
Silver Zone 0대로 괜찮은가?

팀명 : Silver Town

김경목, 맹지호, 민병창, 신제우, 윤규현

목차

1. 구성원 소개 및 역할
2. 문제 제기 및 필요성
3. 결과 및 활용 라이브러리
4. 결론
5. 한계점 및 느낀점
6. QnA



1. 구성원 소개 및 역할

김경목

- 노인 교통 사고 위치 데이터 전처리
- 노인 교통사고, 보호구역 데이터 시각화

맹지호

- 노인 교통 사고 위치 및 법규위반 데이터 전처리
- 노인 복지구역 데이터 전처리
- 노인 교통사고 지도 데이터 시각화

민병창

- 크롤링
- 노인 교통사고 위치 및 법규위반 데이터 전처리
- 노인 교통사고 및 복지구역 데이터 시각화

신제우

- 크롤링
- 노인 교통사고 위치 데이터 전처리,
- 고령화 지표 데이터 시각화

윤규현

- 노인 교통사고 위치 데이터 전처리,
- 노인 교통사고, 보호구역 데이터 시각화

Project overview

“스쿨존은 알아도 실버존은 몰라요”… 관심 사각지대



Data 설명

전체 및 노인 인구 데이터

년도	총 인구	65세 이상
2010	49554	5366
2020	51836	8152
2022	51628	9018
2025	51448	10585
2030	51199	13056
2035	50869	15289
2040	50193	17245
2050	47359	19004
2060	42617	18683
2070	37656	17473

지역구 별 노인 인구

동별	소계
종로구	152211
중구	130785
용산구	233284
성동구	288234
광진구	351252
동대문구	353601
중랑구	390140
성북구	441984
강북구	297702
도봉구	313989

교통사고 법규위반

중앙선침범	신호위반	안전거리미확보	불법유턴
0	2	5	0
0	4	2	0
0	3	2	0
0	3	2	0
0	7	2	0
0	4	2	0
0	3	1	0
0	0	1	0
0	4	0	0
0	2	0	0

노인 부상 정도

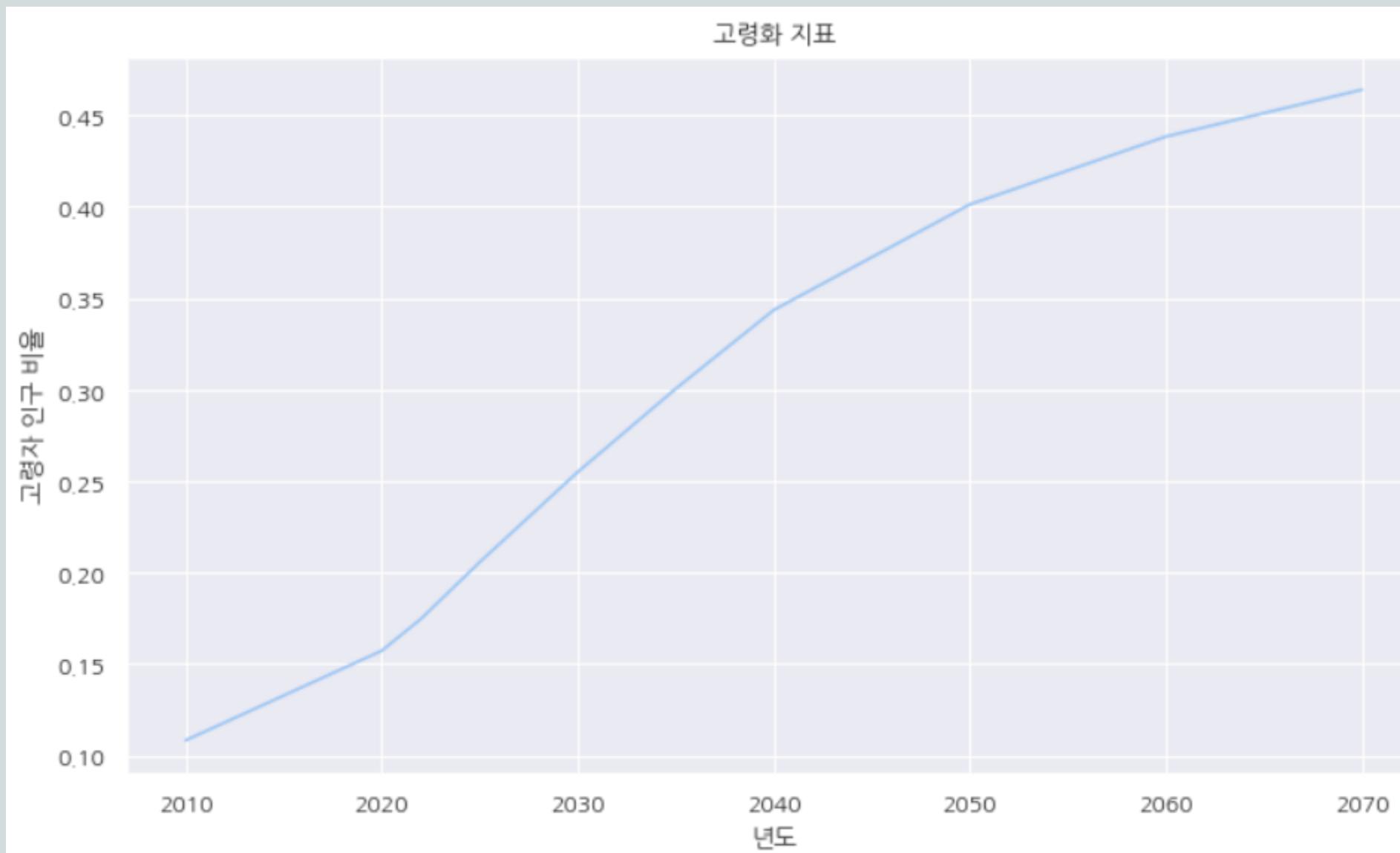
건수	사망	중상	경상	부상
16	0	5	10	1
18	0	5	12	1
14	1	2	10	1
17	0	3	14	0
12	0	3	8	1
11	0	5	6	0
13	0	3	9	1
3	0	1	2	0

보행자 교통사고 사망자

	17년	18년	19년	20년	21년
전체 사망자	1675.000000	1487.000000	1302.000000	1093.000000	1018.000000
노인 사망자	906.000000	842.000000	743.000000	628.000000	601.000000
노인 사망비율	54.089552	56.624075	57.066052	57.456542	59.037328

2. 문제제기 및 필요성

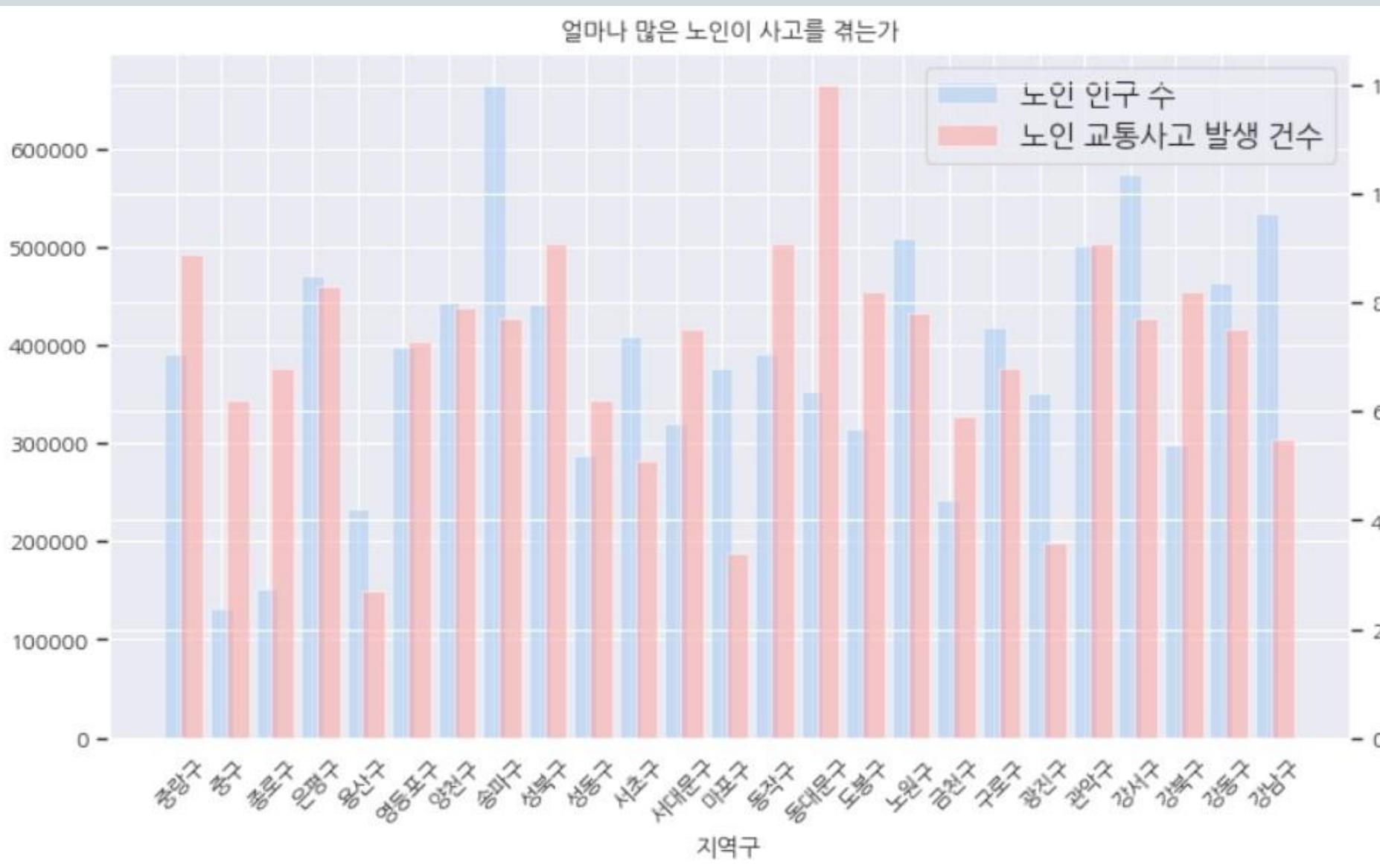
한국 초고속 고령화 진행 중



```
plt.title("고령화 지표")
plt.plot(aging_data["년도"], aging_data["비율"])
plt.xlabel("년도")
plt.ylabel("고령자 인구 비율")
plt.show()
```

2. 문제제기 및 필요성

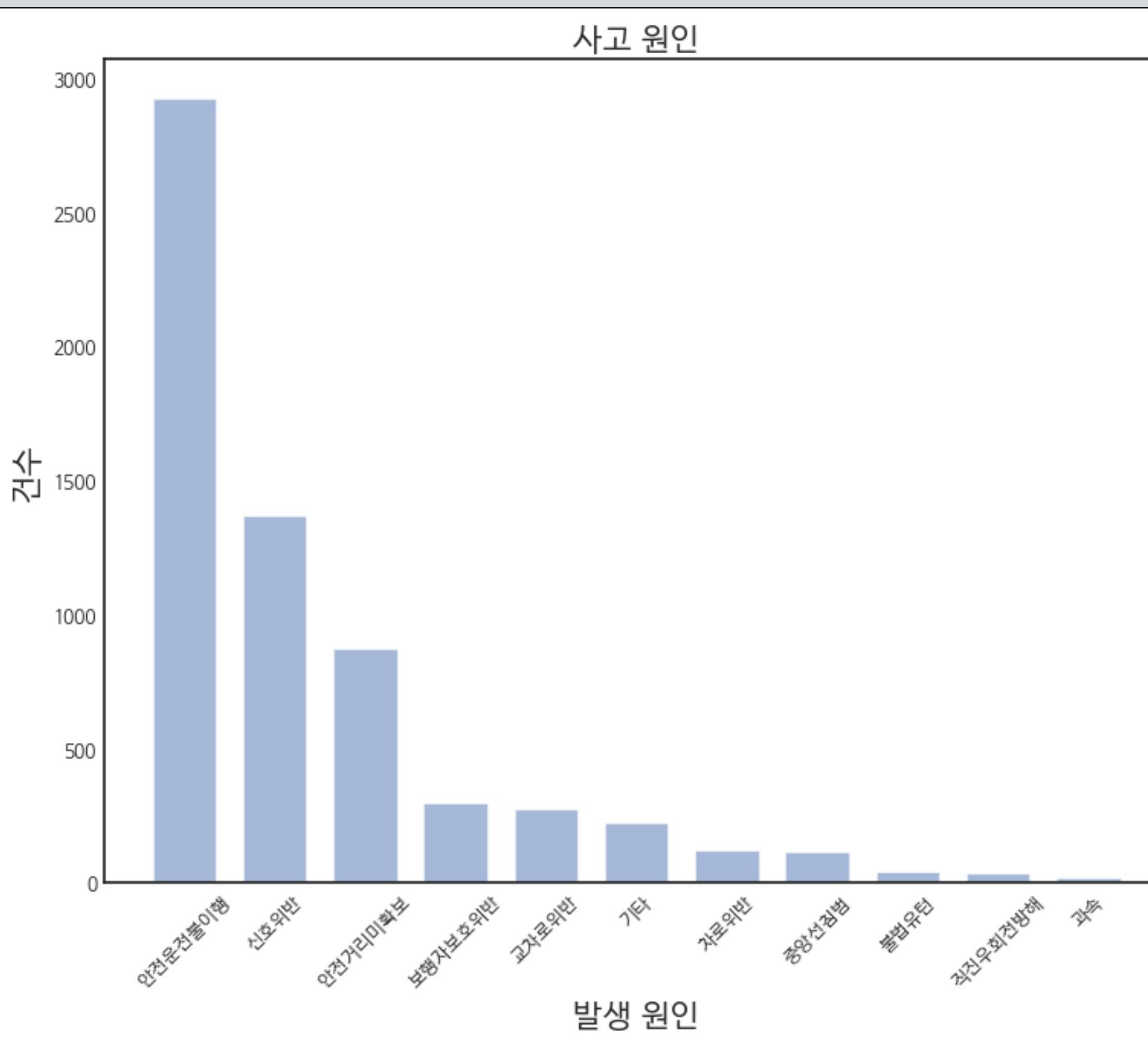
서울 지역구 별 노인 인구 대비 교통 사고 발생 건수



```
fig, ax3 = plt.subplots()  
  
bar_width = 0.35  
alpha = 0.5  
plt.figure(figsize = (15,10))  
N = len(data_older_population['동별'].unique())  
index = np.arange(N)  
p3 = ax3.bar(data_older_population['동별'], data_older_population["소계"], alpha=alpha, color='b', width=0.5)  
  
ax4 = ax3.twinx()  
p4 = ax4.bar(index + bar_width, data_older_accident_number['발생건수'], alpha=alpha, color='r', width=0.5)  
  
ax3.legend((p3, p4), ('노인 인구 수', '노인 교통사고 발생 건수'), fontsize = 12, loc='upper right')  
ax3.set_title("얼마나 많은 노인이 사고를 겪는가")  
  
plt.show()
```

2. 문제제기 및 필요성

노인 교통사고 법규위반

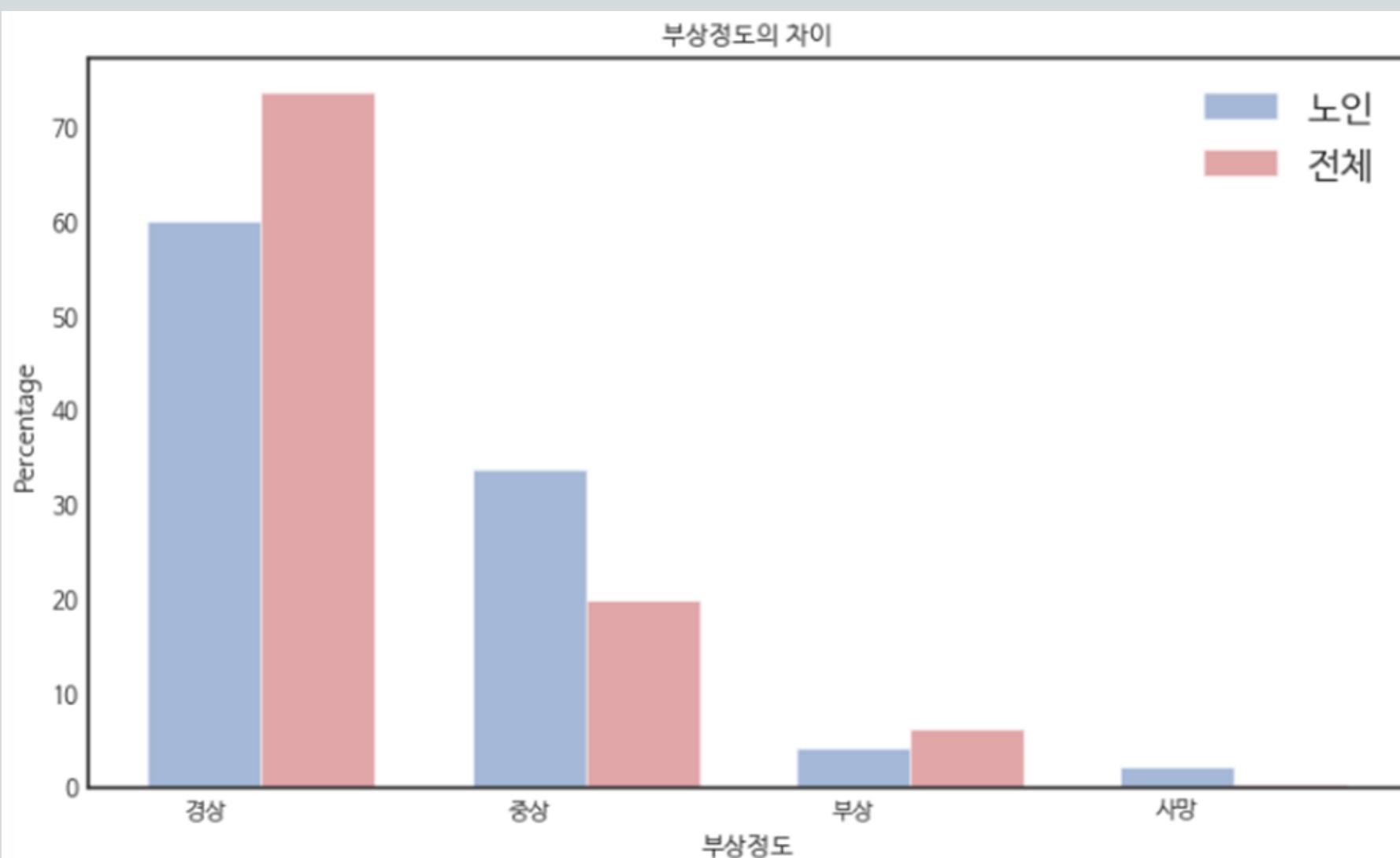


```
law_csv = pd.read_csv('C:\\playdata\\23_02_01_python_mini_project\\law.csv')
plt.figure(figsize = (9,7))
law_csv = law_csv.sum()[:11].sort_values(ascending = False)

plt.title('사고 원인',fontsize = 15)
plt.xlabel('발생 원인',fontsize = 15)
plt.ylabel('건수',fontsize = 15)
plt.xticks(rotation = 45)
plt.bar(law_csv.index,law_csv.values,0.7,color = 'b',alpha = alpha,label = 'old')
plt.show()
```

2. 문제제기 및 필요성

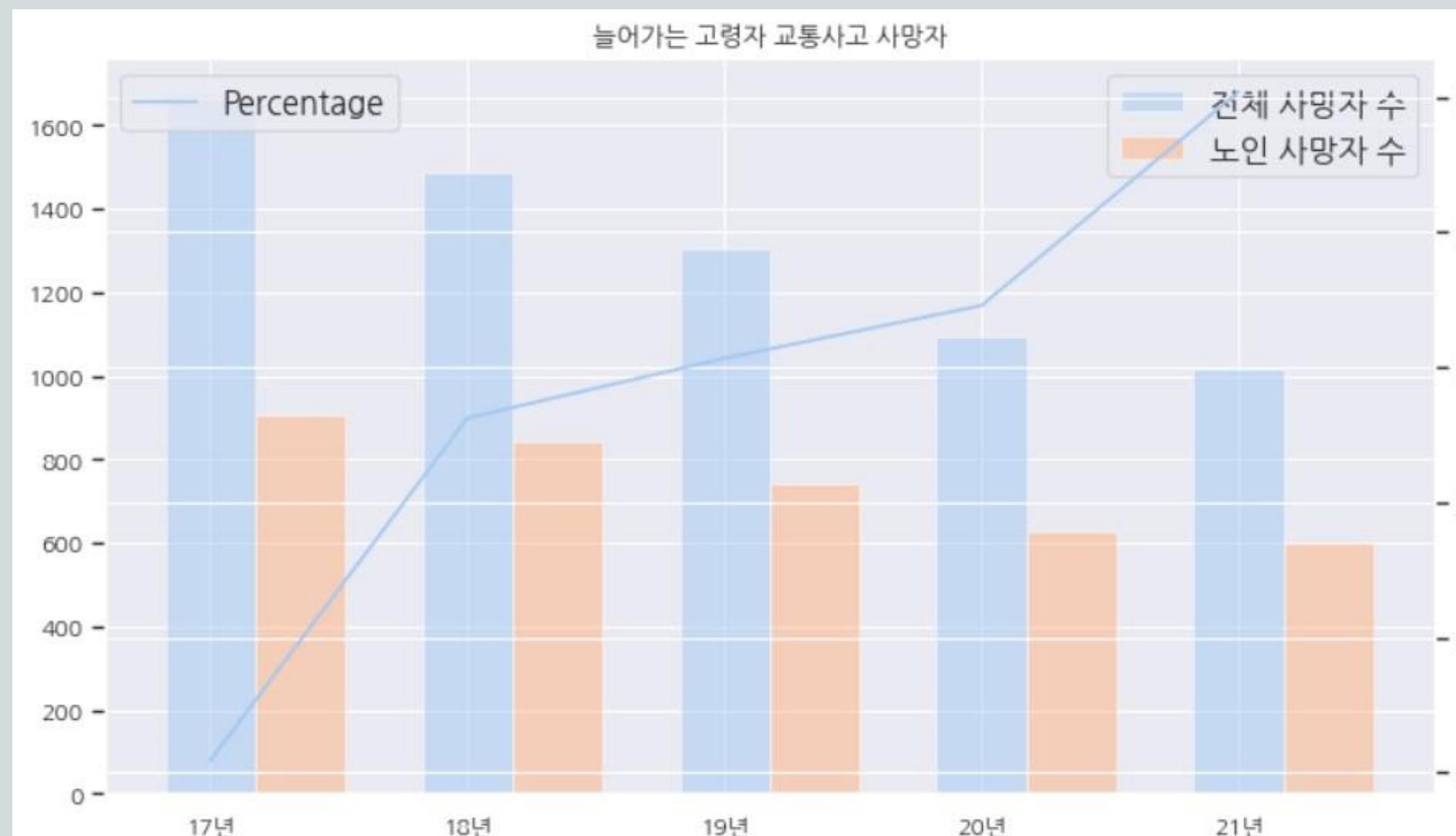
노인과 전체의 부상 정도 차이



```
bar_width = 0.35
alpha = 0.5
plt.figure(figsize = (10,10))
plt.title('부상정도의 차이')
plt.xlabel('부상정도')
plt.ylabel('Percentage')
N = len(acc.unique())
index = np.arange(N)
p1 = plt.bar(index,old_acc.values,bar_width,color = 'b',alpha = alpha,label = 'old')
p2 = plt.bar(index+bar_width,acc_normal.values,bar_width,color = 'r',alpha = alpha,label = 'every')
plt.legend((p1,p2),('노인','전체'),fontsize = 14)
plt.show()
```

2. 문제제기 및 필요성

노인 교통사고 사망률



```
fig, ax1 = plt.subplots()

bar_width = 0.35
alpha = 0.5
plt.figure(figsize = (10,10))
N = len(difference_total_to_old.columns.unique())
index = np.arange(N)
p1 = ax1.bar(difference_total_to_old.columns, difference_total_to_old.loc['전체 사망자'], bar_width, alpha=alpha)
p2 = ax1.bar(index + bar_width, difference_total_to_old.loc["노인 사망자"], bar_width, alpha=alpha)

ax1.legend((p1, p2), ('전체 사망자 수', '노인 사망자 수'), fontsize = 12, loc='upper right')

ax2= ax1.twinx()
p3 = ax2.plot(difference_total_to_old.columns, difference_total_to_old.loc["노인 사망비율"], label="Percentage")
ax2.legend(loc='upper left', fontsize= 12)
ax1.set_title("늘어가는 고령자 교통사고 사망자")
plt.show()
```

3. 분석 결과

I 지역구별 교통사고 발생 건수 및 시각화

II 지역구별 노인 복지 구역 시각화 및 지정 개수

III. 노인 교통사고 발생 지역, 노인보호구역 시각화

Data 설명

지역구 별 교통사고

지역구	발생건수
동대문구	120
관악구	91
성북구	91
동작구	91
중랑구	89
은평구	83

•

•

노인 보호구역

자치구	시설명	위치
종로구	종로노인종합복지관	율곡로19길 17-8
종로구	종로노인종합복지관 무악센터	통일로14길 30
종로구	서울노인복지센터	삼일대로 467
종로구	락희거리	삼일대로 428
종로구	종묘광장공원 동순라길	인의동
중구	신당6동 노인복지관	퇴계로80길 52

•

•

노인 보행자 교통사고 발생 지점

지점명	위치	발생건수
양재전화국입구	서울특별시 강남구 도곡1동 207	18
신논현역 사거리	서울 강남구 강남대로 470	14
을지병원 앞 사거리	서울 강남구 논현동 279	17
대치우성아파트 사거리	서울 강남구 영동대로 230	12
하나저축은행 강남지점(남쪽)	서울 강남구 테헤란로 127	11
구역삼세무서	서울 강남구 역삼동 806	13
서울대청초등학교 부근(북동)	서울 강남구 양재대로55길 28	3
선릉역 사거리	서울 강남구 테헤란로 335	13

•

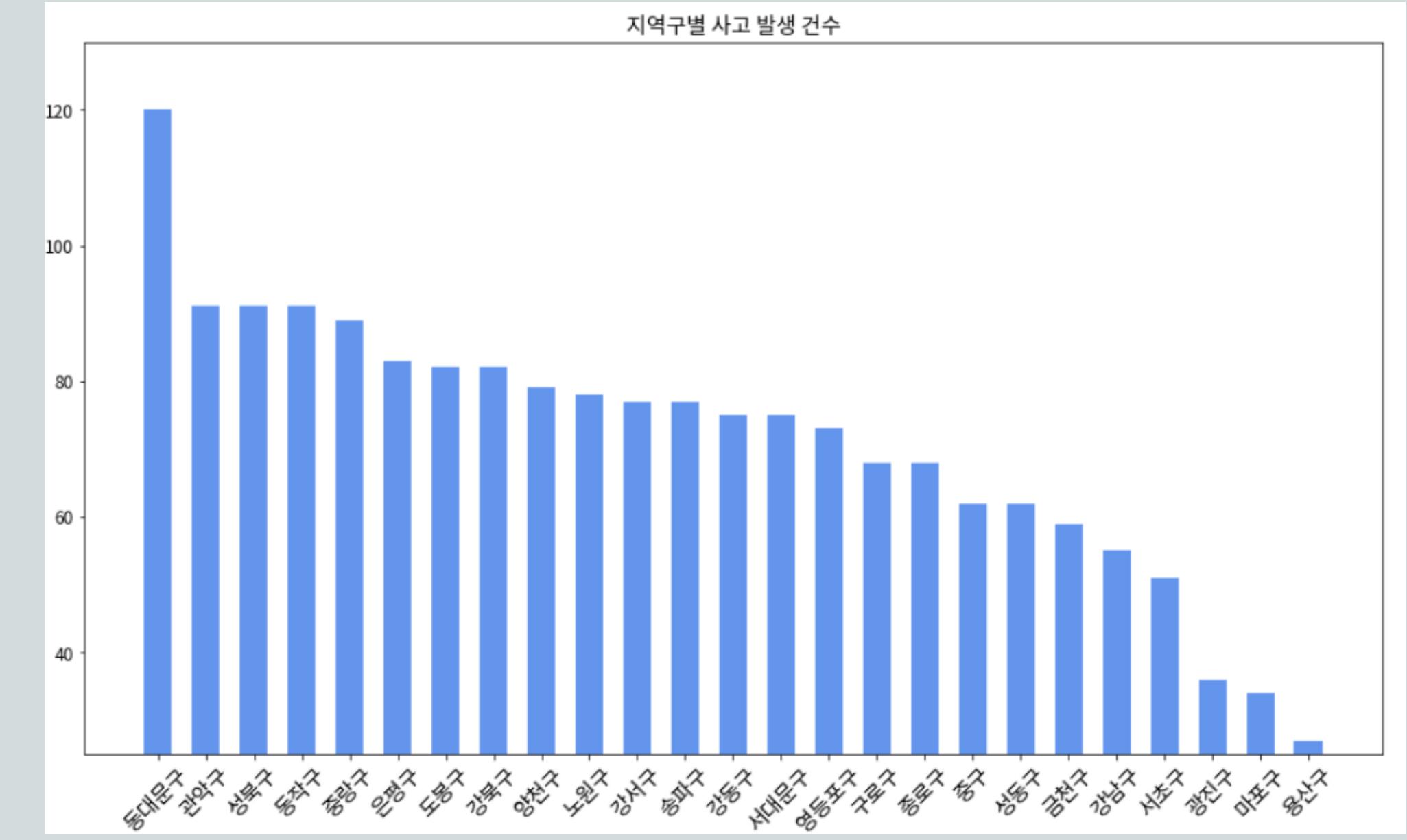
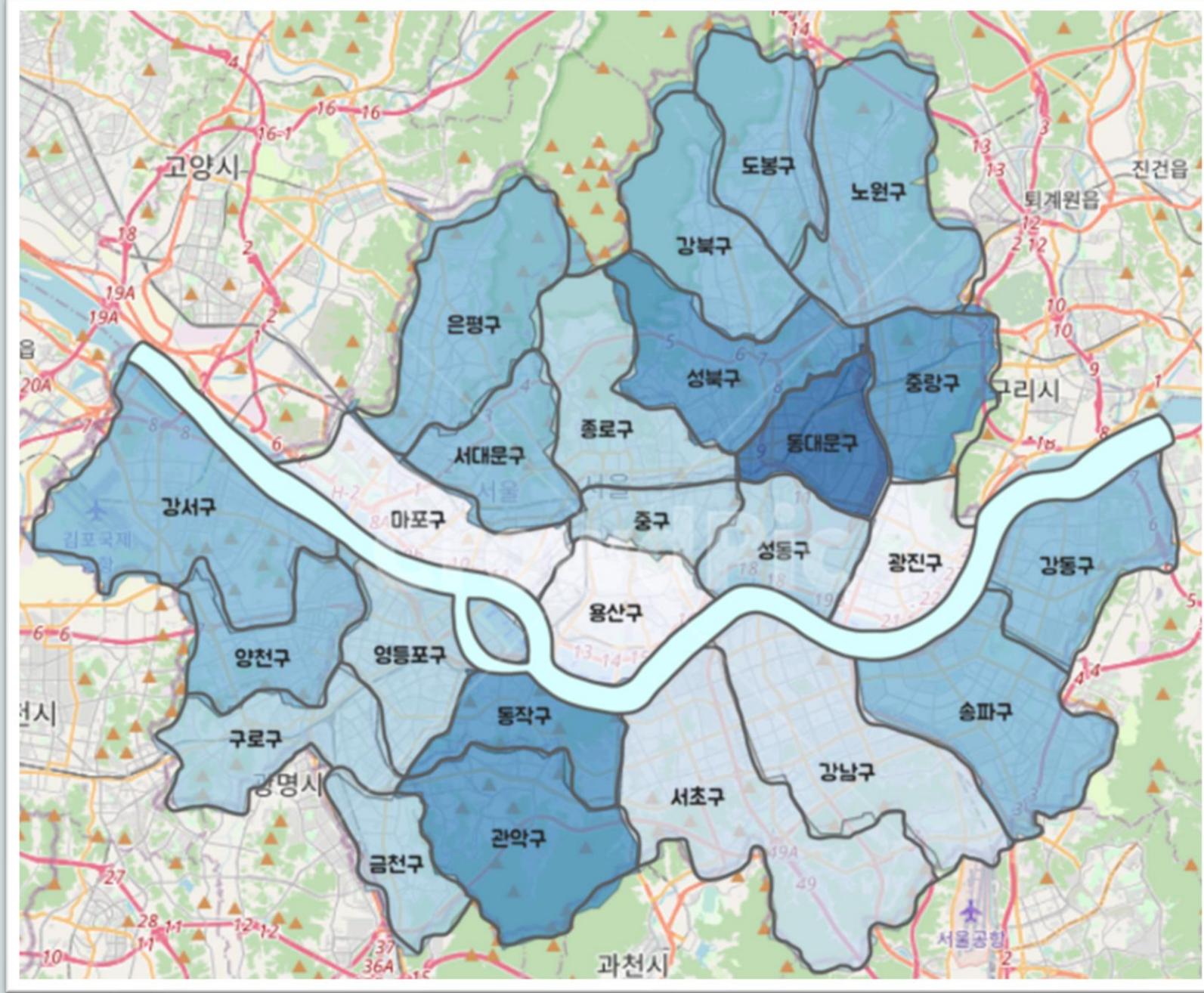
•

개발 환경 및 사용 라이브러리

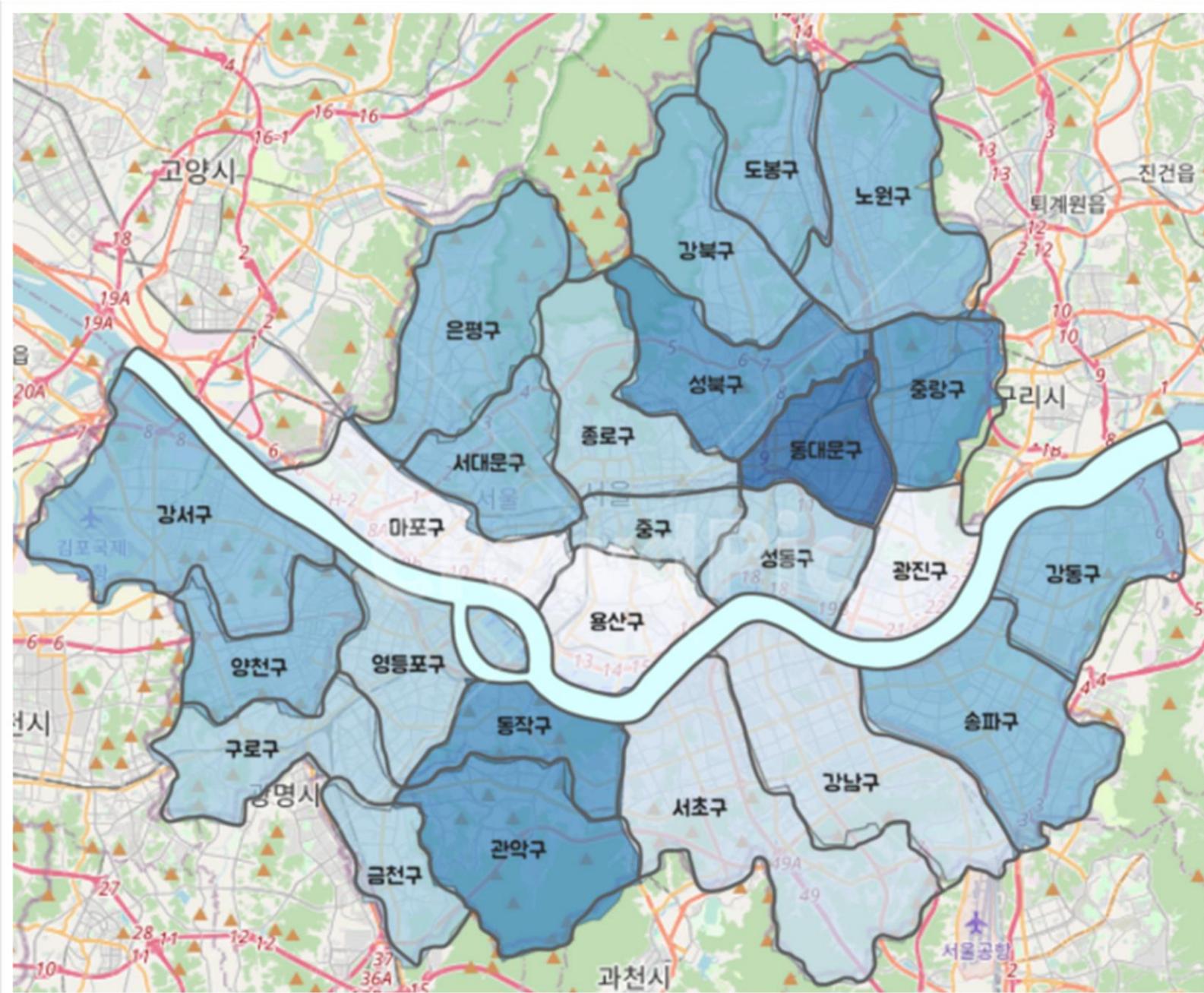


- Pandas
- Matplotlib
- Folium
- Seaborn
- Json
- Request

I. 지역구별 교통사고 발생 건수 및 시각화



개발 환경 및 라이브러리

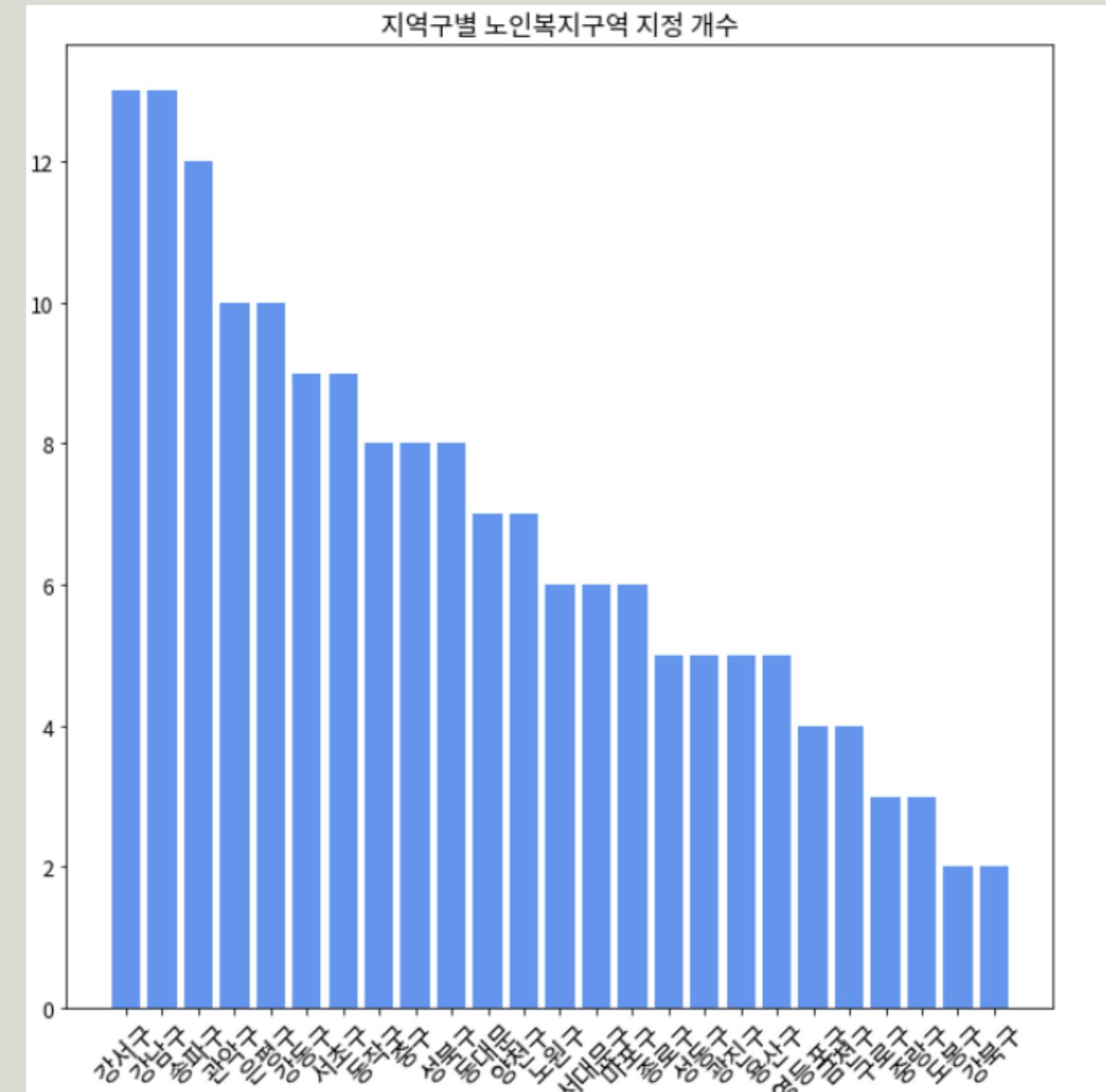
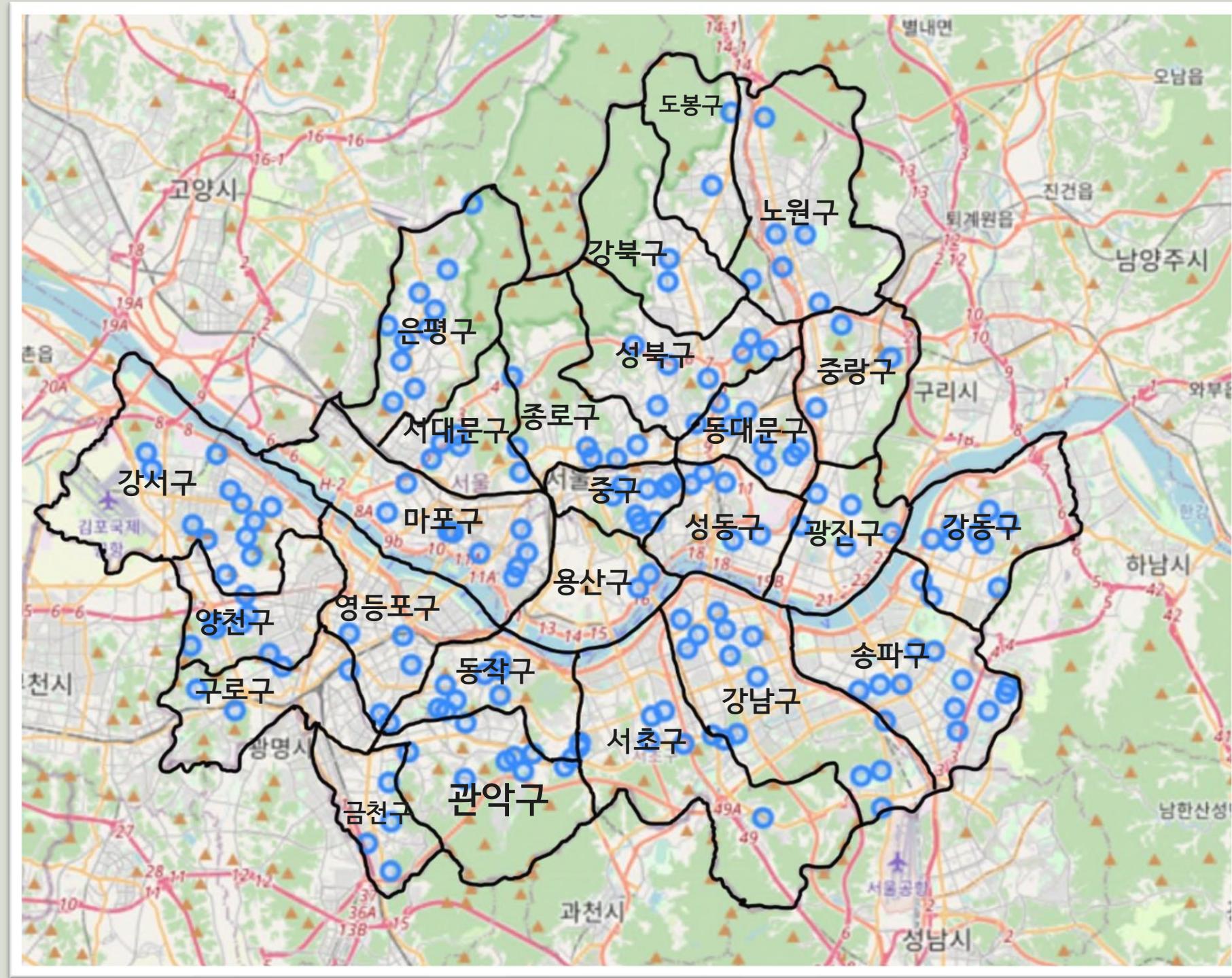


```
import folium
m = folium.Map(location=[37.562225, 126.978555], tiles="OpenStreetMap", zoom_start=11)

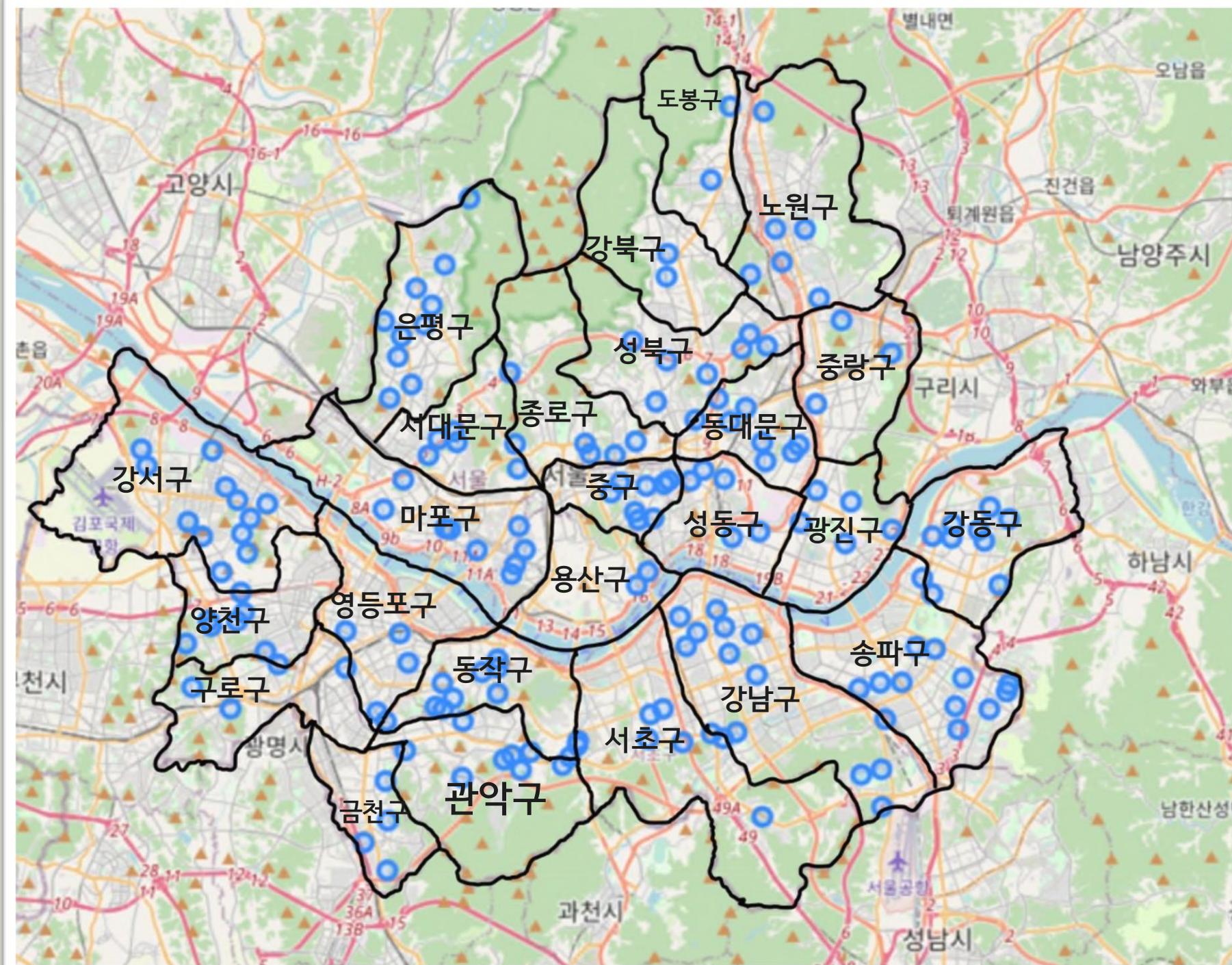
m.choropleth(
    geo_data=state_geo,
    legend_name ='사건 발생 건수',
    data= ped_group,
    columns=['지역구', '발생건수'],
    key_on='feature.properties.name',
    fill_color='Blues',
    fill_opacity=0.7,
    line_opacity=0.3,
    color = 'gray'
)

folium.LayerControl(collapsed = False).add_to(m)
m
```

II. 지역구별 노인보호구역 시작화 및 지정 개수



개발 환경 및 라이브러리



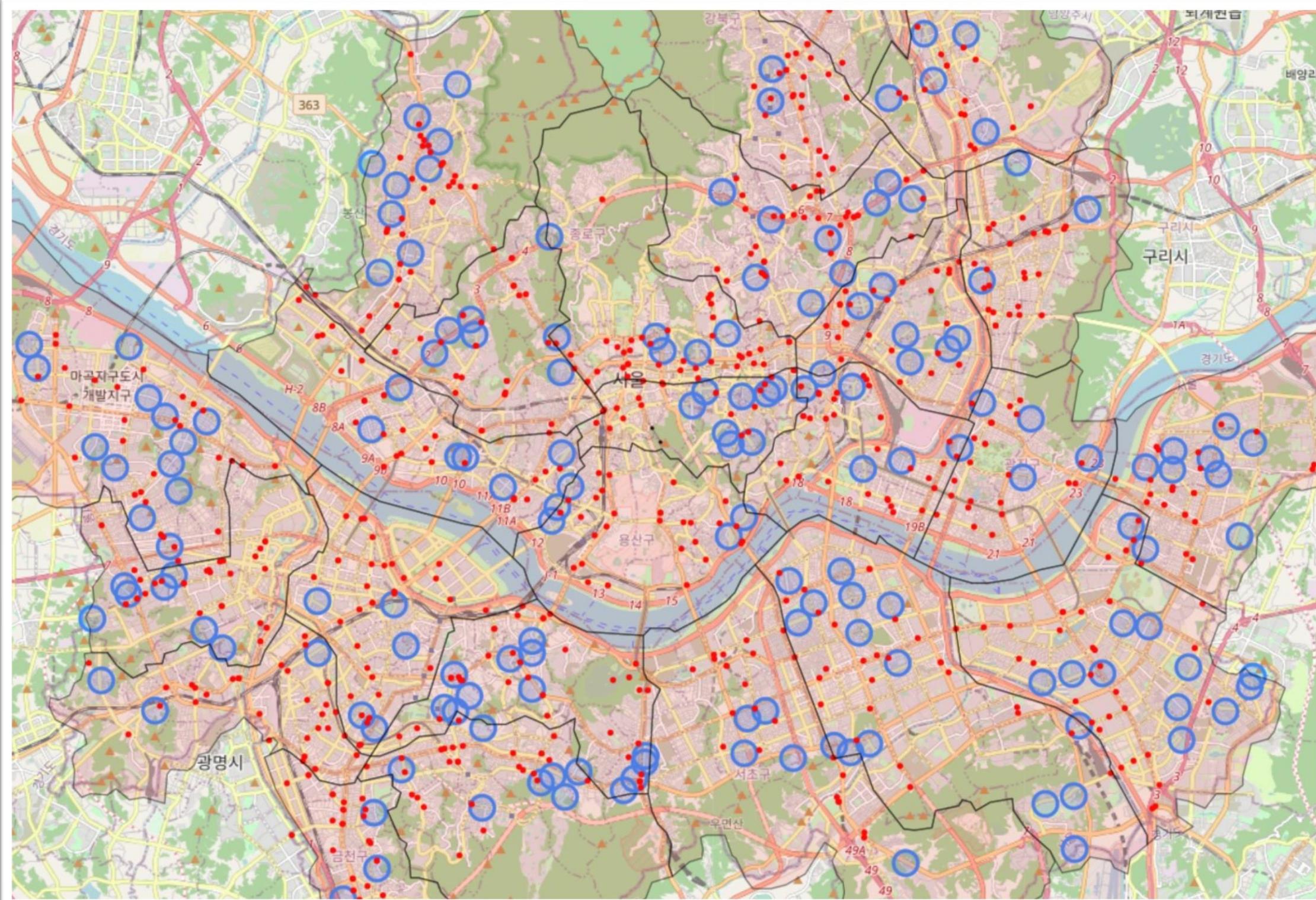
```
import requests, json

def get_location(address):
    try:
        url = 'https://dapi.kakao.com/v2/local/search/address.json?query=' + address
        headers = {"Authorization": "KakaoAK 개인키"}
        api_json = json.loads(str(requests.get(url,headers=headers).text))
        address = api_json['documents'][0]['address']
        rcd = str(address['y']) + ',' + str(address['x'])
        address_name = address['address_name']
        return rcd
    except:
        return 'r,c'

import folium
map_osm = folium.Map(location=[x[13],y[13]],zoom_start=14)
for i in range(len(x)):
    folium.Circle([x[i],y[i]], popup='%s' % silver_address_df.시설명[i],
                 icon=folium.Icon(color='blue', icon='info-sign'),
                 radius=300, fill='red').add_to(map_osm)

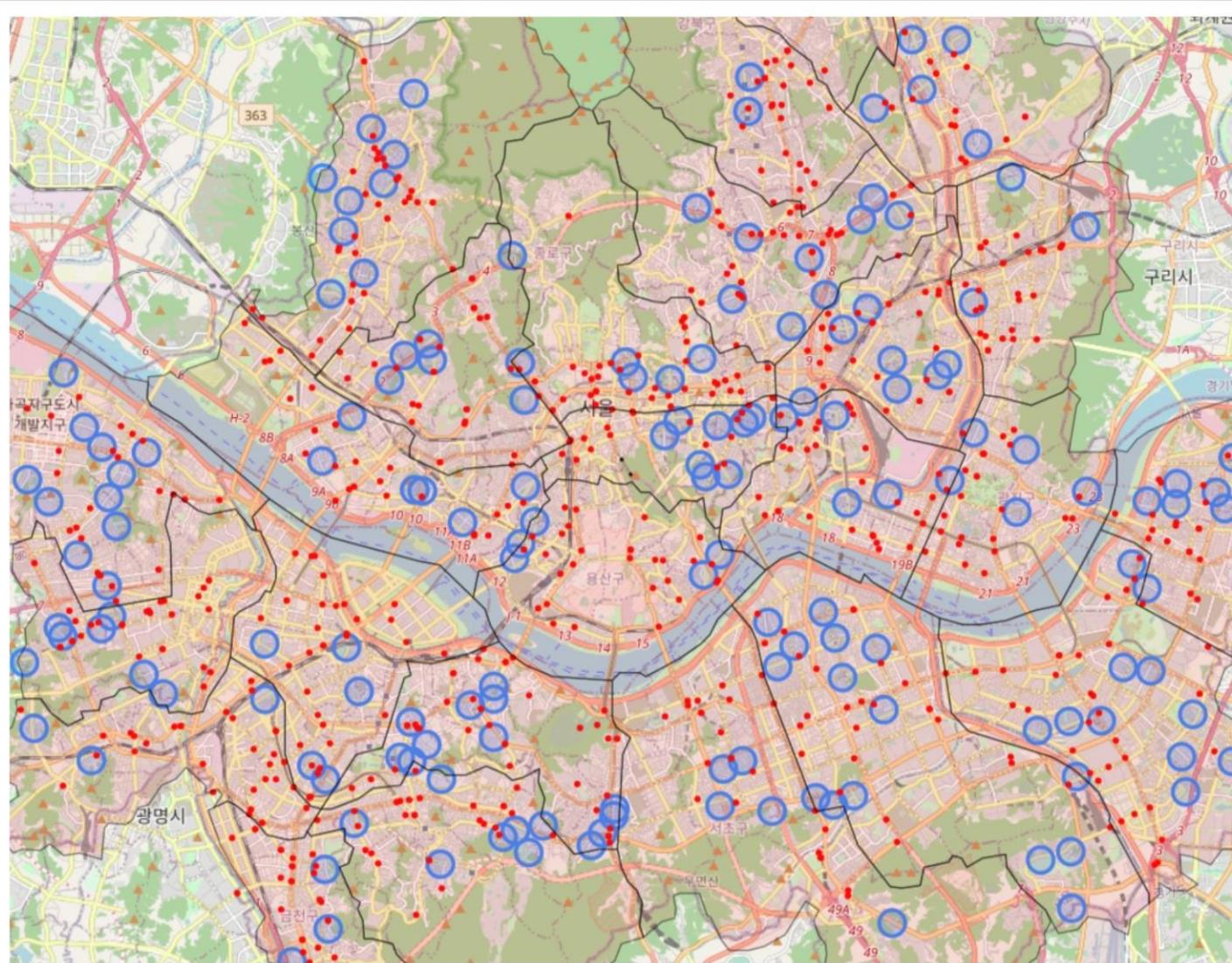
#map저장
map_osm.save('Seoul_SWC.html')
map_osm
```

III. 노인 교통사고 발생지역과 노인 보호구역



노인보호구역: 노인복지센터 근방 300m 이내

개발 환경 및 라이브러리



```
import folium
map_osm = folium.Map(location=[x[13],y[13]],zoom_start=10)
for i in range(len(x)):
    folium.Circle([x[i],y[i]], popup='<p style="width:120px;">%s</p>'%address_df2.시설명[i],
    icon=folium.Icon(color='blue', icon='info-sign'),radius=300,fill='red').add_to(map_osm)
for i in range(len(a)):
    folium.Circle([a[i],b[i]], popup='<p style="width:120px;">%s</p>'%address_df3.지점명[i],
    color='red',radius=30).add_to(map_osm)
# 지도에서 범위 지정

#map저장
folium.Choropleth(
    geo_data=seoul_geojson,
    fill_color="red",
    fill_opacity = 0.1,
    line_opacity=0.6,
).add_to(map_osm)
map_osm.save('Seoul_SWC.html')
```

4. 결론

- 노인 교통사고 발생 시 부상 정도가 심하며 사망률도 4배 이상인데, 교통 안전 개선 필요한 것에 비해 관심이 부족
- 노인 보호구역이 복지센터 근방 300m 이내로 지정되어 있는데 이보다는 통계적인 수치나 데이터를 보고 선정 필요
- Silver zone 외에 사고가 silver zone 내의 사고보다 많기에 보호 구역 외에도 노인을 보호할 수 있는 방안 필요

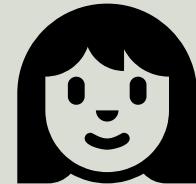
5. 느낀점 및 한계점

데이터 잘 선택하여 수집하는 것이 정말 중요하구나!



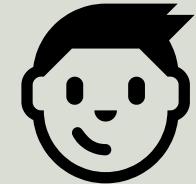
김경목

- 그동안 배웠던 파이썬으로 팀원들과 작업을 함으로써 뜻 깊은 경험이 되었다.
- 문제점들을 해결하는 과정에서 아직 미숙한 부분이 있었다.



맹지호

- 카카오 api를 사용해서 어떤 결과를 얻을 수 있는지 알 수 있었다.
- 그동안 배운 파이썬을 활용해서 시각화를 해낸게 의미있었다.



민병창

- 크롤링을 성공하지 못했던 것이 아쉽다.
- 데이터의 가치를 수 있는 시간이었다.
- 작업 분담의 어려움을 느낄 수 있었다.



윤규현

- 데이터를 가공하는 일이 쉽지 않다는 것을 알았다.
- 파이썬을 활용한 시각화를 통하여 원하는 데이터를 표현할 수 있었다.
- 프로젝트를 통해 같은 정보를 다른 시각으로 보는 것이 의미 있었다.



신제우

- 프로젝트 계획을 잘 구성하지 못했던 게 데이터를 얻는데 영향을 줬다.
- 데이터를 어떻게 다뤄야 원하는 결과로 만들 수 있는지 이해할 수 있었다.
- 데이터마다 특성에 있어서 중점적으로 확인해야하게 다른데 이 부분에서 면밀하게 점검하지 못해 불필요한 데이터 재가공 업무를 해야했다.

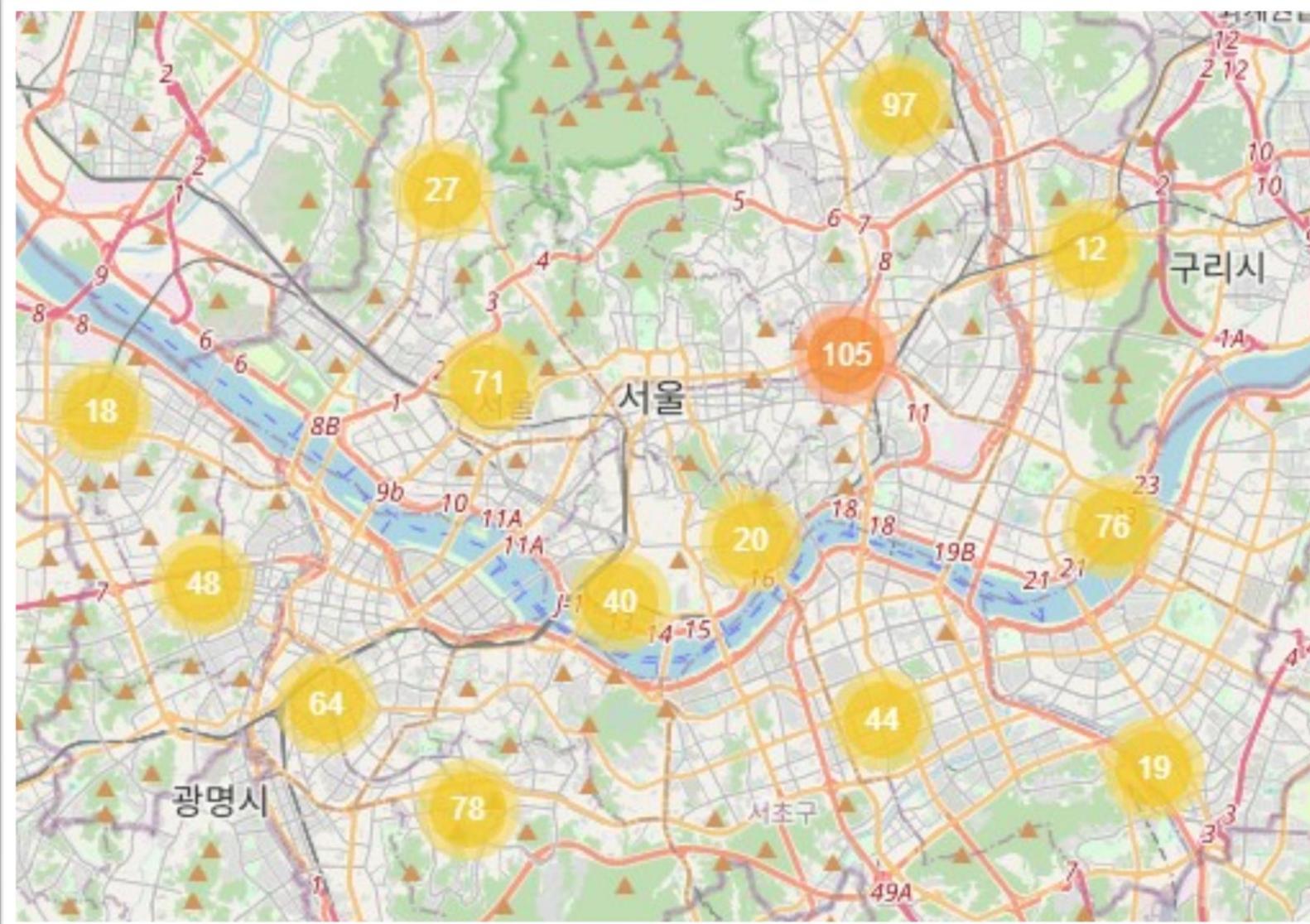
트러블 슈팅

지점명	발생건수	사망자수	보기
수서역사거리	4	0	지점상세보기 >
테헤란아이파크아파트 앞 삼거리	3	0	지점상세보기 >
수서역 부근 삼거리(북쪽)	3	0	지점상세보기 >
신논현역 사거리	2	1	지점상세보기 >
역삼세무서 서쪽 삼거리	3	0	지점상세보기 >
아파트 앞 삼거리	3	0	지점상세보기 >
타워팰리스2차 앞 삼거리(E동)	2	0	지점상세보기 >
경복아파트 사거리(동남)	2	0	지점상세보기 >
일원파출소 앞 삼거리	3	0	지점상세보기 >
소재 숯내교사거리 숯내교사거리	2	0	지점상세보기 >
신사역 오거리	2	0	지점상세보기 >
포이사거리(영동2교 남단)	2	0	지점상세보기 >
역삼e편한세상 앞(101동)	2	0	지점상세보기 >
...	지점상세보기 >

```
mn25" style="position: absolute; overflow: hidden; width: 350px; overflow-wrap: break-word; white-space: normal; display: inline; left: 260px; top: 81px; height: 15px;">신논현역 사거리</span>
14</span>
1</span>
▶<div class="rMateH5__ImageItemRenderer rMateH5__DataGridColumn3 1" style="position: absolute; width: 112px; height: 15px; left: 870px; top: 81px;">...</div> == $0
서울</span>
강남구</span>
```

웹 크롤링을 시도하여 데이터를 획득하려 했으나 웹페이지 저장 방식이 순서가 섞여 있는 방식이었고, 내부 데이터 스크롤시 html 값이 초기화되어 크롤링 어려운 사이트였다.

ipyleaflet을 활용한 시각화



```
from ipyleaflet import Map, Marker, MarkerCluster
marker = []
m = Map(center=(37.5, 127), zoom=5)
for i in range(len(silver_trans_latitude)):
    #데이터에서 뽑아온 위도 경도를 넣고 찍어준다.
    marker.append(Marker(location = (silver_trans_latitude[i], silver_trans_longitude[i])))

marker_cluster = MarkerCluster(markers = marker)
m.add_layer(marker_cluster)
m
```

데이터 출처

TMACS, [서울시 사고 다발 지점 전체 사고 데이터], 2021 사고누적지 사고 지표

TMACS,[서울시 교통약자다발지점 고령자 사고 데이터], 2021 교통약자다발지점 사고지표

통계청, [2022 고령자 통계], 2022.09.29

서울 열린데이터 과정, [서울시 노인 장애인 보호구역 지정 현황], 2022.03.17

행정안전부, [빅데이터 분석으로 노인 교통사고 줄인다.], 2022.10.11

QnA

김경목, 맹지호, 민병창, 신제우, 윤규현