

python을 이용한 빅데이터분석 미니프로젝트

다회용컵의 최적효율을 위한 참여 지역 및 수거함 위치 제안

발표: 5조(Five-Gs)

조장: 최서윤

조원: 강나현, 서유빈, 이소민, 제소현

환경문제가 고조됨에 따라 사람들의 관심사 또한 높아져 가고 있는 요즘, 특히 폭발적으로 증가하는 일회용컵의 사용량은 쉽게 해결되지 않아 사회적 · 환경적으로 많은 문제를 초래하고 있다.

플라스틱은 대부분이 재활용되지 않고 환경오염을 발생시키고 있는데, 이러한 문제를 해결하기 위해서는 근본적인 플라스틱 쓰레기의 발생량을 줄이고, 재사용이 가능한 구조로 전환되어야 할 필요성이 있다.

이러한 문제를 해결하기 위해 2020년 6월 2일 국무 회의에서 자원 재활용법 개정안이 의결됨에 따라 '일회용컵 보증제'가 2022년 6월 본격적으로 시행될 예정이라고 한다.

일회용컵 보증제란 소비자가 보증금이 포함된 가격으로 구매를 한 뒤, 나중에 컵을 반환하면 그 보증금 만큼의 금액을 돌려받는 제도이다. 그러나 똑같이 회수하여 다시 쓰자는 취지의 행동이라면, 플라스틱 사용을 보다 효율적으로 줄일 수 있는 '다회용 공유컵'을 활용하는 제도가 더 선호될 수 있다 생각한다. 그에 따라 우리 5조에서는 부산시 내 다회용 공유컵 사업 추천 지역과 최적의 수거함 위치 제안을 통해 다회용컵 사업을 독려, 이를 통한 환경보호에 앞장서고자 이 프로젝트를 진행하게 되었다.

본 프로젝트에서는 부산시 내 상권 밀집지역과 카페 밀집지역의 분포를 조사하여 최종 후보군을 도출해 낸 다음, 최다 이용 버스정류장 혹은 지하철역, 공원위치와 대규모상점 등의 데이터를 이용해 그 양상을 분석하고 그에 따라 가장 적재적소라 생각되는 위치에 수거함 설치를 제안하고 있다.

부산시 다회용컵 수거함 사업 참여 지역 및 수거함 위치 제안

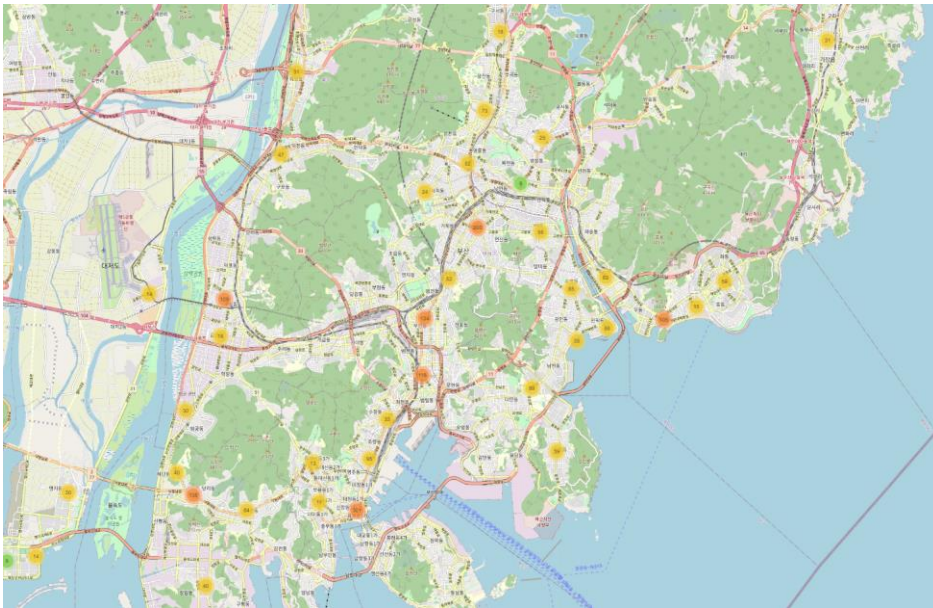
1.상권데이터 분석(상권 밀집지역 도출)

상권좌표 마커를 개수로 지도에 표시

```
m = folium.Map(location = [35.172883, 129.130644], zoom_start = 13)
cluster = MarkerCluster().add_to(m)
nrow = 0

for i in range(df_busan['좌표'].shape[0]):
    try:
        for item in df_busan['좌표'][i].iteritems():
            subtime = item[1].split(',')
            # print(subtime[0], subtime[1])
            folium.Marker(location = [subtime[1], subtime[0]]).add_to(cluster)
    except:
        continue

m
```



부산시 구별 간 경계선 데이터 불러오기

```
geo_path = './data/busan_gu.json'

import json
geo_json = json.load(open(geo_path, encoding='utf-8'))
```

```
geo_json['features'][0]['properties']['name']
```

‘중구’

```
lat = 35.181784
long = 129.052852
(lat, long)
```

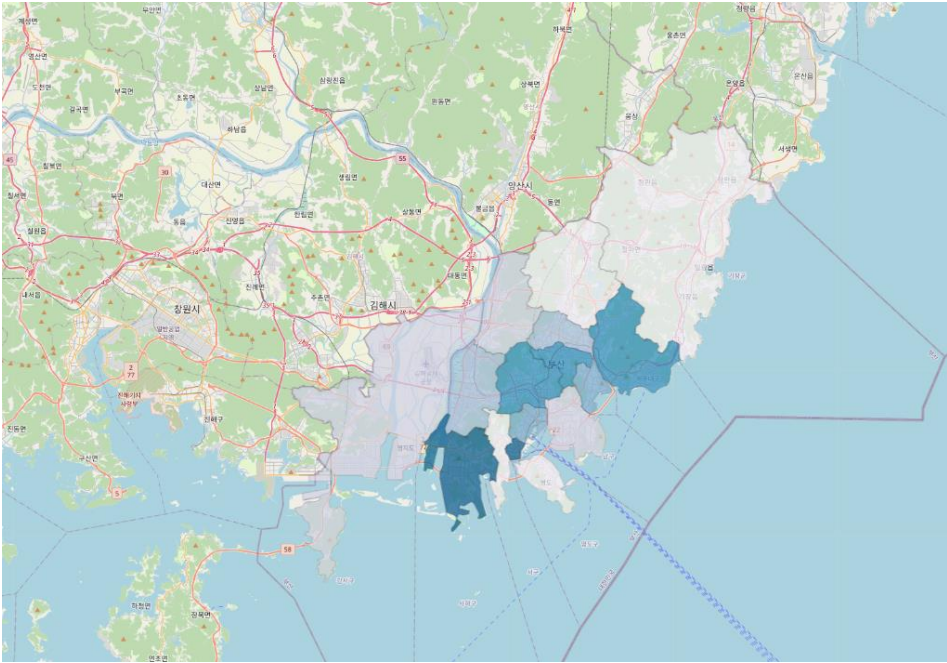
(35.181784, 129.052852)

상권 수를 기준으로 부산시 구 별 밀집도 시각화

```
map = folium.Map([lat, long])

folium.Choropleth(
    geo_data=geo_json,
    name='choropleth',
    data=busan_group,
    columns=['시군구', '상권좌표수'],
    key_on='feature.properties.name',
    fill_color='PuBu',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='상권 수'
).add_to(map)

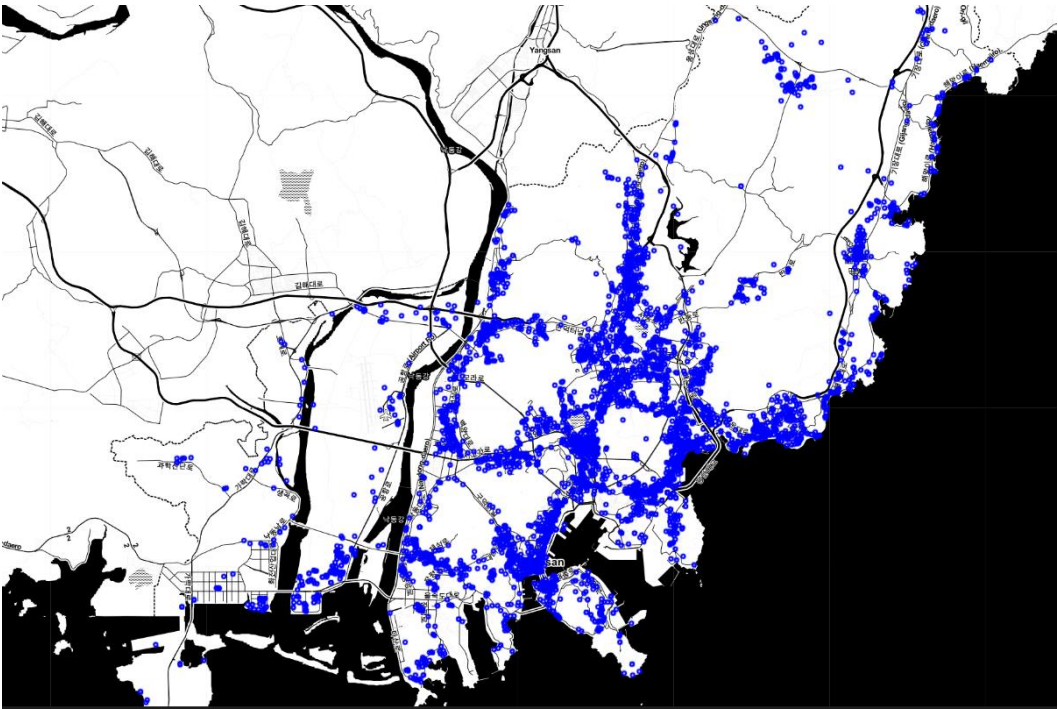
map
```



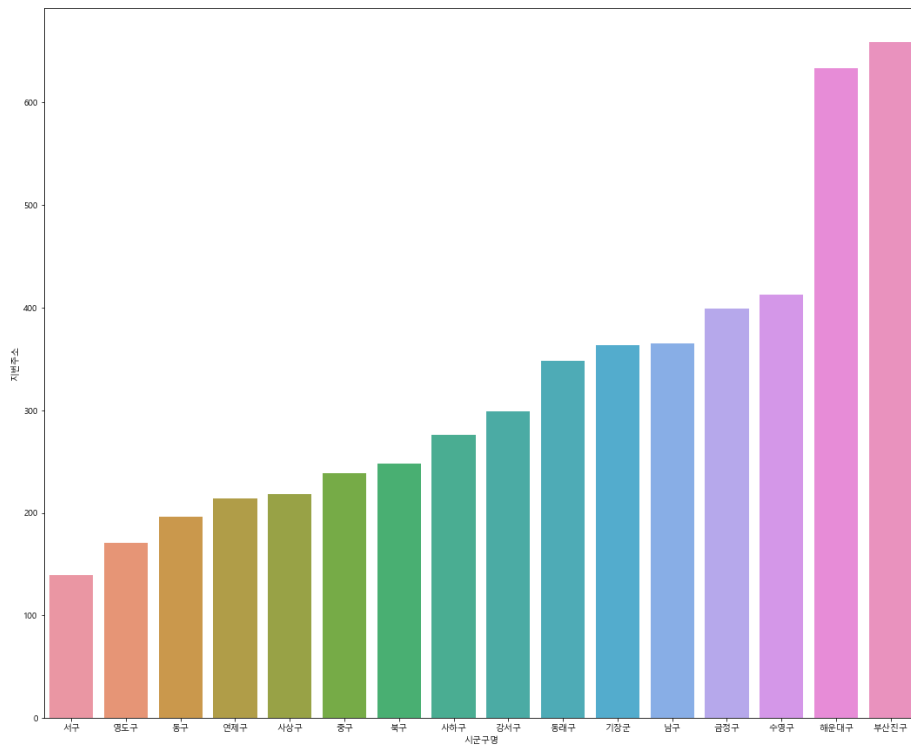
상권 밀집 지역

색이 진할수록 상권이 밀집되어 있다는 것을 의미한다.

2. 카페데이터 분석 (카페 밀집 지역 도출)

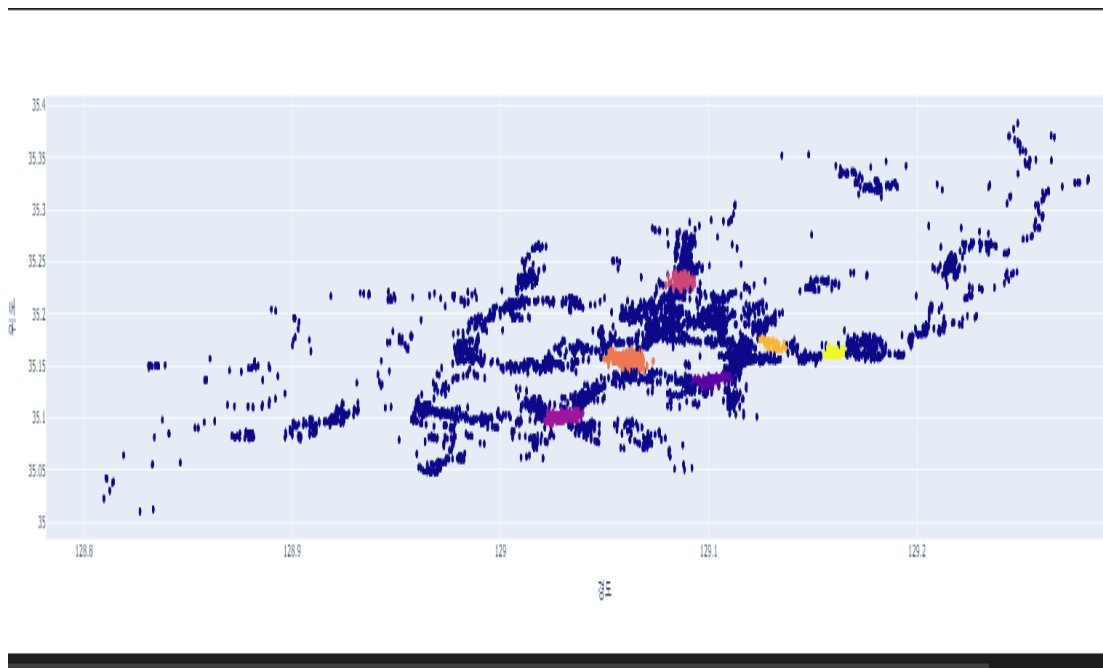


-부산시 내 카페 분포 현황



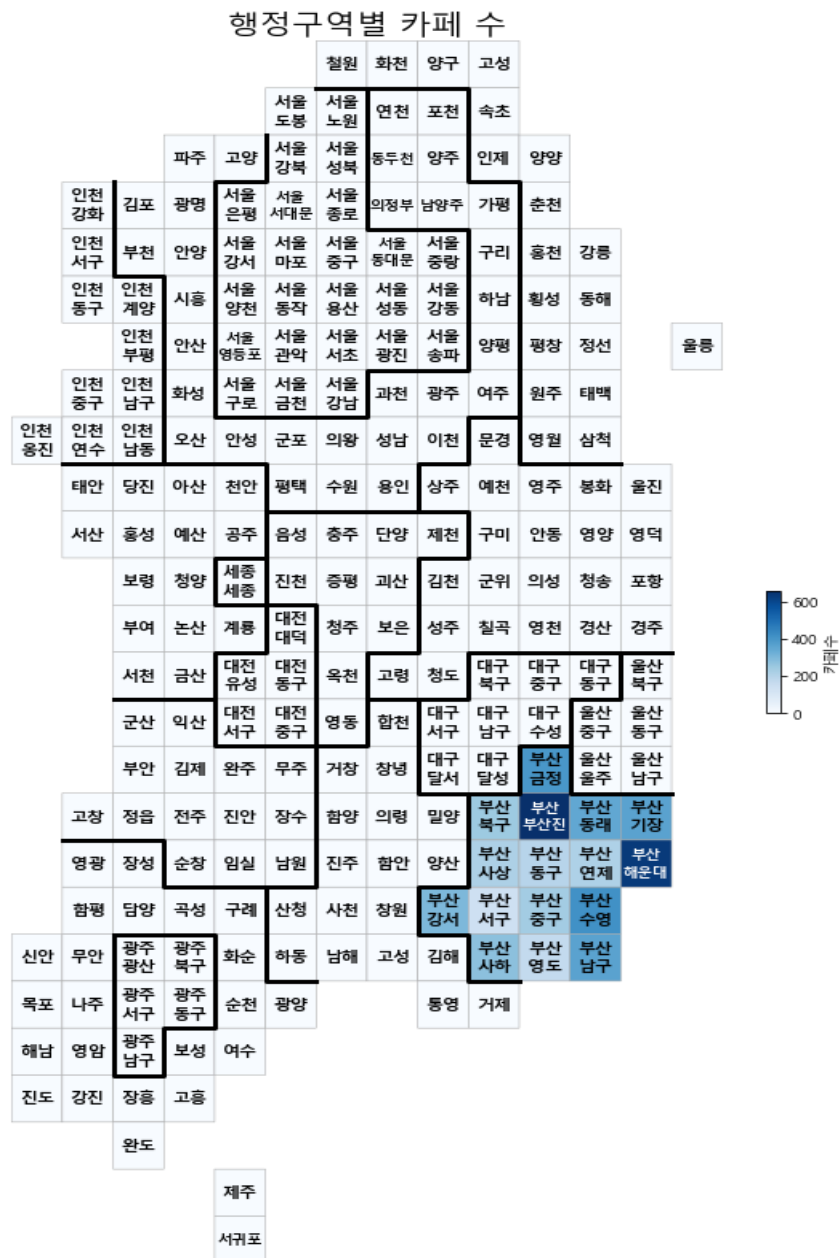
-부산시 내 카페 분포도

-부산진구에 가장 많은 카페가 위치해 있음



- 위경도를 중심으로 카페가 많이 밀집되어 있는 지역을 확인할 수 있다.

- 부산진구, 사하구, 금정구, 해운대구 등



부산시 구 별 상권 밀집도와 카페 밀집도 시각화

부산시 구 별 상권 밀집도와 카페 밀집도 시각화

```
# colors = ['red','blue','green','purple','orange','beige','darkblue','darkgreen','darkpurple','white','pink','lightblue','lightgreen','lightgrey','black','grey']

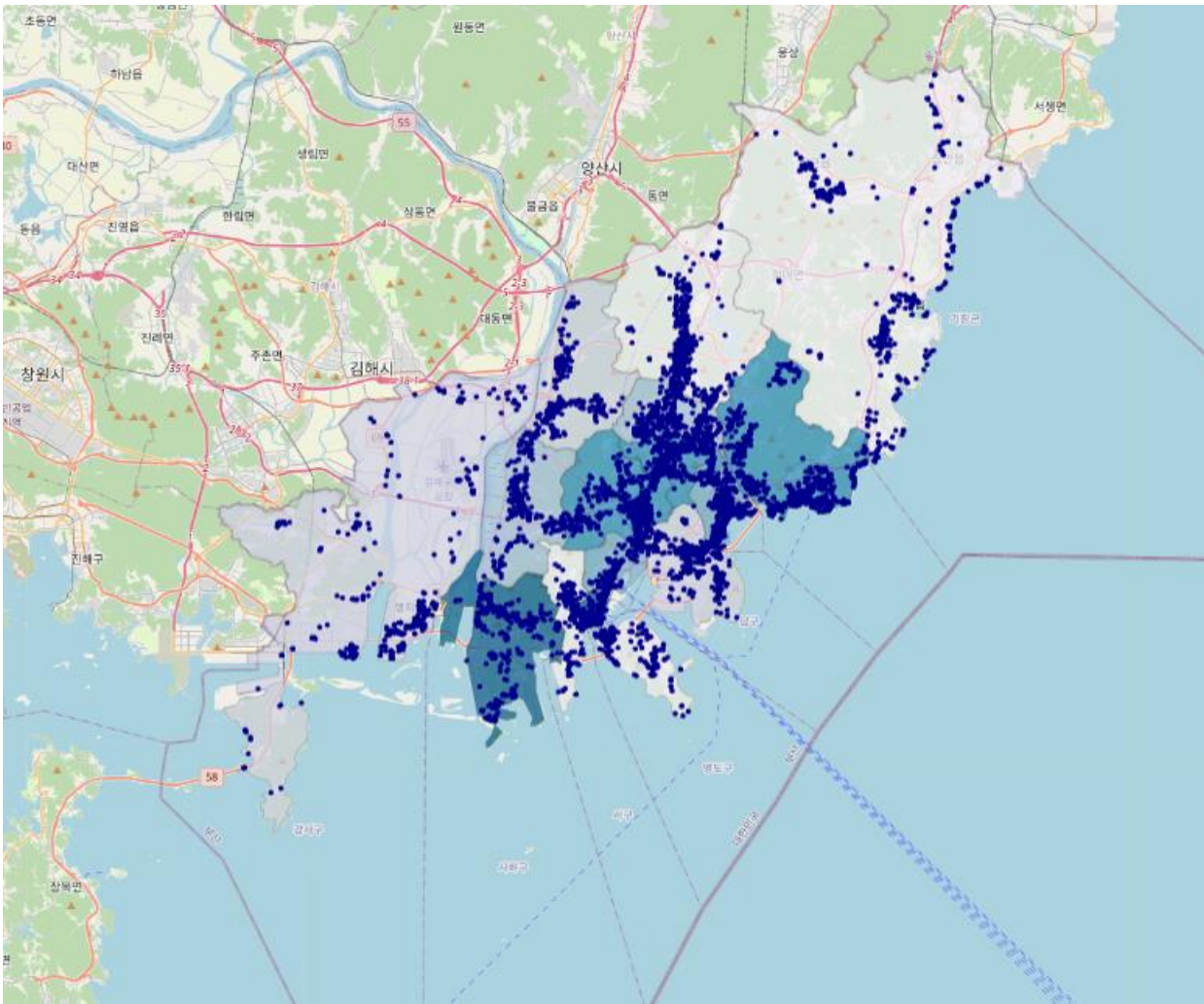
map = folium.Map([lat, long], zoom_start=11)

~folium.Choropleth(
    geo_data=geo_json,
    name='choropleth',
    data=busan_group,
    columns=['시군구', '상권좌표수'],
    key_on='feature.properties.name',
    fill_color='PuBu',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='상권 수'
).add_to(map)

~for i in cafe.index:
    shop_lat = cafe.loc[i, '위도']
    shop_long = cafe.loc[i, '경도']
    title = f'{cafe.loc[i, '도로명주소']}'
    color = 'darkblue'

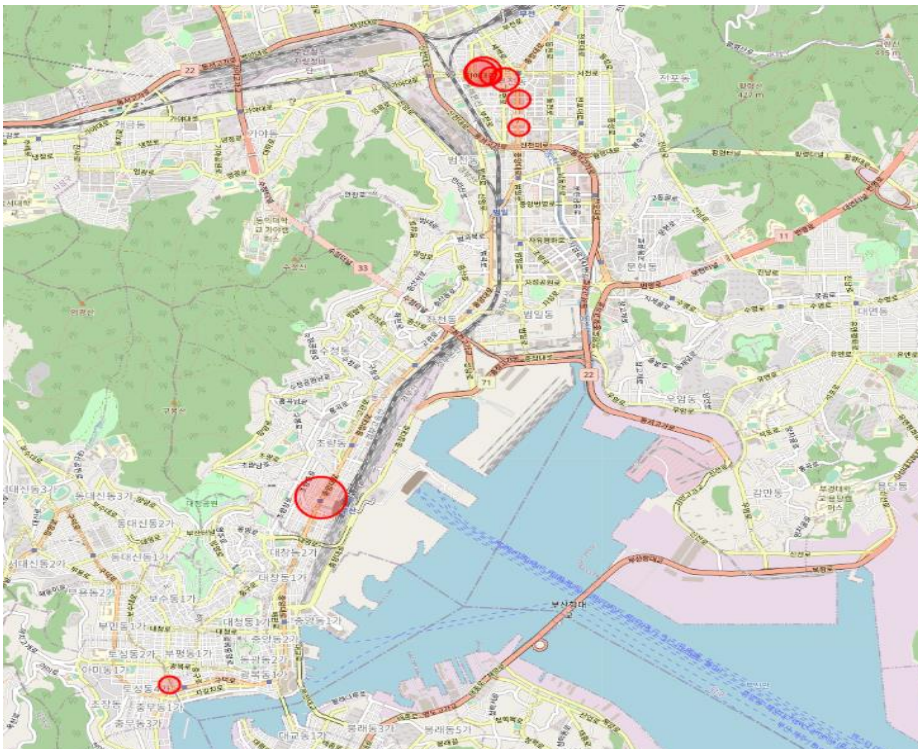
    folium.CircleMarker([shop_lat, shop_long], radius=1, color=color, tooltip=title).add_to(map)

map
```

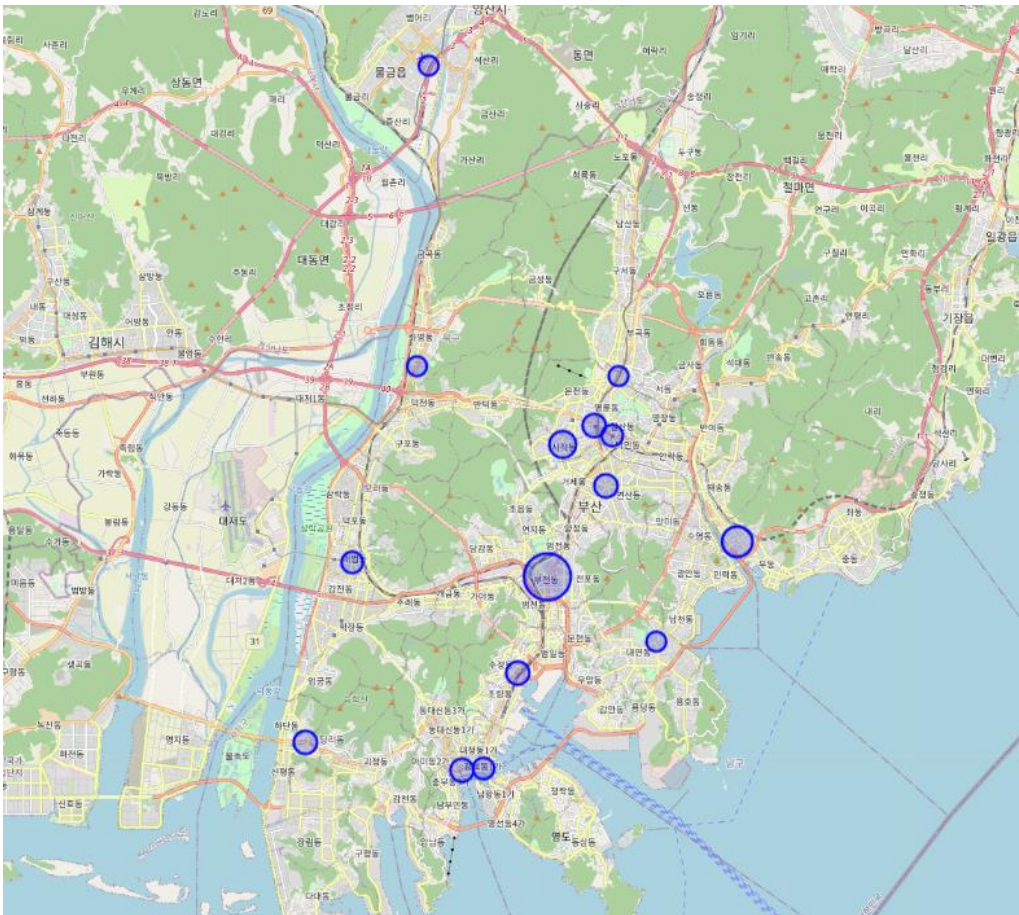



수거함 후보군 제안

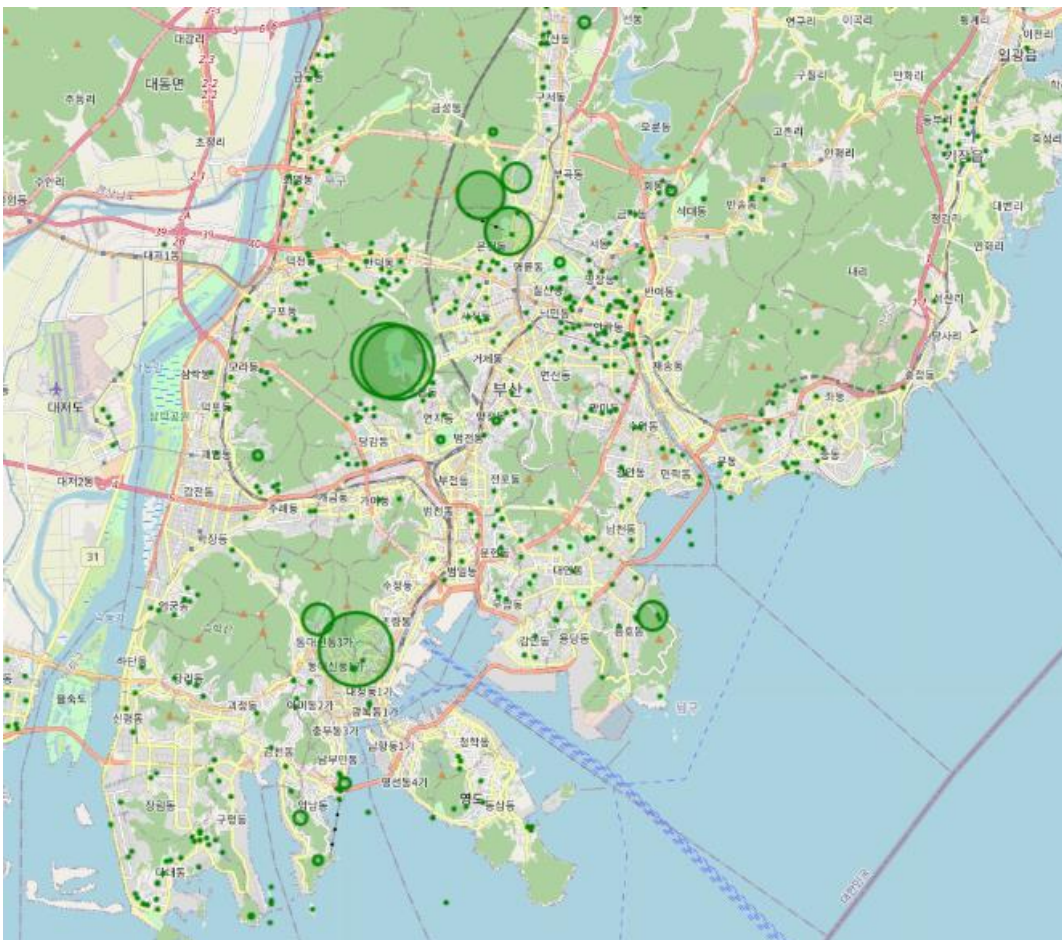
-버스, 지하철, 공원, 대규모상점 데이터 이용



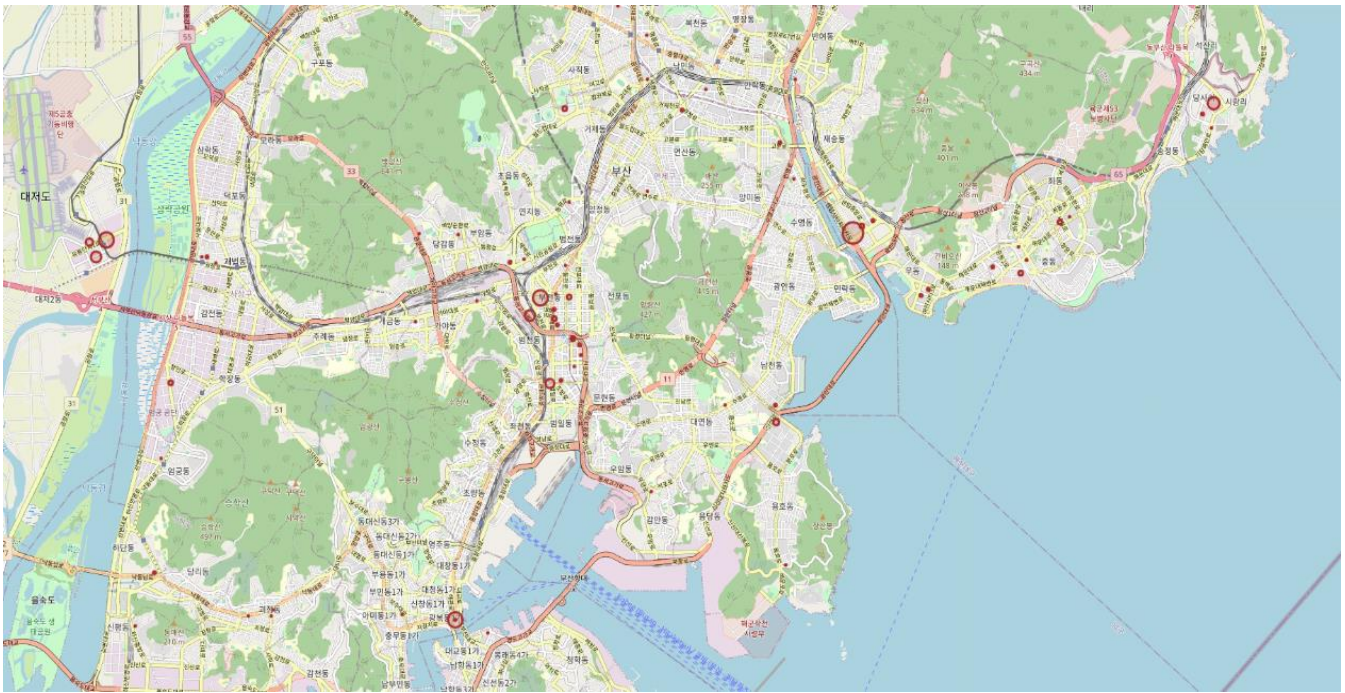
-최다 이용 버스정류장(원의 크기가 클수록 이용객이 많음)



-승/하차 승객이 가장 많은 상위 15개 지하철역(원의 크기가 클수록 승객이 많음)

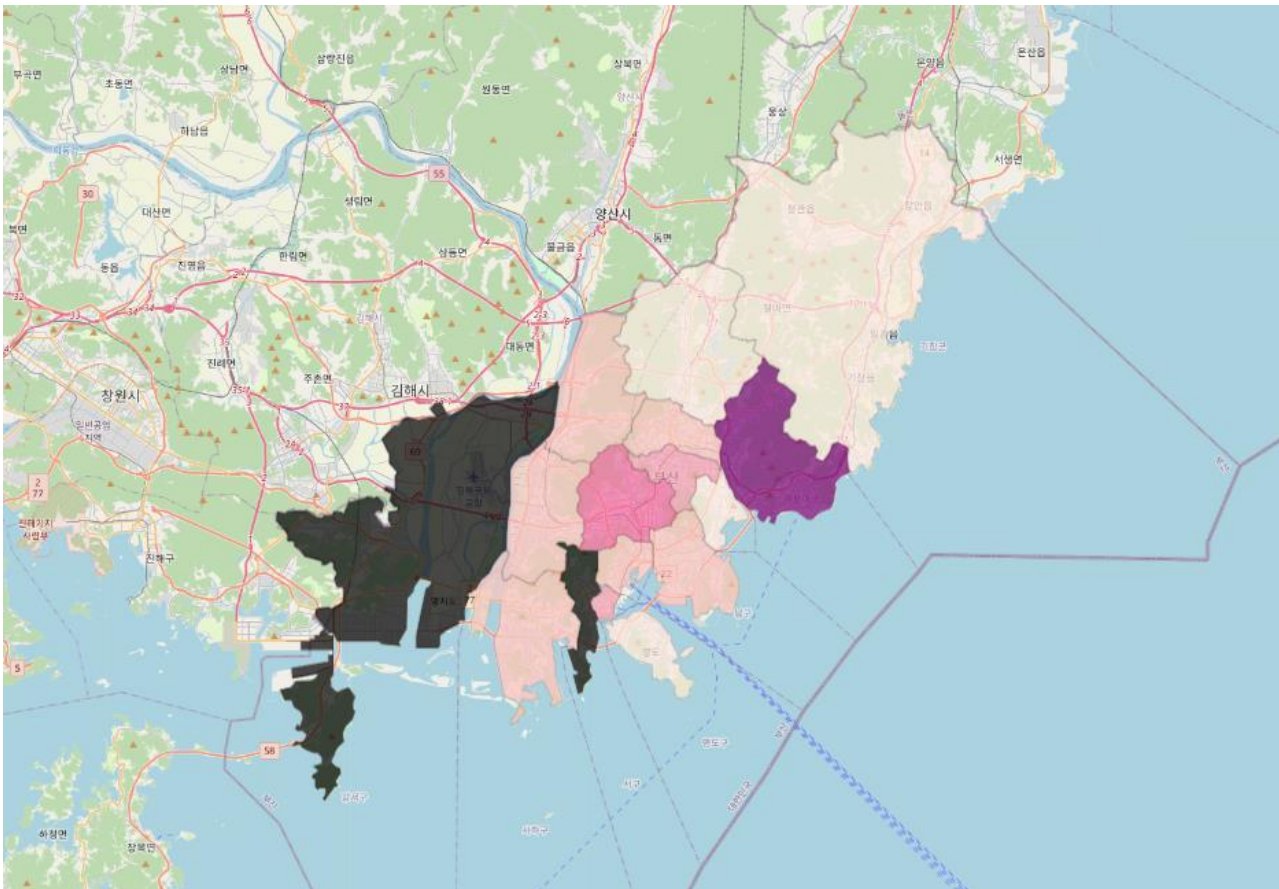


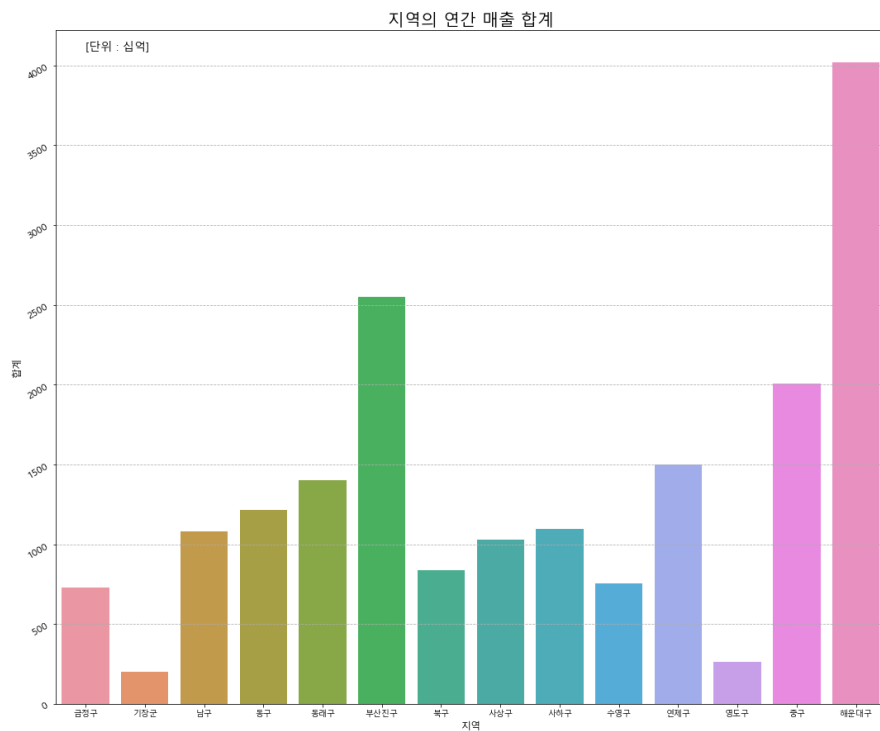
-부산시 내 위치한 공원 현황(원의 크기가 클수록 면적이 넓음)



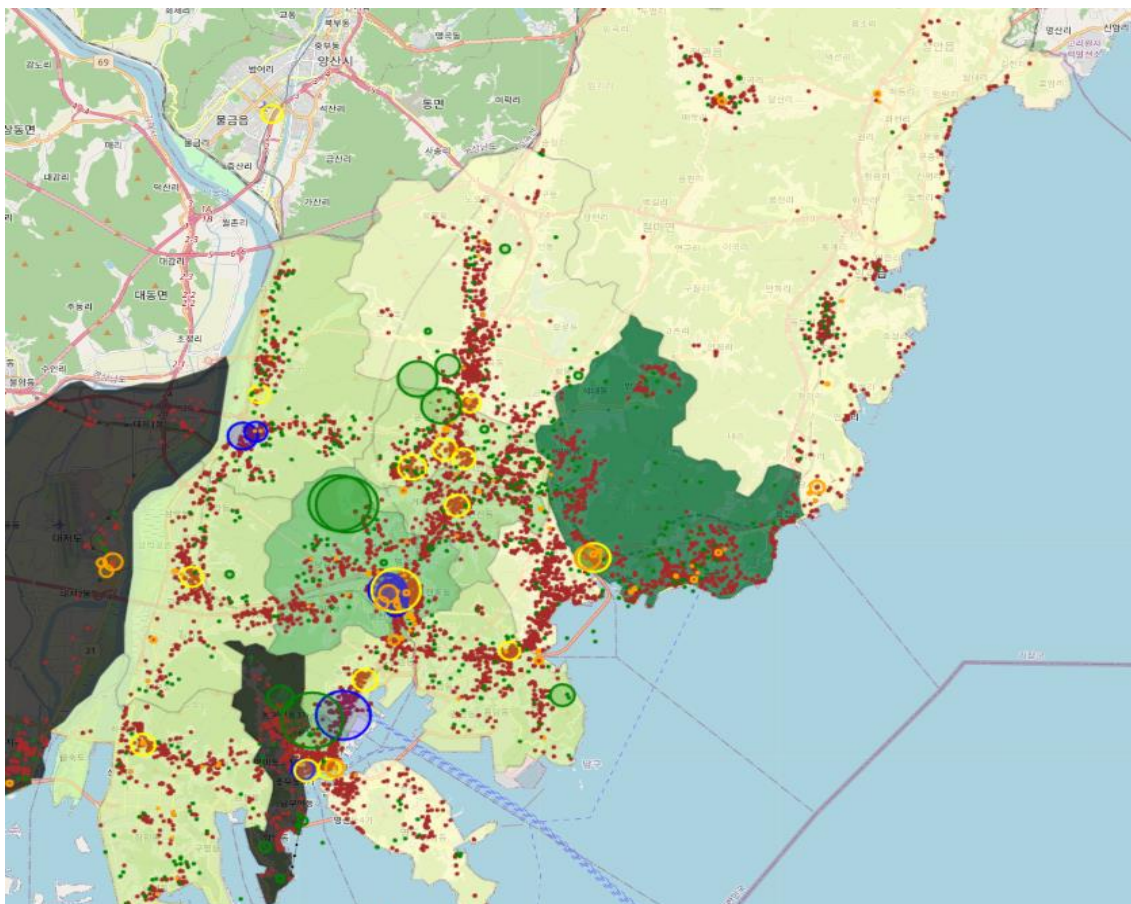
-부산시 내 위치한 대규모점포 현황(원의 크기가 클수록 규모가 큼)

지역별 연간 매출 합계





최종 후보지 결정



앞서 상권데이터와 카페데이터로 도출해낸 결과인 부산진구, 해운대구, 중구, 동래구, 금정구, 기장군 확인

버스, 지하철, 공원, 대규모상점 데이터가 겹쳐지는 구간 확인

[-부산진구 11개 / -중구 8개 / -동래구 5개 / -해운대구 2개 / -금정구 3개 / -기장군 1개]

=> **부산진구, 중구, 동래구**로 유력후보지 선정

1순위 후보군 부산진구의 다회용컵 수거함 위치 제안

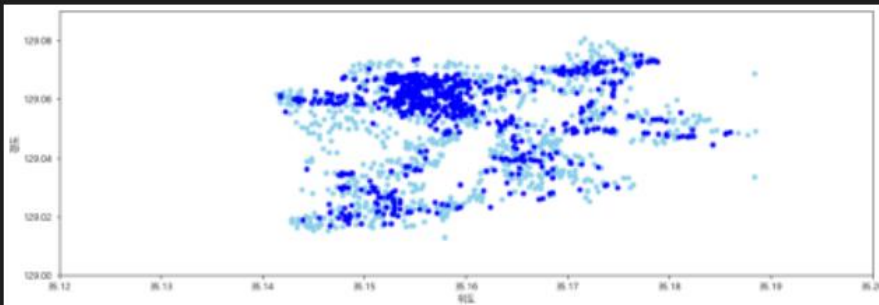
부산 진구 전체 CCTV좌표를 후보군으로 설정

부산 진구 모든 CCTV 좌표를 초기 후보군으로 설정

- 하늘색 점 : CCTV 좌표
- 파란색 점 : 카페 좌표

```
fig, ax = plt.subplots(figsize=(15,5))
plt.xlim(35.12, 35.20)
plt.ylim(129.00, 129.09)
cctv[['위도', '경도']].plot.scatter(x='위도', y='경도', ax=ax, color='skyblue')
cafe[['위도', '경도']].plot.scatter(x='위도', y='경도', ax=ax, color='blue')
print("초기 후보군: ",len(cctv))
```

초기 후보군: 1008



카페와 50m 이내에 있는 후보군인 경우는 제외

- 다회용컵 '수거'보다 '반납'이 용이하기 때문에 제외

```
# 카페 50m 이내의 후보군 좌표는 제외하고 50m 이외의 후보군만 추출
addresses = []
lats = []
lons = []

for address,lat,lon in zip(cctv['소재지도로명주소'], cctv['위도'], cctv['경도']):
    in50m = False
    candidate = (lat, lon)
    for cafe_lat,cafe_lon in zip(cafe['위도'], cafe['경도']):
        cafe_50 = (cafe_lat, cafe_lon)
        if haversine(candidate, cafe_50)*1000 <= 50: # 50m
            in50m=True
            break

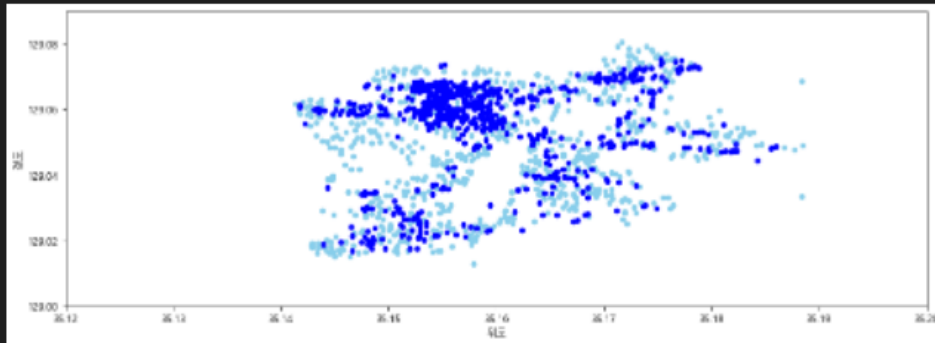
    if in50m == False:
        addresses.append(address)
        lats.append(lat)
        lons.append(lon)
```

```
# 추출된 후보군을 (module) pd r 변수에 담기
candidates_far = pd.DataFrame([])
candidates_far['도로명주소']=addresses
candidates_far['위도']=lats
candidates_far['경도']=lons
```

추출된 후보군

```
fig, ax = plt.subplots(figsize=(15, 5))
plt.xlim(35.12, 35.20)
plt.ylim(129.00, 129.09)
candidates_far.plot.scatter(x='위도',y='경도',ax=ax,color='skyblue')
cafe.plot.scatter(x='위도',y='경도',ax=ax,color='blue')
print("현재 후보군: ",len(candidates_far))
```

현재 후보군: 759



카페와의 거리가 15분 이상인 후보군 제외

- 수거함이 위치하기에는 너무 먼 거리라 판단하여 제외

```
# 카페와의 거리가 15분 이상인 후보군 좌표는 제외하고 15분 미만인 후보군만 추출
```

```
#카페와의 최소 거리 계산
```

```
distance_cafe = []
```

```
for address,lat,lon in zip(candidates_far['도로명주소'], candidates_far['위도'],candidates_far['경도']):
    minimum = 1000000
    candidate = (lat, lon)
    for cafe_lat,cafe_lon in zip(cafe['위도'],cafe['경도']):
        cafe_15 = (cafe_lat, cafe_lon)
        if haversine(candidate, cafe_15)*1000 < minimum:
            minimum = haversine(candidate, cafe_15)*1000
    distance_cafe.append(minimum)
```

```
# 연령별 보행속도의 평균 구하기
```

```
walk = np.average([1.3,1.4,1.49,1.35,1.41,1.41,1.61,1.32])
```

```
walk
```

```
1.41125
```

```
# 15분보다 멀리 있는 후보군은 제외.
```

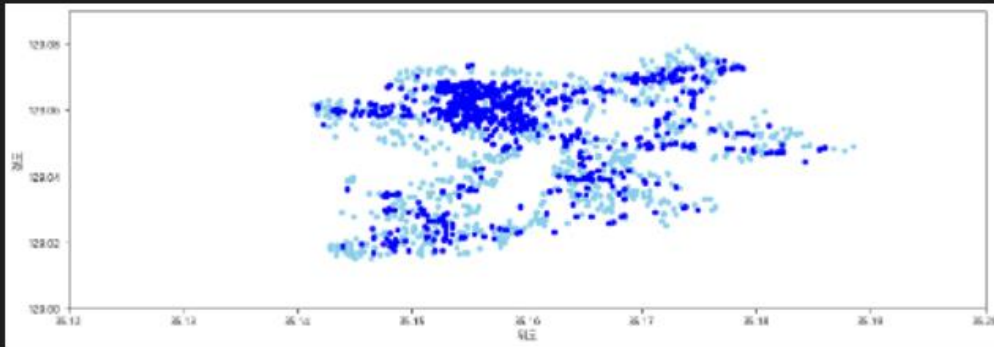
```
candidates_far['distance_cafe'] = distance_cafe
```

```
candidates_far=candidates_far[candidates_far['distance_cafe']<=(5*60*walk)]
```


추출된 후보군

```
fig, ax = plt.subplots(figsize=(15,5))
plt.xlim(35.12, 35.20)
plt.ylim(129.00, 129.09)
candidates_far.plot.scatter(x='위도',y='경도',ax=ax,color='skyblue')
cafe.plot.scatter(x='위도',y='경도',ax=ax,color='blue')
print("현재 후보군: ",len(candidates_far))
```

현재 후보군: 728



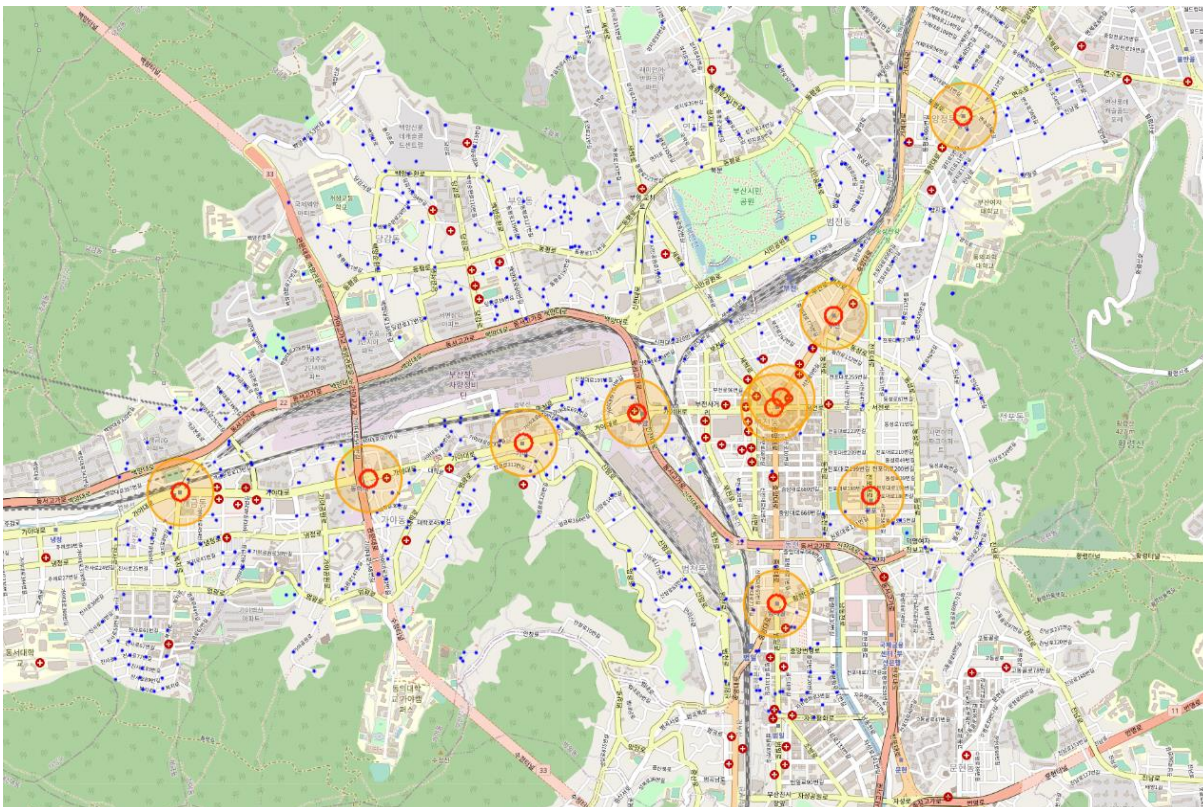
```
len(candidates_far)
```

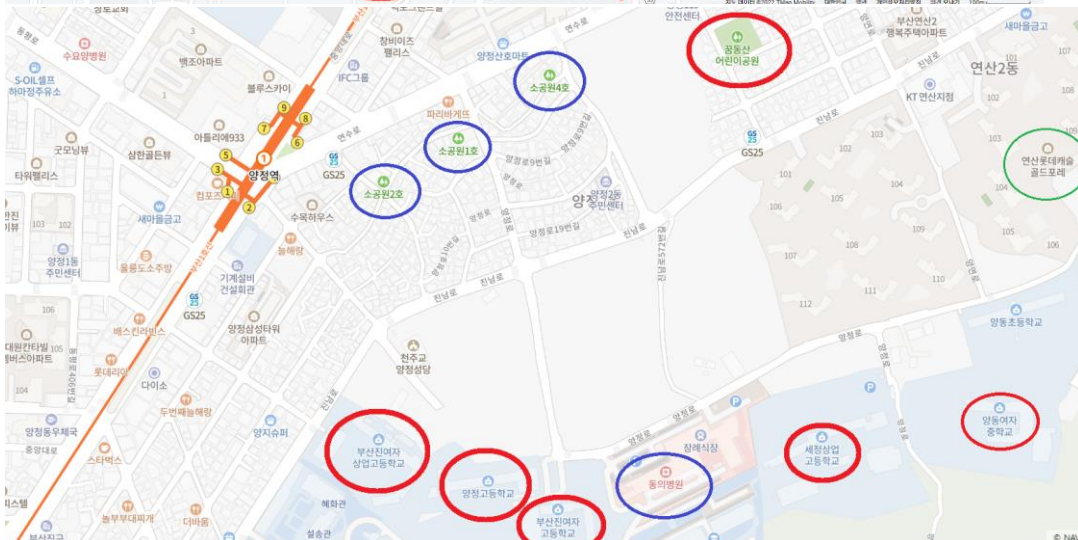
728

-최종 추출 후보군 1008 --> 728

추출된 후보군과 부산 진구 지하철 위치를 비교하여 최종 후보군 설정

-부산 진구 지하철 역 반경 50m 이내에 포함되는 후보군을 최종 후보군으로 설정





최종 수거함 설치 장소

동의대역

양정역

부전역