

In [1]:

```

import pandas as pd
import numpy as np

import warnings
warnings.filterwarnings('ignore')

seongnam = pd.read_csv("./data/final_성남시.csv",encoding='CP949')
seongnam['구'] = seongnam['소재지도명주소'].str.split(" ",expand=True)[2]
seongnam

```

Out[1]:

	대상 시설명	소재지 도로명 주소	위도	경도	ada	신 호 등 개 수	엘 로 우 카 펫	울 타 리 길 이	횡 단 보 도 폭	횡 단 보 도 개 수	CCTV 설치 대수	어 린 이 놀 이 시 설 개 수	안 전 표 지 판 개 수	과 속 방 지 턱	구
0	대원초등학교	경기도 성남시 중원구 희망로 343번길 24	37.436671	127.155099	0	1	1	434	3.0	1	6	0	1	11	중원구
1	대일초등학교	경기도 성남시 중원구 금상로 55번길 22	37.438468	127.164185	0	1	1	430	3.0	1	3	1	1	31	중원구
2	성남동초등학교	경기도 성남시 중원구 자혜로 32번길 10	37.449215	127.162178	0	2	0	966	5.0	2	5	5	1	46	중원구
3	내정초등학교	경기도 성남시 분당구 내정로 174번길 19	37.372402	127.122264	0	2	1	792	8.0	2	2	2	1	27	분당구
4	상대원초등학교	경기도 성남시 중원구 순환로 214번길 16	37.442382	127.181591	0	1	1	944	3.0	1	3	2	2	16	중원구
...
68	위례푸른초	경기도 성남시 수정구 위례순	37.470731	127.150981	0	4	1	1488	6.7	4	0	6	4	10	수정구

구	과속방지턱	안전표지판개수	어린이놀이시설개수	CCTV설치대수	횡단보도개수	횡단보도폭	울타리길이	옐로우카펫	신호등개수	ada	경도	위도	소재지 도로명주소	대상시설명
수정구	2	4	11	1	4	6.7	1062	1	4	0	127.135874	37.471899	경기도성남시수정구위례순환로71	위례중앙초등학교
수정구	10	4	8	5	4	6.7	2289	1	4	0	127.098687	37.428469	경기도성남시수정구대왕판교로959번길48	왕남초등학교
수정구	1	4	3	5	4	6.7	501	1	4	0	127.149115	37.447498	경기도성남시수정구공원로370번길29	희망대초등학교
수정구	18	4	4	6	4	6.7	424	1	4	1	127.130386	37.462470	경기도성남시수정구성남대로1480번길25	북정초등학교

73 rows × 15 columns

```
In [2]: seongnam["ada"].value_counts()
```

```
Out[2]: 0    38
        1    35
        Name: ada, dtype: int64
```

```
In [3]: print("분당구 :", len(seongnam.loc[(seongnam['구'] == '분당구') & (seongnam["ada"] ==
print("수정구 :", len(seongnam.loc[(seongnam['구'] == '수정구') & (seongnam["ada"] ==
print("중원구 :", len(seongnam.loc[(seongnam['구'] == '중원구') & (seongnam["ada"] ==
```

```
분당구 : 19
수정구 : 9
중원구 : 7
```

```
In [4]: seongnam["구"].value_counts()
```

```
Out[4]: 분당구      38  
수정구      18  
중원구      17  
Name: 구, dtype: int64
```

```
In [5]: no_accident = seongnam[seongnam["ada"]==0]  
yes_accident = seongnam[seongnam["ada"]==1]
```

```
In [6]: print("[사고가 안일어날 지역]")  
print(no_accident.drop(["위도", "경도", "ada"],axis=1).mean())
```

```
[사고가 안일어날 지역]  
신호등개수      3.184211  
옐로우카펫      0.921053  
울타리길이     871.342105  
횡단보도폭      6.110526  
횡단보도개수     3.157895  
CCTV설치대수    3.631579  
어린이놀이시설 개수    7.157895  
안전표지판 개수    3.500000  
과속방지턱      14.500000  
dtype: float64
```

```
In [7]: print("[사고가 일어날 지역]")  
print(yes_accident.drop(["위도", "경도", "ada"],axis=1).mean())
```

```
[사고가 일어날 지역]  
신호등개수      5.342857  
옐로우카펫      0.914286  
울타리길이     749.028571  
횡단보도폭      6.234286  
횡단보도개수     5.342857  
CCTV설치대수    4.428571  
어린이놀이시설 개수    7.314286  
안전표지판 개수    5.628571  
과속방지턱     19.514286  
dtype: float64
```

지도 시각화

```
In [8]: import folium as f  
from folium.plugins import MiniMap  
  
m = f.Map(  
    location =[37.407366, 127.116217], # 성남시 중심  
    zoom_start = 12,  
    tiles = 'cartodbpositron'  
    ) # 0 ~ 18 허용되는 범위 조금씩 확대됨  
  
# 미니맵 추가  
minimap = MiniMap()  
minimap.add_to(m)
```

```
Out[8]: <folium.plugins.minimap.MiniMap at 0x2c79061b070>
```

```
In [9]: import geopandas as gpd
import pyproj
from shapely.ops import transform

# shp 파일 읽어오기
elementary = gpd.read_file("C:\WMyNeuralNetwork\공모전\스쿨존 내 사고 방지\경기도\초등")

transformer = pyproj.Transformer.from_crs("EPSG:5179", "EPSG:4326", always_xy=True)
elementary_transformed = elementary.geometry.apply(lambda geom: transform(transformer.transform, geom))
elementary.geometry = elementary_transformed

elementary.crs = "EPSG:4326" # 이걸 해줘야 정상적으로 나옴

# 스타일 설정
elementary_style = lambda x: {
    'color': 'black',
    'weight': 1,
    'fillOpacity': 0.6
}

f.GeoJson(elementary.boundary, name='json_data', style_function=elementary_style).add
```

Out[9]: <folium.features.GeoJson at 0x2c7930709d0>

```
In [10]: # shp 파일 읽어오기
df = gpd.read_file('C:\WMyNeuralNetwork\공모전\스쿨존 내 사고 방지\경기도\WARD_ADM_SEC')
seongnam = df[df['SGG_NM'].str.contains('성남시')]

# folium 지도에 나타내기 위해 좌표계 변경
transformer = pyproj.Transformer.from_crs("EPSG:5179", "EPSG:4326", always_xy=True)
seongnam_transformed = seongnam.geometry.apply(lambda geom: transform(transformer.transform, geom))
seongnam.geometry = seongnam_transformed

seongnam.crs = "EPSG:4326"

# 스타일 설정
seongnam_style = lambda x: {
    'fillColor': 'green' if x['properties']['SGG_NM'] == '성남시 분당구' else 'blue',
    'color': 'black',
    'weight': 1,
    'fillOpacity': 0.2
}

f.GeoJson(seongnam, style_function=seongnam_style).add_to(m) # 행정구역 경계 데이터 포
```

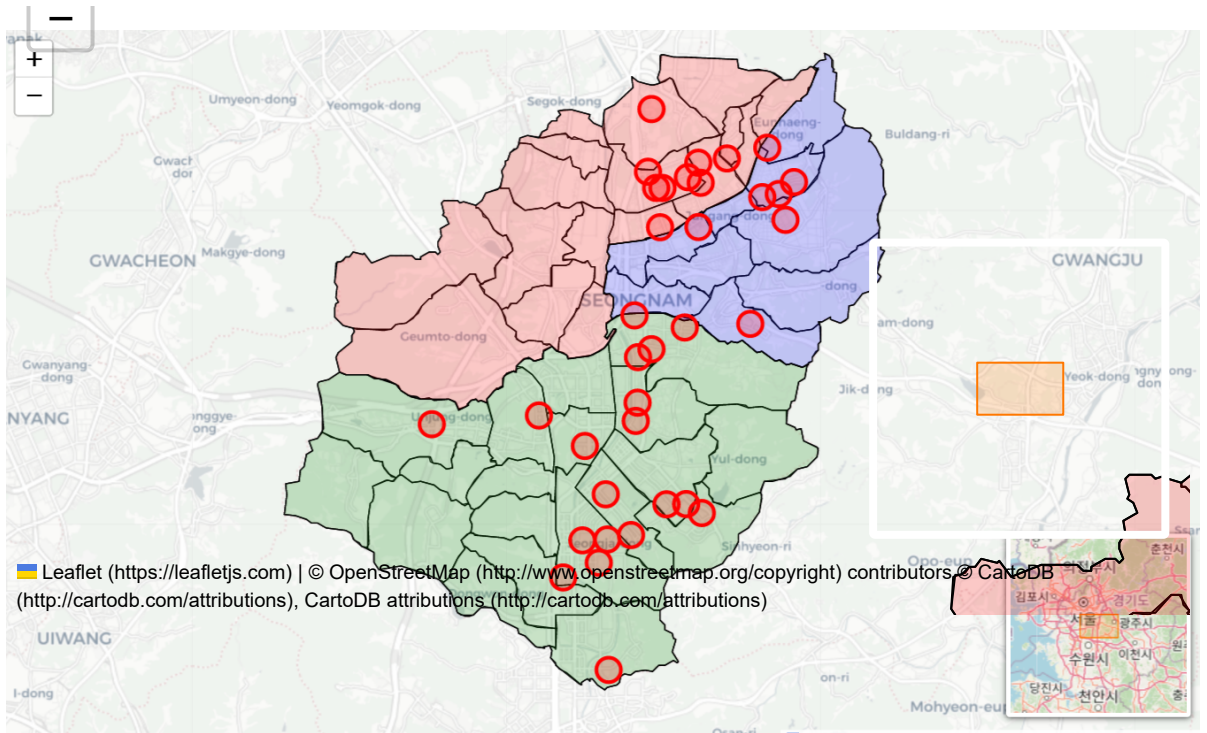
Out[10]: <folium.features.GeoJson at 0x2c794915d00>

```
In [11]: for fac, lat, lon in zip(yes_accident['대상시설명'], yes_accident['위도'], yes_accident['경도']):
    f.Circle([lat, lon],
            radius = 300,
            color='red',
            fill_color='red',
            tooltip= fac,
            ).add_to(m)

m
```

Out[11]: Make this Notebook Trusted to load map: File -> Trust Notebook





In []: