WebMercator(),  
    scale='crime_ratio', limits=(5, 50), hue='crime_ratio', scheme=scheme, cmap='Blues',  
    legend=True, legend_var='hue', figsize=(12, 12))
```

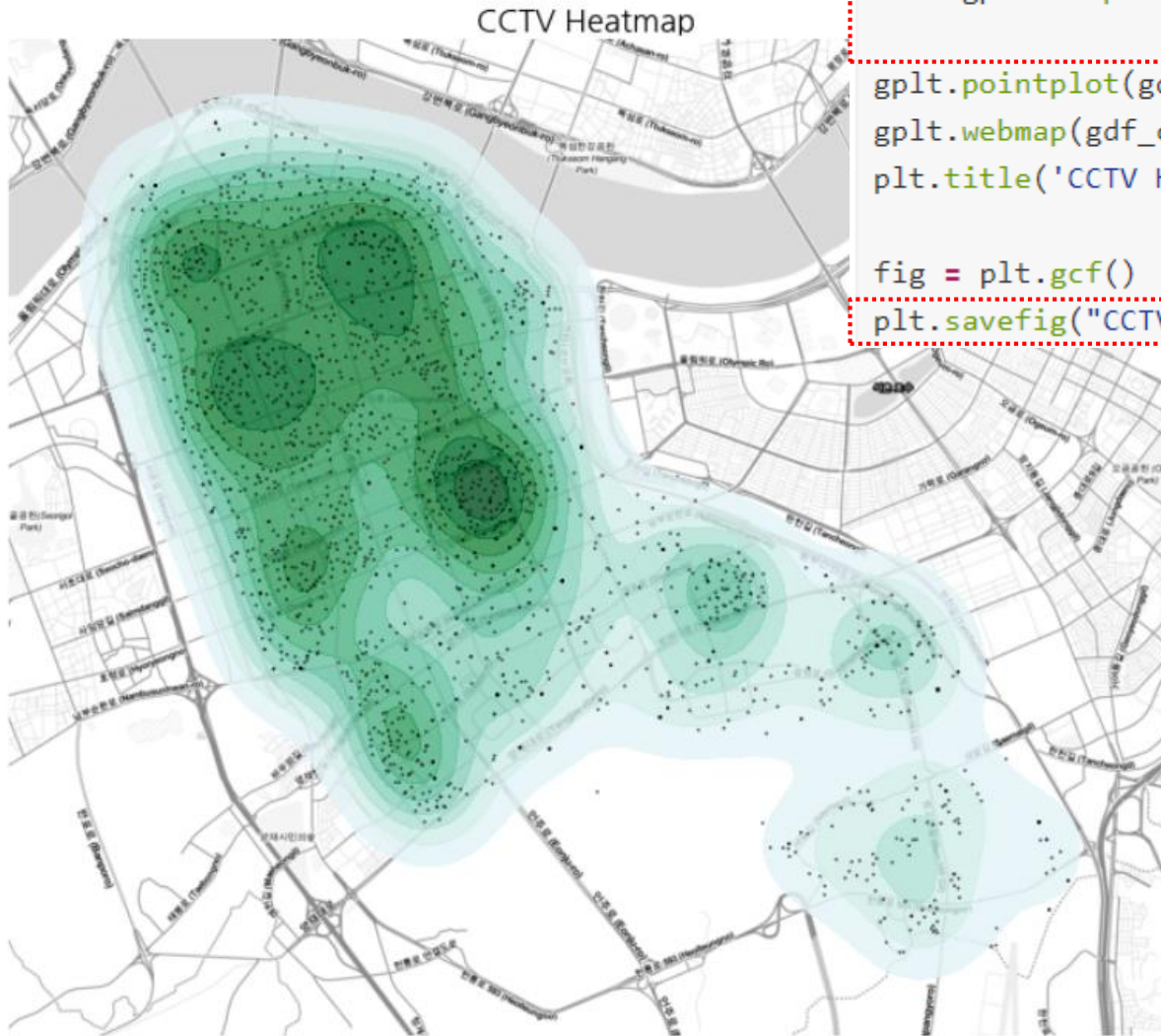
```
gplt.webmap(gdf_gu4326, ax=ax, extent=extent, provider='ST_TONER_LITE')
```



<https://github.com/darribas/contextily>

<http://maps.stamen.com/>

강남구 CCTV Geographic Heat Map



```
gdf_cctv4326 = gdf_cctv3.to_crs(epsg=4326) #강남구 CCTV 좌표계를 4326으로 변환

ax = gplt.kdeplot(gdf_cctv4326, cmap='BuGn', projection=gcrs.WebMercator(),
                 figsize=(12, 12), shade=True, alpha=0.8)
gplt.pointplot(gdf_cctv4326, s=1, color='black', alpha=0.5, ax=ax)
gplt.webmap(gdf_cctv4326, ax=ax, provider='ST_TONER_LITE')
plt.title('CCTV Heatmap', fontsize=18)

fig = plt.gcf()
plt.savefig("CCTV_Heatmap.png") #, bbox_inches='tight', pad_inches=0.01)
```

참조 URL

https://residentmario.github.io/geoplot/api_reference.html?highlight=kdeplot#geoplot.geoplot.kdeplot

<http://seaborn.pydata.org/generated/seaborn.kdeplot.html#seaborn.kdeplot>

실폭도로 공간 데이터 로딩

```
# 실폭도로 데이터 로딩
gdf_road = gpd.read_file('./data/org/Z_KAIS_TL_SPRD_RW_서울/Z_KAIS_TL_SPRD_RW_11000.shp',
                          encoding='euc-kr') #euc-kr, euckr, utf-8, cp949 중 선택

gdf_road.crs
```

Out[409]:

```
{}
```

```
# 실폭도로 좌표계를 지정하고 다시 확인
gdf_road.crs = "epsg:5181"
gdf_road.crs
```

Out[410]:

```
'epsg:5181'
```

```
gdf_road3857 = gdf_road.to_crs(epsg=3857) #도로 좌표계를 3857로 변환
gdf_road3857.plot()
```



강남구 도로만 공간 추출

```
# 구별 행정구역 공간데이터프레임에서 강남구만 속성 추출
```

```
gdf_gu_gn = gdf_gu3[gdf_gu3['SIG_KOR_NM'].isin(['강남구'])]
```

```
# gdf_gu_gn.plot()
```

```
# 실폭도로에서 강남구 부분만 공간 추출
```

```
gdf_rd_gn = gpd.overlay(gdf_gu_gn, gdf_road3857, how='intersection')
```

```
gdf_rd_gn.plot()
```



도서관, CCTV, 보안등 Buffering

```
# 도서관, CCTV, 보안등 BUFFER
```

```
from shapely.geometry import Polygon
```

```
#from shapely.ops import cascaded_union
```

```
gdf_lib_buf = gpd.GeoDataFrame()
```

```
gdf_cctv_buf = gpd.GeoDataFrame()
```

```
gdf_slight3 = gpd.GeoDataFrame()
```

```
gdf_slight_buf = gpd.GeoDataFrame()
```

```
#도서관은 버퍼하면서 명칭 속성을 붙여주기
```

```
gdf_lib_buf['geometry'] = gdf_library4.buffer(500) #500m
```

```
gdf_lib_buf = gpd.GeoDataFrame(gdf_library4['name'], geometry=gdf_lib_buf.geometry,
```

```
crs=from_epsg(3857))
```

```
gdf_cctv_buf['geometry'] = gdf_cctv3.buffer(50) #50m
```

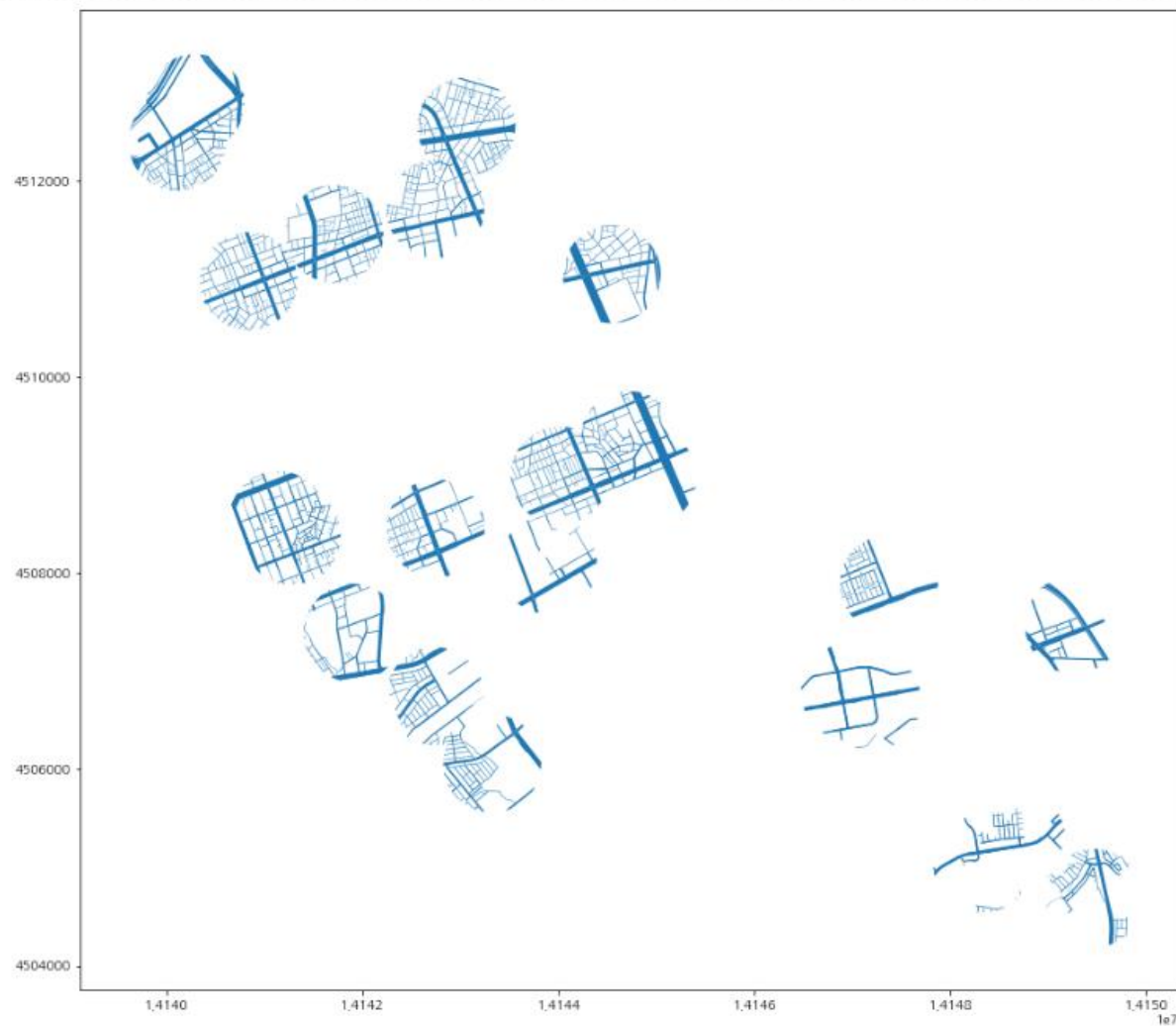
```
gdf_slight3 = gdf_slight2[gdf_slight2.is_valid] #도형 오류 제거 polygon = polygon.buffer(0)
```

```
gdf_slight_buf['geometry'] = gdf_slight3.buffer(20)
```

도서관 버퍼에서 실폭도로 부분만 공간 추출

```
# 도서관 버퍼에서 실폭도로 부분만 공간 추출
```

```
gdf_lib_rd = gpd.overlay(gdf_lib_buf, gdf_rd_gn, how='intersection')  
gdf_lib_rd.plot()
```



```
#도서관 버퍼는 명칭으로 디졸브
```

```
gdf_lib_rd2 = gdf_lib_rd.dissolve(by='name')  
gdf_lib_rd2.reset_index(level=['name'], inplace = True) #name을 column으로  
gdf_lib_rd2.info()  
gdf_lib_rd2.head(2)
```

CCTV 버퍼와 보안등 버퍼 UNION

```
# CCTV와 보안등 UNION
```

```
gdf_cclight = gpd.overlay(gdf_cctv_buf, gdf_slight_buf, how='union')
```

```
gdf_cclight.plot()
```

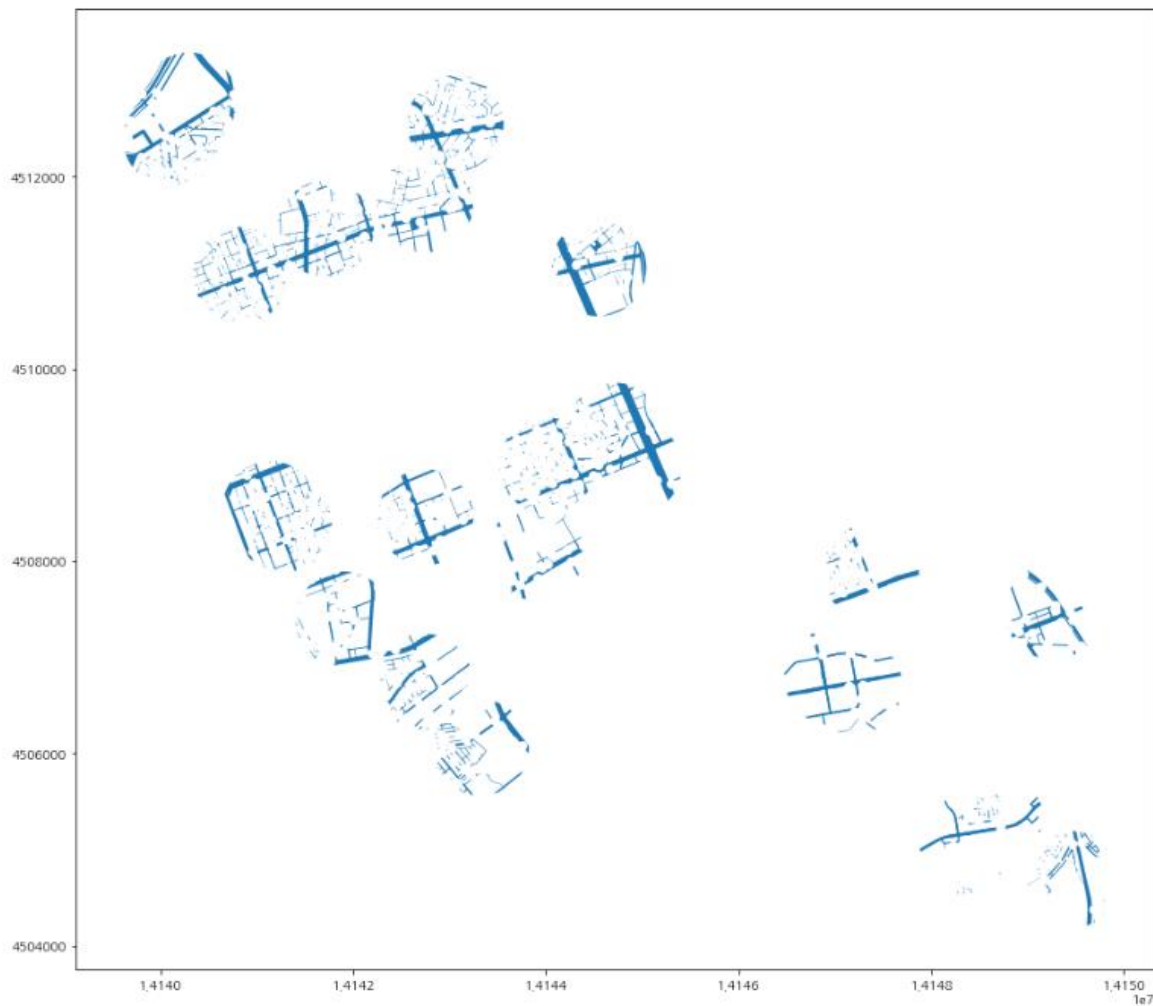


도서관 주변 실폭도로에서 CCTV-보안등 버퍼 영역 제거

```
# 도서관 주변 실폭도로에서 CCTV-보안등 버퍼 UNION를 제거
```

```
gdf_becareful = gpd.overlay(gdf_lib_rd2, gdf_cclight, how='difference')
```

```
gdf_becareful.plot()
```



남은 도로의 도형 면적을 속성 항목에 저장하고 좌표계 변환

```
gdf_becareful['area'] = gdf_becareful['geometry'].area #도형 면적  
gdf_becareful.head()
```

Out[74]:

	name	geometry	area
0	개포4동작은도서관	MULTIPOLYGON (((14143518.872 4506360.488, 1414...	82540.350300
1	논현도서관	MULTIPOLYGON (((14141247.827 4511395.722, 1414...	136204.683208
2	논현정보도서관	MULTIPOLYGON (((14141330.373 4510882.348, 1414...	107900.380208
3	대치1작은도서관	MULTIPOLYGON (((14143819.391 4507832.098, 1414...	58174.022332
4	대치도서관	MULTIPOLYGON (((14145324.439 4508848.413, 1414...	125081.070969

```
gdf_becareful4326 = gdf_becareful.to_crs(epsg=4326) #EPSG 4326 좌표계로 좌표 변환  
gdf_becareful4326.reindex(['idx'])  
minx, miny, maxx, maxy = gdf_becareful4326.geometry.total_bounds #바운더리 좌표 받기  
print( (minx + (maxx - minx)) , (miny + (maxy - miny)) ) #레이어 중심좌표 확인  
print(minx, miny)
```

127.10998111782676 37.530331667500334

127.01842500534462 37.46562700636925

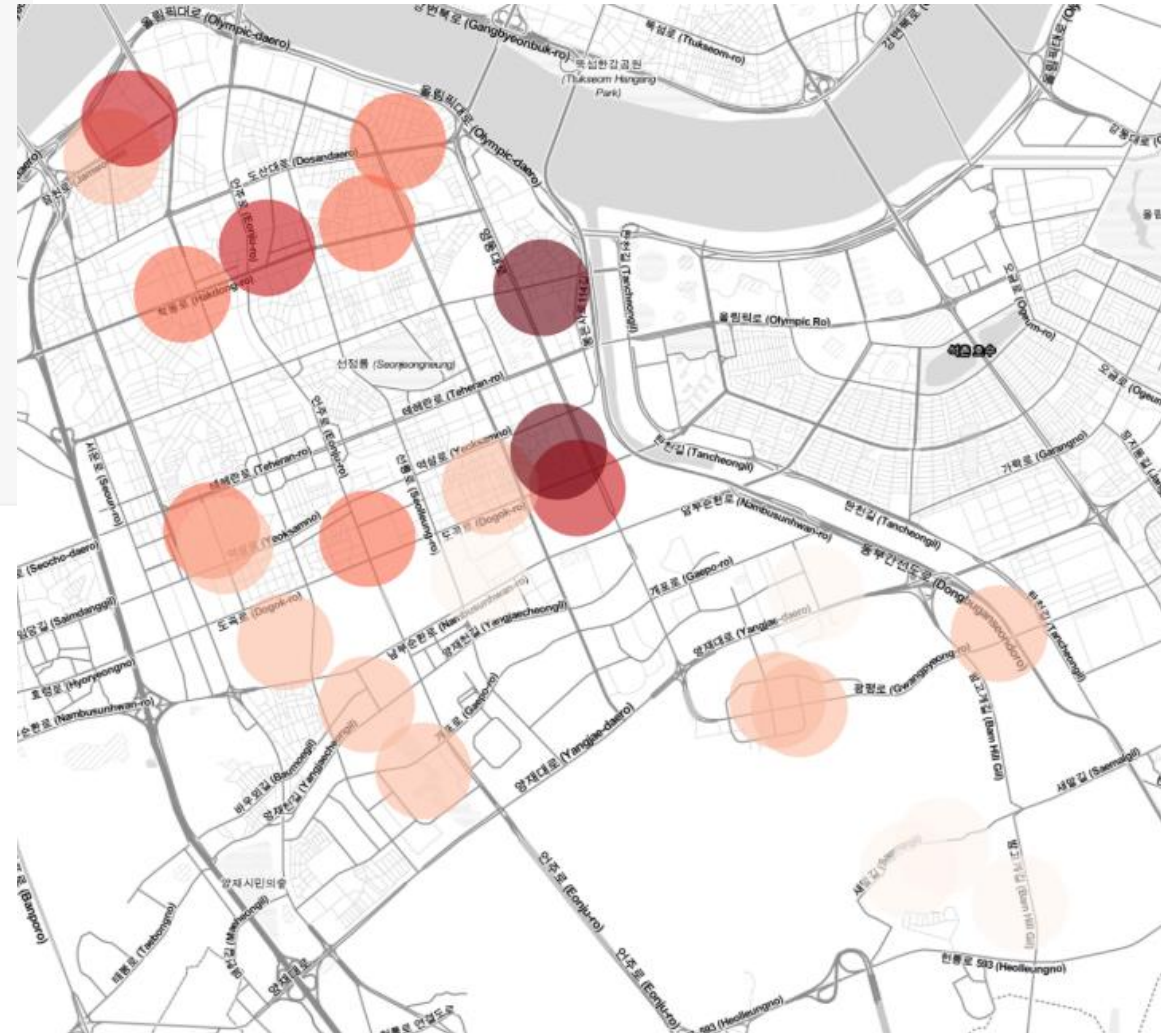
최종 분석 결과 시각화 - geoplot

```
#scheme = mc.Quantiles(gdf_libb4326['area'], k=5)
extent = gdf_libb4326.total_bounds

ax = gplt.webmap(gdf_becareful4326, projection=gcrs.WebMercator(),
                 provider='ST_TONER_LITE', figsize=(24, 24))
gplt.choropleth(gdf_libb4326, ax=ax, hue='area', cmap='Reds',
                projection=gcrs.WebMercator(), scheme='FisherJenks', alpha=0.6)

#ax = gplt.choropleth(gdf_libb4326, hue='area', cmap='Reds',
#                    # projection=gcrs.WebMercator(), scheme='FisherJenks',
#                    # figsize=(17, 17), alpha=0.6)
#gplt.webmap(gdf_becareful4326, ax=ax, provider='ST_TONER_LITE')
plt.title('도서관 주변 CCTV-보안등 적은 도로', fontsize=18)

fig = plt.gcf()
plt.savefig("도서관 주변 비안심도로.png") #, bbox_inches='tight', pad_inches=0.01)
```



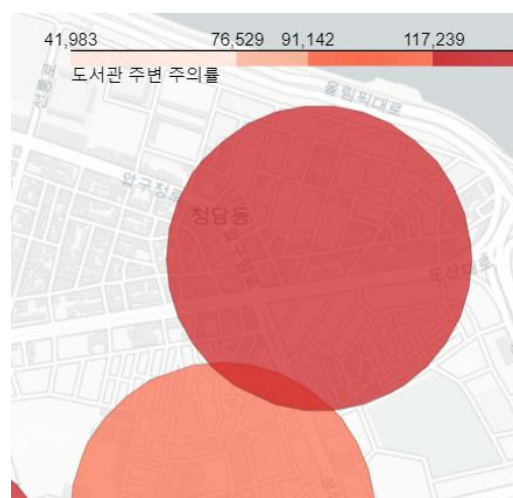
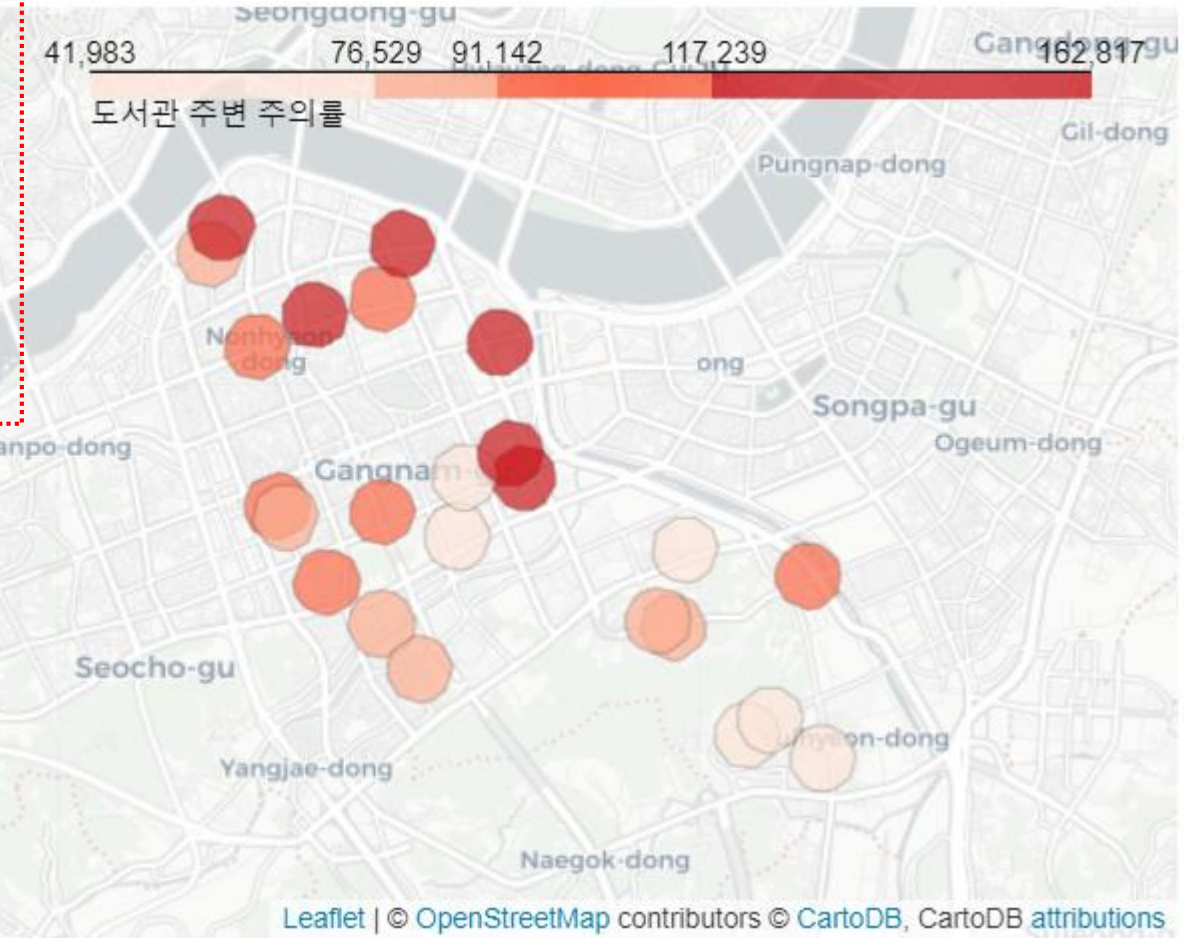
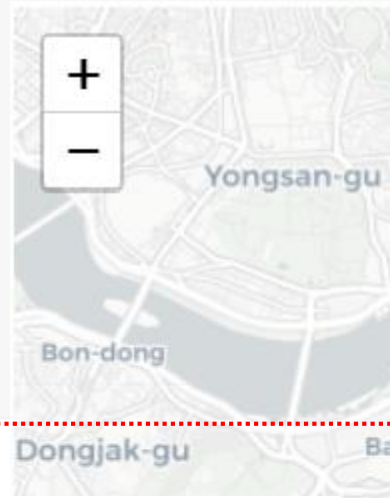
최종 분석 결과 시각화 - Folium

```
bins = list(gdf_libb4326['area'].quantile([0, 0.25, 0.5, 0.75, 1]))
```

```
map = folium.Map(location=[37.500, 127.049], zoom_start=12, tiles='CartoDB positron')  
# , width=500,height=600) #CartoDB positron #Stamen Toner
```

```
map.choropleth(geo_data = geo_libb4326, #g_bcare_j  
name='choropleth',  
data = gdf_libb4326,  
columns = ['name', 'area'],  
key_on = 'feature.properties.name',  
fill_color = 'Reds', #PuRd  
fill_opacity=0.7, line_opacity=0.2,  
legend_name='도서관 주변 주의를',  
bins=bins, reset=True )
```

map



참조 URL

<https://python-visualization.github.io/folium/modules.html#module-folium.map>

Leaflet | © OpenStreetMap contributors © CartoDB, CartoDB attributions

기타 - 단계구분도 색상 선택

- <http://colorbrewer2.org/>

정량 속성은 Sequential/ Diverging에서 선택
정성 속성은 Qualitative 선택

원하는 색상 선택시 url의 scheme 파라미터 값을 활용

<http://colorbrewer2.org/#type=sequential&scheme=BuGn&n=3>

Number of data classes: 3

Nature of your data: ☒ sequential ☐ diverging ☐ qualitative

Pick a color scheme:

Multi-hue: ☒ Single hue: ☐

Only show: ☒ colorblind safe ☒ print friendly ☒ photocopy safe

Context: ☐ roads ☐ cities ☒ borders

Background: ☒ solid color ☐ terrain

color transparency

3-class BuGn

HEX

#e5f5f9

#99d8c9

#2ca25f

EXPORT

how to use | updates | downloads | credits

COLORBREWER 2.0
color advice for cartography

© Cynthia Brewer, Mark Harrower and The Pennsylvania State University

[Source code and feedback](#)

[Back to Flash version](#)

[Back to ColorBrewer 1.0](#)

axismaps