

2021

파이썬, R을 활용한 빅데이터 시각화 구현

PEDIATRICS⁺

(소아청소년과)

INDEX



개요



데이터 수집



데이터 전처리



데이터 분석



결론

개요

주제 및 목적

일정관리


개발환경

가설설정

주제 & 목적

SUBJECT & PURPOSE

Subject :

 소중한
아이들을 위한
소아과

Purpose :

가까이 더 가까이



		 계획	 완료	8/23	8/24	8/25	8/26	8/27
기획 및 설계	선정 및 요구사항, 일정관리							
	설계, 기능정의, 데이터수집							
서론 및 이론적 배경	연구 배경 및 목적							
	연구 내용 및 방법							
	프로젝트의 구성							
	이론적 배경							
연구 모형 및 조사 설계	연구모형 및 연구가설							
	변수의 조작적 정의 및 측정 항목							
실증 분석 및 결론	자료 수집 및 표본의 특성							
	측정항목의 기술통계분석 및 시각화							
	신뢰도 및 타당성 검증 및 시각화							
	각자에 맞는 데이터분석 및 시각화							
	추가 분석 및 시각화							
테스트 및 수정								

OS

Windows 10 Pro

Language

Python 3.8.8

IDE

Jupyter Notebook 6.3.0

Open Source

BeautifulSoup, geopy, folium



첫 번째.

아동 많은 지역은
곧 소아과 많은 지역



두 번째.

인구 감소로 인한
소아과 부족현상 없음



데이터 수집

자료 출처
자료 수집



소아과 목록

대한소아청소년과학회 ([웹 크롤링](#))



인구 현황

행정안전부 인구통계 ([다운로드](#))



지번 주소

행정안전부 개발자센터 ([다운로드](#))



행정구역 : 서울시와 경기도의 ‘**읍면동**’



아동수 : 2021년 7월 등록된 **만0~9세**

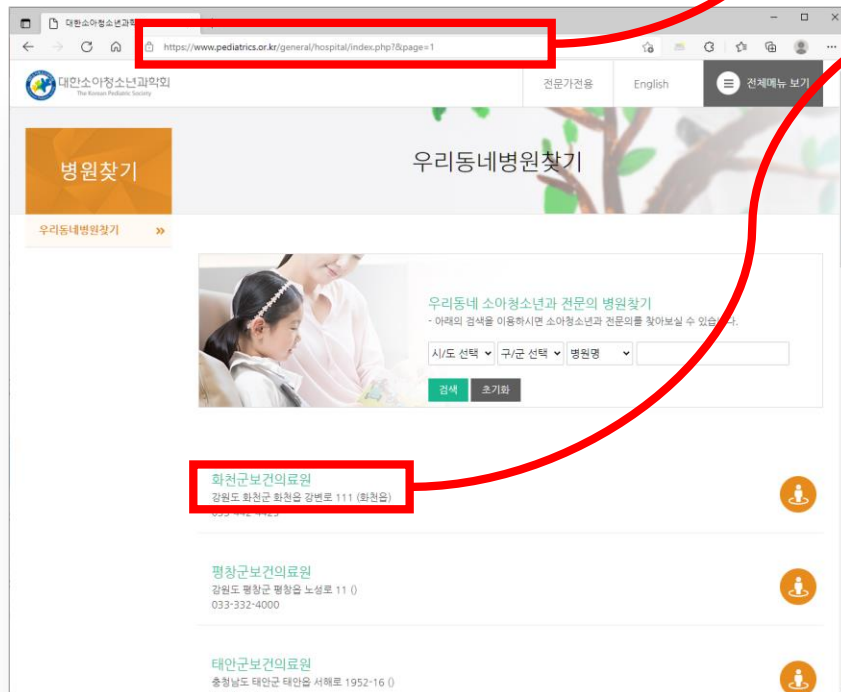


소아청소년과는 약칭으로 ‘소아과’로 칭함



소아과학회 SITE에 공개된 병원명단을 기준으로 함

Pediatrics



```
import requests
from bs4 import BeautifulSoup
pediatrics = []
error = []
for u in range(1, 995) : # 1~994 데이터 있는 페이지
    try :
        url = 'https://www.pediatrics.or.kr/general/hospital/index.php?page='+str(u)
        p = requests.get(url)
        soup = BeautifulSoup(p.content, "html.parser")
        p_name = soup.select("dl.hospitalInfo > dt > span:nth-of-type(1)")
        p_address = soup.select("dl.hospitalInfo > dt > span:nth-of-type(2)")
        for idx in range(len(p_name)):
            pediatrics.append([p_name[idx].text, p_address[idx].text])
    except Exception as e :
        error.append(u)
    pass
```

	병원명	주소
0	화천군보건의료원	강원도 화천군 화천읍 강변로 111 (화천읍)
1	평창군보건의료원	강원도 평창군 평창읍 노성로 11 ()
2	태안군보건의료원	충청남도 태안군 태안읍 서해로 1952-16 ()

Population

주인등록 인구통계 행정안전부

주인등록 인구통계

주인등록 인구 및 세대현황

통계표

행정구역: 전국 시군구

등록구분: 전체

조회기간: 2021년 07월 ~ 2021년 07월

구분: ☒ 남여 구분 ☒ 남여 구성비 ☒ 세대당인구

정렬순서: 행정기관코드 오름차순

현재화면 ☐ 전체시군구현황 ☐ 전체읍면동현황 ☒ 전체읍면동현황

csv 파일 다운로드

주인등록 인구 및 세대현황

행정기관	2021년 07월					
	총 인구수	세대수	세대당 인구	남자 인구수	여자 인구수	남여 비율
전국	51,671,569	23,294,707	2.22	25,768,310	25,903,259	0.99
서울특별시	9,558,153	4,409,964	2.17	4,645,177	4,912,976	0.95

```
1 under10_df.info()

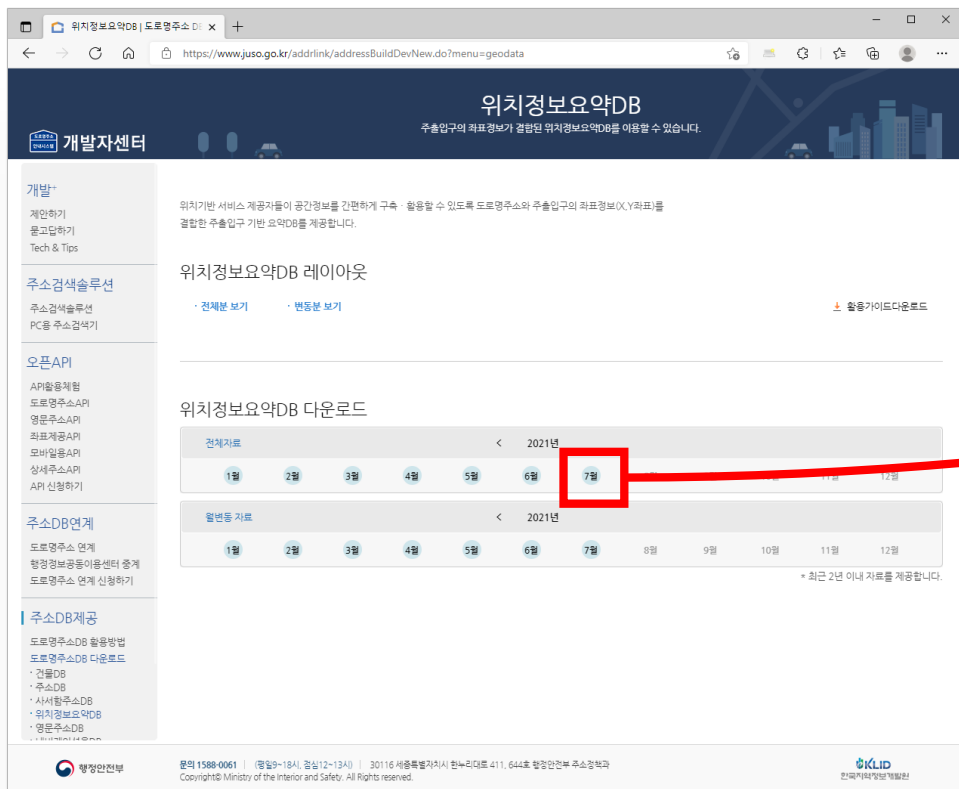
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3848 entries, 0 to 3847
Data columns (total 4 columns):
#   Column                                Non-Null Count  Dtype
---  ---                                ---
0   행정 구역                             3848 non-null   object
1   2021년07월_계_총인구수                 3848 non-null   object
2   2021년07월_계_연령구간인구수           3848 non-null   object
3   2021년07월_계_0~9세                     3848 non-null   object
dtypes: object(4)
memory usage: 120.4+ KB

1 under10_df.isnull().sum()

행정 구역                0
2021년07월_계_총인구수    0
2021년07월_계_연령구간인구수  0
2021년07월_계_0~9세      0
dtype: int64

1 under10_df.head(1)

      행정 구역  2021년07월_계_총인구수  2021년07월_계_연령구간인구수  2021년07월_계_0~9세
0  서울특별시 (11000000000)           9,558,153                599,600                599,600
```



```
1 import pandas as pd
2 seoul_addr_df = pd.read_csv('data/entrc_seoul.txt', sep='|', encoding='cp949', header=None) # 서울
3 gyunggi_addr_df = pd.read_csv('data/entrc_gyunggi.txt', sep='|', encoding='cp949', header=None) # 경기
```

```
1 seoul_addr_df.isnull().sum()
```

```
0      0
1      0
2      0
3      0
4      0
5      0
6      0
7      0
8      0
9      0
10     0
11    856114
12      0
13    1179
14      0
15     745
16    2047
17    2047
dtype: int64
```

```
1 seoul_addr_df.head(1)
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
	0	11110	760	1111010100	서울특별시	영문	정읍시	111103100012	자하문	0	94	0	NaN	3047	근린생활시설	0	정읍호차	953241.683263	1.954023e+06

데이터 전처리

컨텐츠
전처리

Python Contents

Contents 🔄 ⚙️

- ▼ 1 데이터 준비
 - ▼ 1.1 웹 크롤링 : 대한소아청소년과학회 / 전국의 소아청소년과 병원 목록
 - 1.1.1 pediatrics_df 파일 저장 : 소아과 목록 (도로명 주소 포함)
 - 1.2 다운로드 : 행정안전부 / 전국의 행정구역별 만0~9세 인구현황
 - 1.3 다운로드 : 행정안전부 개발자센터 / 위치정보DB(주소별 읍면동)
- ▼ 2 데이터 전처리 (각 데이터별)
 - 2.1 소아과
 - ▼ 2.2 아동
 - 2.2.1 under10_df : 간단한 전처리 후 저장
 - 2.3 읍면동
- ▼ 3 소아과 + 읍면동 + 아동수 + merge // 위도, 경도 변환값 받기
 - 3.1 소아과df와 읍면동df merge
 - ▼ 3.2 위도와 경도 정보 받기
 - 3.2.1 ped_addr_df 저장 : 소아과목록 + 읍면동별 위도, 경도
 - ▼ 3.3 행정구역별 소아과수, 아동수, 위도, 경도
 - 3.3.1 df 파일저장 : 행정구역, 아동수, 읍면동, 소아과
 - ▼ 3.4 남은 소수의 결측치 확인 및 처리
 - 3.4.1 df_1 저장 : 결측치 없음 // 행정구역 + 읍면동 + 아동수 + 소아과 + 위도 + 경도 + 비율
- ▼ 4 분석 및 시각화
 - 4.1 분포도 (소아과 & 아동) // 단, 경기도는 중심부분만
 - ▼ 4.2 상위10곳 수치그래프 (소아과 & 아동)
 - 4.2.1 df_2 저장 : 동일생활권 추출 읍면동 삭제
 - 4.3 아동대비 소아과 비율 분포도
 - 4.4 소아과가 없는 읍면동
 - 4.5 소아과 한곳당 아동 비율 높은 상위 60개 읍면동
 - 4.6 비율 상위 10곳 + 비율 상위 10번째보다 아동수가 많지만 소아과 없는 곳 (지도)
 - 4.7 비율 상위 10곳 (그래프)
 - 4.8 비율 상위 10번째보다 아동수가 많지만 소아과 없는 곳 (그래프)

Data Preprocessing

```
# 서울
seoul_addr_df = pd.read_csv('data/entrc_seoul.txt', sep='|', encoding='cp949', header=None)
seoul_addr_df = seoul_addr_df[[3,4,5,7,9]]
seoul_addr_df.columns = ['시도', '시군구', '읍면동', '도로명', '건물번호']
seoul_addr_df['도로명주소'] = seoul_addr_df['도로명'] + ' ' + seoul_addr_df['건물번호'].astype(str)

# 경기
gyunggi_addr_df = pd.read_csv('data/entrc_gyunggi.txt', sep='|', encoding='cp949', header=None)
gyunggi_addr_df = gyunggi_addr_df[[3,4,5,7,9]]
gyunggi_addr_df.columns = ['시도', '시군구', '읍면동', '도로명', '건물번호']
gyunggi_addr_df['도로명주소'] = gyunggi_addr_df['도로명'] + ' ' + gyunggi_addr_df['건물번호'].astype(str)

under10_df = under10_df[['행정구역', '2021년07월_계_0-9세']]
under10_df = under10_df[under10_df['행정구역'].str.startswith(('서울', '경기'))]
under10_df['행정구역'] = under10_df['행정구역'].str.replace(r'##(.*)', '')
under10_df['행정구역'] = under10_df['행정구역'].str.strip()
under10_df = under10_df[under10_df['행정구역'].str.count('.') >= 2]
under10_df.columns = ['행정구역', '아동수']
under10_df['아동수'] = under10_df['아동수'].str.replace('.', '')
under10_df['행정구역'] = under10_df['행정구역'].str.replace('.', '').str.replace(r'##제##동', '동')
under10_df['행정구역'] = under10_df['행정구역'].str.replace(r'##d+##가', '').str.replace(r'##d+##동', '동')

under10_df.index = range(under10_df.shape[0])
for idx in range(len(under10_df)):
    if (under10_df['행정구역'][idx].startswith('경기도')) & (under10_df['행정구역'][idx].endswith('구')):
        under10_df.drop(idx, inplace=True)
under10_df.index = range(under10_df.shape[0])

under10_df['아동수'] = under10_df['아동수'].apply(int)
under10_df = under10_df.groupby(['행정구역'])['아동수'].apply(sum).reset_index()

under10_df['읍면동'] = None
for idx in range(len(under10_df)):
    temp_index = under10_df['행정구역'][idx].rfind('.')+1
    temp = under10_df['행정구역'][idx][temp_index:]
    under10_df['읍면동'][idx] = temp
```

```
ped_addr_df = pd.read_csv('data/ped_addr_df.csv', encoding='utf-8')
ped_addr_df['행정구역'] = ped_addr_df[['시도', '시군구', '읍면동']].apply(lambda x: ' '.join(x), axis=1) # 행정-
ped_addr_df = ped_addr_df[['행정구역', '읍면동', '위도', '경도']] # 필요한 컬럼만
ped_addr_df['행정구역'] = ped_addr_df['행정구역'].str.replace('.', '').str.replace(r'##제##동', '동')
ped_addr_df['행정구역'] = ped_addr_df['행정구역'].str.replace(r'##동##가', '동').str.replace(r'##d+##동', '동')
ped_addr_df['행정구역'] = ped_addr_df['행정구역'].str.replace(r'##d+##가', '동')
ped_addr_df['읍면동'] = ped_addr_df['읍면동'].str.replace('.', '').str.replace(r'##제##동', '동')
ped_addr_df['읍면동'] = ped_addr_df['읍면동'].str.replace(r'##동##가', '동').str.replace(r'##d+##동', '동')
ped_addr_df['읍면동'] = ped_addr_df['읍면동'].str.replace(r'##d+##가', '동')
ped_addr_temp_df = ped_addr_df.copy()
ped_addr_df = ped_addr_df.value_counts(['행정구역', '읍면동']).to_frame('소아과').reset_index() # 행정구역별 소아과
ped_addr_df.groupby(['행정구역', '읍면동'])['소아과'].apply(sum).reset_index()
ped_addr_df, ped_addr_temp_df, on=['행정구역', '읍면동'], how='left')
ped_addr_df['소아과'] = ped_addr_df['소아과'].fillna(0)
ped_addr_df['소아과'] = ped_addr_df['소아과'].astype(int)
ped_addr_df['소아과'] = ped_addr_df['소아과'].str.replace('.', '').str.replace(r'##제##동', '동')
ped_addr_df['소아과'] = ped_addr_df['소아과'].str.replace(r'##동##가', '동').str.replace(r'##d+##동', '동')
ped_addr_df['소아과'] = ped_addr_df['소아과'].str.replace(r'##d+##가', '동')
```

```
for idx in range(0, len(df)): # 위도, 경도 1차
    try:
        if any(df.loc[idx, ['위도', '경도']].isnull()):
            print(idx, end='\n\t')
            df['위도'][idx], df['경도'][idx] = app.geocode(df['행정구역'][idx])[1]
    except Exception:
        pass

for idx in range(0, len(df)): # 위도, 경도 2차 ('경기도' 삭제)
    try:
        if any(df.loc[idx, ['위도', '경도']].isnull()):
            print(idx, end='\n\t')
            temp_addr = df['행정구역'][idx].replace('경기도', '')
            df['위도'][idx], df['경도'][idx] = app.geocode(temp_addr)[1]
    except Exception:
        pass

for idx in range(0, len(df)): # 위도, 경도 3차 : 시_00구 -> 시
    try:
        if any(df.loc[idx, ['위도', '경도']].isnull()):
            print(idx, end='\n\t')
            temp_addr = re.sub(r'##시##. {2,3}##구', '시', df['행정구역'][idx])
            df['위도'][idx], df['경도'][idx] = app.geocode(temp_addr)[1]
    except Exception:
        pass

for idx in range(0, len(df)): # 위도, 경도 4차 : 경기도 00시 -> 삭제
    try:
        if any(df.loc[idx, ['위도', '경도']].isnull()):
            print(idx, end='\n\t')
            temp_addr = re.sub(r'##경기도##. {2}##시##', '', df['행정구역'][idx])
            df['위도'][idx], df['경도'][idx] = app.geocode(temp_addr)[1]
    except Exception:
        pass
```


🧪 문제점 : 비율이 과도하게 높거나 낮은 **이상치** 발견

🧪 원 인 : 1. 사실상 하나의 생활권인 '읍면동'
2. 아동수가 극소수인 '읍면동'

🧪 해 결 : 이상치 데이터 **제거**

```
temp_df = df_1.copy()

# 아동 500명이상이면서, 소아과가 0인 곳은 인근 '읍면동'과 동일생활권인 것으로 간주 = 137개
temp_df = temp_df.drop(temp_df[(temp_df['소아과']==0) & (temp_df['아동수'] >= 500)].index)

# 소아과수 대비 아동수가 400이상인 곳은 인근 '읍면동'과 동일생활권인 것으로 간주 = 100개
temp_df = temp_df.drop(temp_df[(temp_df['비율']>=400) & (temp_df['비율']!=np.inf)].index)

# 아동이 50명 이내이면서 소아과가 1곳 이상인 곳은 인근 '읍면동'과 동일생활권인 것으로 간주 = 239개
temp_df = temp_df.drop(temp_df[(temp_df['소아과']>0) & (temp_df['아동수']<=50)].index)

df_2 = temp_df.copy()    # df_1 -> df_2  // 893개 -> 417개
df_2.index = range(df_2.shape[0])
```



Preprocessed Date

```
1 df.isnull().sum()
행정구역    0
읍면동      0
아동수      0
소아과      0
위도        0
경도        0
비율        0
소아과_단위    0
dtype: int64
```

```
1 df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 893 entries, 0 to 892
Data columns (total 8 columns):
#   Column      Non-Null Count  Dtype
---  -
0   행정구역    893 non-null    object
1   읍면동      893 non-null    object
2   아동수      893 non-null    int64
3   소아과      893 non-null    int64
4   위도        893 non-null    float64
5   경도        893 non-null    float64
6   비율        893 non-null    float64
7   소아과_단위 893 non-null    object
dtypes: float64(3), int64(2), object(3)
memory usage: 55.9+ KB
```

```
1 df.sort_values('소아과', ascending=False)[:300]
```

	행정구역	읍면동	아동수	소아과	위도	경도	비율	소아과_단위
625	서울특별시 강서구 화곡동	화곡동	10496	71	37.542134	126.844213	147.83	50~
282	경기도 안산시 상록구 월피동	월피동	2144	8	37.333936	126.848205	268.00	~10
757	서울특별시 성북구 동선동	동선동	549	1	37.593646	127.018207	549.00	~10

데이터 분석

| 결과 도출

종류 : 분포도

대상 : 소아과와 아동수

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
df_1 = pd.read_csv('data/df_1.csv', encoding='utf-8')

gyunggi_df = df_1[df_1['행정구역'].str.startswith('경기')]
seoul_df = df_1[df_1['행정구역'].str.startswith('서울')]

gyunggi_center = gyunggi_df[(37 < gyunggi_df['위도']) & (gyunggi_df['위도'] < 38)
                             & (126.75 < gyunggi_df['경도']) & (gyunggi_df['경도'] < 127.2)]

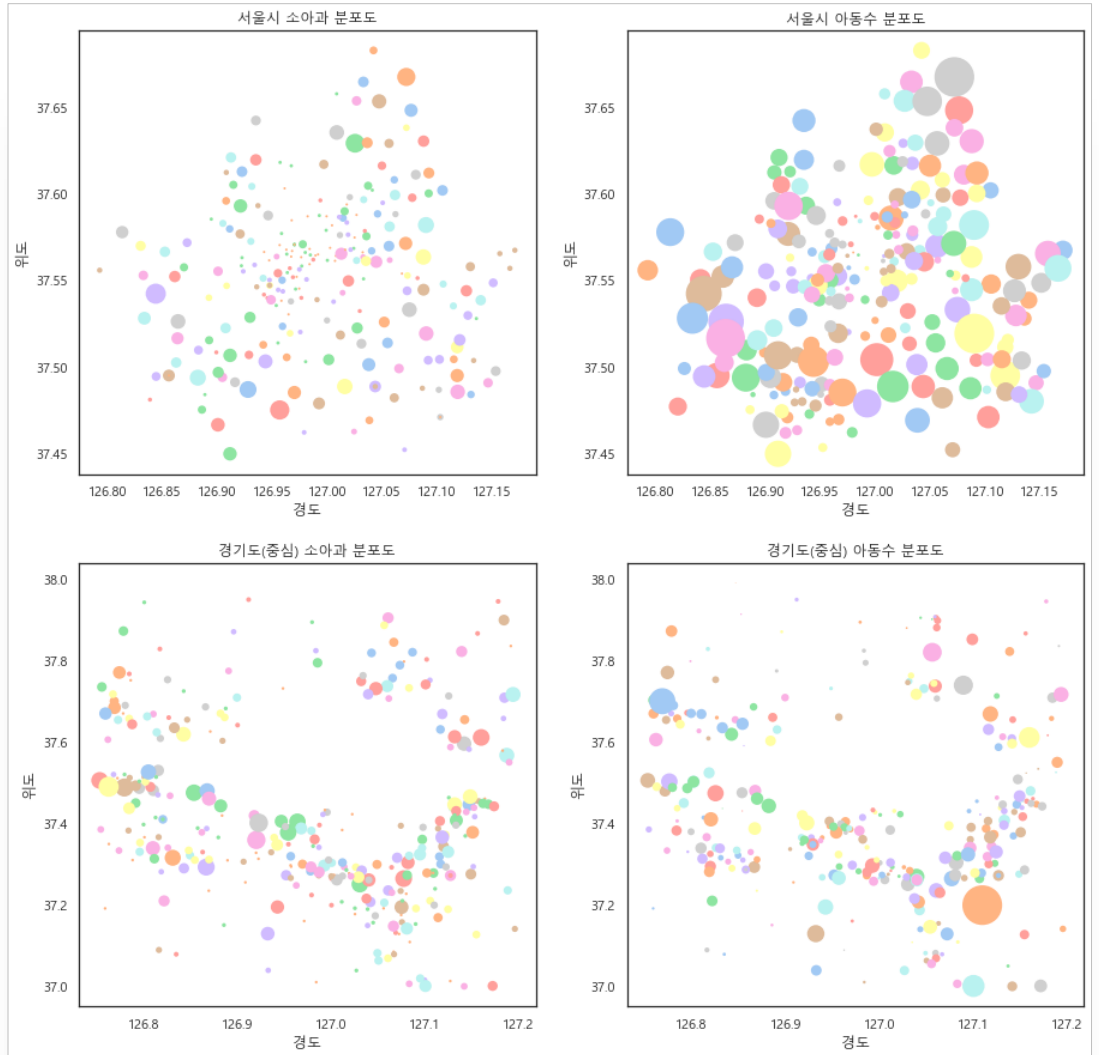
sns.set(style='white')
plt.rc('font', family='Malgun Gothic')
fig, axes = plt.subplots(ncols=2, nrows=2, figsize=(15,15))
plt.subplots_adjust(wspace=0.2, hspace=0.2)
maxsize_p = max(df_1['소아과']) * 4
maxsize_k = max(df_1['아동수']) * 0.02

g1 = sns.scatterplot(data=seoul_df, x='경도', y='위도', size='소아과', hue='소아과', sizes=(0, maxsize_p),
                    palette='pastel', ax=axes[0,0])
g2 = sns.scatterplot(data=seoul_df, x='경도', y='위도', size='아동수', hue='아동수', sizes=(0, maxsize_k),
                    palette='pastel', ax=axes[0,1])
g3 = sns.scatterplot(data=gyunggi_center, x='경도', y='위도', size='소아과', hue='소아과', sizes=(0, maxsize_p),
                    palette='pastel', ax=axes[1,0])
g4 = sns.scatterplot(data=gyunggi_center, x='경도', y='위도', size='아동수', hue='아동수', sizes=(0, maxsize_k),
                    palette='pastel', ax=axes[1,1])

g1.set_title('서울시 소아과 분포도')
g2.set_title('서울시 아동수 분포도')
g3.set_title('경기도(중심) 소아과 분포도')
g4.set_title('경기도(중심) 아동수 분포도')

g1.get_legend().remove()
g2.get_legend().remove()
g3.get_legend().remove()
g4.get_legend().remove()

plt.savefig('data/4_1_분포도.png', bbox_inches='tight')
plt.show()
```



종류 : 막대그래프

대상 : 소아과와 아동수 (상위 10곳)

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
df_1 = pd.read_csv('data/df_1.csv', encoding='utf-8')

gyunggi_df = df_1[df_1['행정구역'].str.startswith('경기')]
seoul_df = df_1[df_1['행정구역'].str.startswith('서울')]

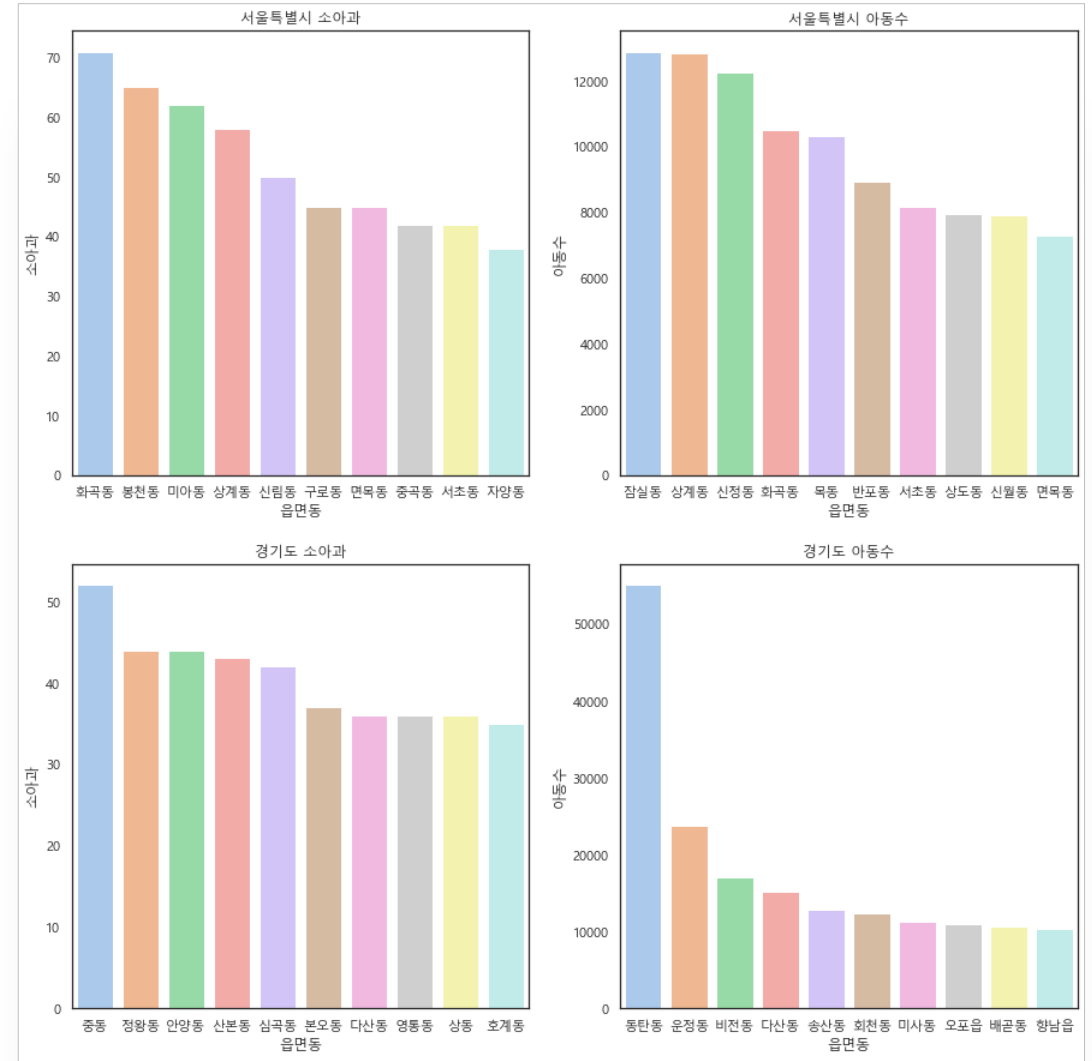
d1 = seoul_df.sort_values('소아과', ascending=False).head(10)
d2 = seoul_df.sort_values('아동수', ascending=False).head(10)
d3 = gyunggi_df.sort_values('소아과', ascending=False).head(10)
d4 = gyunggi_df.sort_values('아동수', ascending=False).head(10)

fig, axes = plt.subplots(ncols=2, nrows=2, figsize=(15,15))
plt.subplots_adjust(wspace=0.2, hspace=0.2)

g1 = sns.barplot(data=d1, x='읍면동', y='소아과', palette='pastel', ax=axes[0,0])
g2 = sns.barplot(data=d2, x='읍면동', y='아동수', palette='pastel', ax=axes[0,1])
g3 = sns.barplot(data=d3, x='읍면동', y='소아과', palette='pastel', ax=axes[1,0])
g4 = sns.barplot(data=d4, x='읍면동', y='아동수', palette='pastel', ax=axes[1,1])

g1.set(title = '서울특별시 소아과')
g2.set(title = '서울특별시 아동수')
g3.set(title = '경기도 소아과')
g4.set(title = '경기도 아동수')

plt.savefig('data/4_2_그래프.png', bbox_inches='tight')
plt.show()
```



종류 : 분포도

대상 : 아동대비 소아과 비율

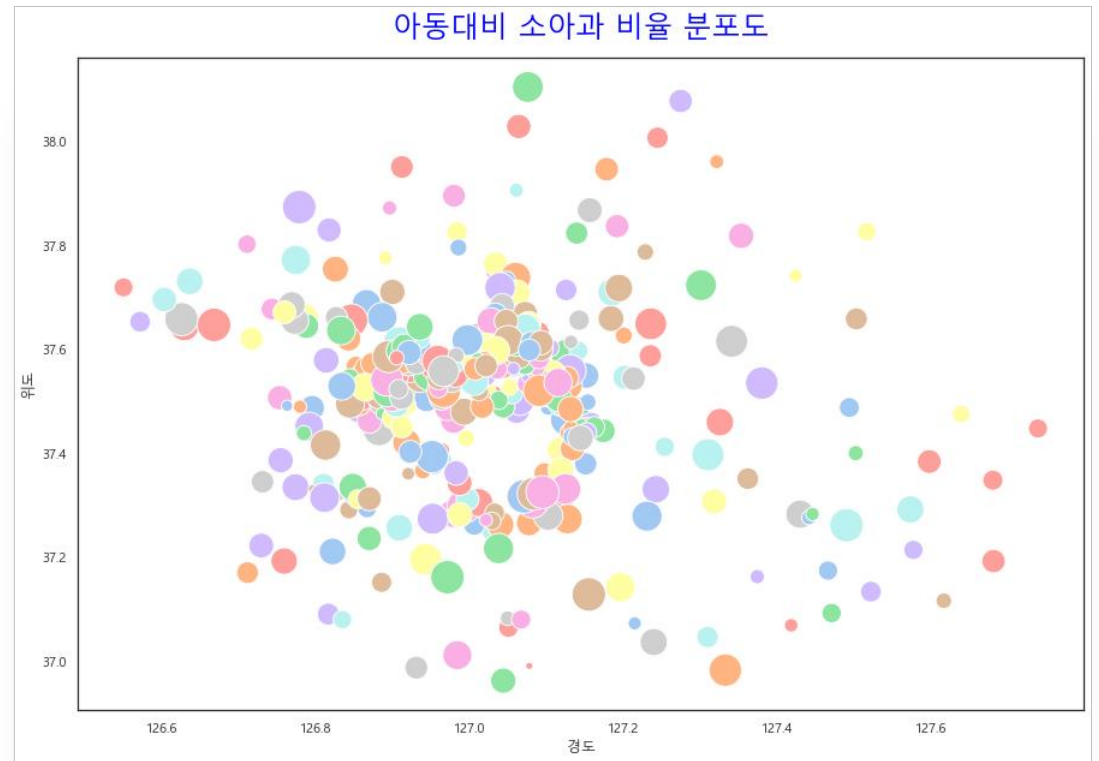
```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
df_2 = pd.read_csv('data/df_2.csv', encoding='utf-8')

d1 = df_2[df_2['비율']!=np.inf].copy()

plt.figure(figsize = (15,10))
sns.set(style='white')
plt.rc("font", family="Malgun Gothic")

maxsize = max(d1['비율']) * 2
g1 = sns.scatterplot(data=d1, x='경도', y='위도', size='비율', hue='비율', sizes=(0, maxsize), palette='pastel')
g1.set_title('아동대비 소아과 비율 분포도', fontdict={'fontsize': 24, 'color':'blue'}, pad=17)
g1.get_legend().remove()

plt.savefig('data/4_3_분포도.png', bbox_inches='tight')
plt.show()
```



종류 : 막대그래프

대상 : 소아과 없는 곳의 아동수

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
df_2 = pd.read_csv('data/df_2.csv', encoding='utf-8')

d1 = df_2[df_2['비율']!=np.inf].sort_values('아동수', ascending=False).copy()
d1['읍면동_아동'] = d1[['읍면동', '아동수']].astype(str).apply(lambda x : ' '.join(x), axis = 1)
d1.index = range(d1.shape[0])

plt.figure(figsize = (15,20))
g1 = sns.barplot(data=d1, y='행정구역', x='아동수', palette='pastel', ci=None)
g1.set_title('소아과 없는 읍면동', fontdict={'fontsize': 24, 'color': 'blue'}, pad=17)

for idx in range(len(d1)) :
    plt.text(d1['아동수'].iloc[idx]+1, idx+0.3, d1['읍면동_아동'].iloc[idx])

plt.savefig('data/4_4_그래프.png', bbox_inches='tight')

plt.show()
```



종류 : 막대그래프

대상 : 아동비율 상위60곳

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
df_2 = pd.read_csv('data/df_2.csv', encoding='utf-8')

d1 = df_2[df_2['비율']!=np.inf].sort_values('비율', ascending=False).head(60).copy()
d1['읍면동_비율'] = d1[['읍면동', '비율']].astype(str).apply(lambda x : ' '.join(x), axis = 1)
d1.index = range(d1.shape[0])

plt.figure(figsize = (15,20))
g1 = sns.barplot(data=d1, y='행정구역', x='비율', palette='pastel', ci=None)
g1.set_title('소아과 한곳당 아동 비율(상위 60개 읍면동)', fontdict={'fontsize': 24, 'color':'blue'}, pad=17)

for idx in range(len(d1)) :
    plt.text(d1['비율'].iloc[idx]+1, idx+0.3, d1['읍면동_비율'].iloc[idx])
g1.set_xlim(left=275, right=425)

plt.savefig('data/4_5_그래프.png', bbox_inches='tight')

plt.show()
```



종류 : 막대그래프

대상 : 소아과 부족지역 선정 (비율 상위 10곳)

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
df_2 = pd.read_csv('data/df_2.csv', encoding='utf-8')

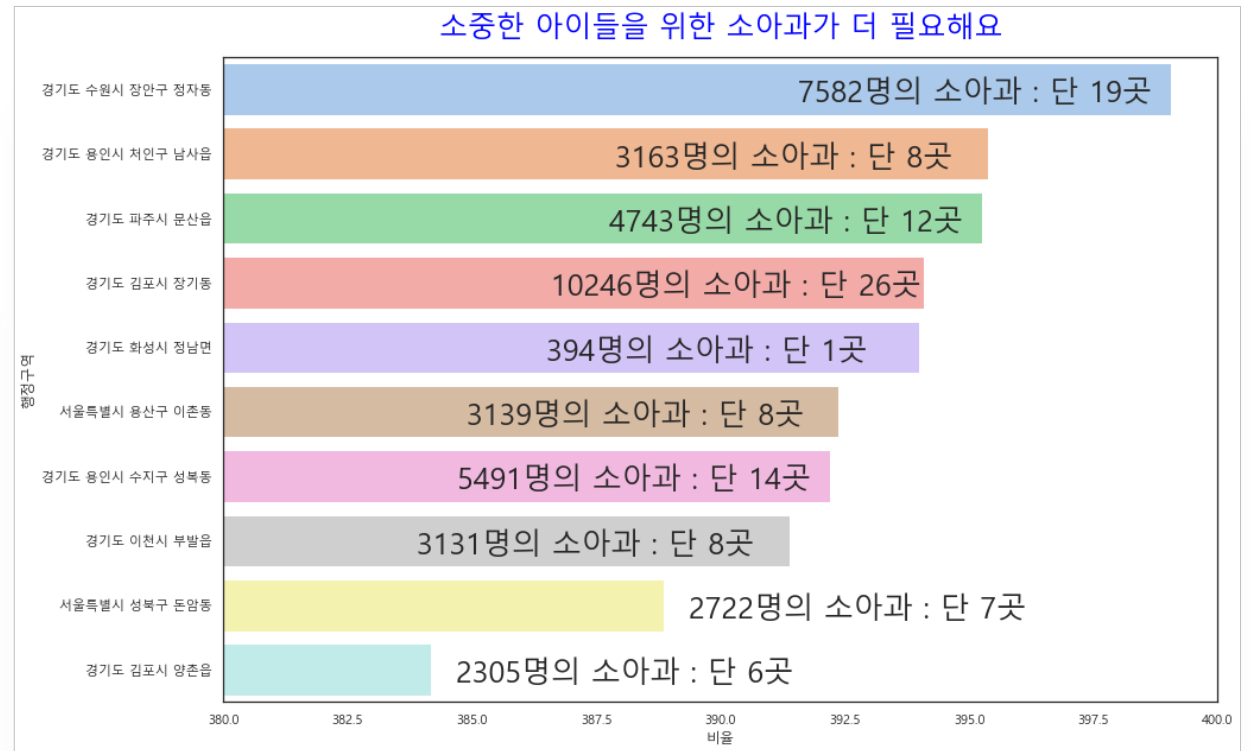
# d1 = 비율 높은 상위 10곳   d2 = 소아과가 없는 곳중 d1의 비율보다 아동이 많은 곳
d1 = df_2.copy()
d1 = d1[d1['비율']!=np.inf].sort_values('비율', ascending=False).head(10)
d1.index = range(d1.shape[0])

d2 = df_2[df_2['비율']==np.inf].sort_values('아동수', ascending=False).copy()
d2 = d2[(d2['아동수']) >= (d1['비율'].iloc[-1])]
d2.index = range(d2.shape[0])

plt.figure(figsize = (15,10))
g1 = sns.barplot(data=d1, x='비율', y='행정구역', palette='pastel')
g1.set_xlim(left=380, right=400)
g1.set_title('소중한 아이들을 위한 소아과가 더 필요해요', fontdict={'fontsize': 24, 'color':'blue'}, pad=17)

for idx in range(len(d1)) :
    text = d1['아동수'].iloc[idx].astype(str) + '명의 소아과 : 단 ' + d1['소아과'].iloc[idx].astype(str) + '곳'
    if d1['비율'].iloc[idx] > 390 :
        plt.text(d1['비율'].iloc[idx]-7.5, idx*0.2, text, size=25)
    else :
        plt.text(d1['비율'].iloc[idx]+0.5, idx*0.2, text, size=25)

plt.savefig('data/4_7_그래프.png', bbox_inches='tight')
plt.show()
```



종류 : 막대그래프

대상 : 소아과 필요지역 선정 (비율 상위 10곳보다 아동수 많은 곳)

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
df_2 = pd.read_csv('data/df_2.csv', encoding='utf-8')

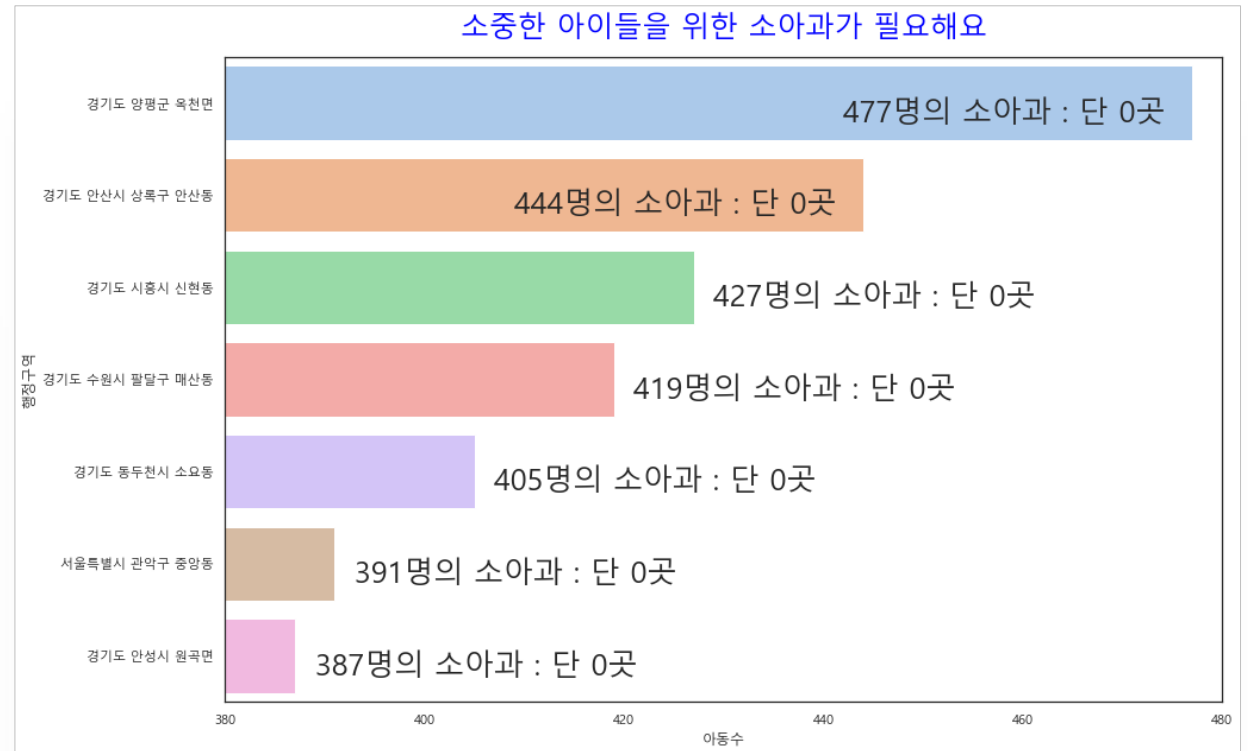
# d1 = 비율 높은 상위 10곳 d2 = 소아과가 없는 곳중 d1의 비율보다 아동이 많은 곳
d1 = df_2.copy()
d1 = d1[d1['비율']!=np.inf].sort_values('비율', ascending=False).head(10)
d1.index = range(d1.shape[0])

d2 = df_2[df_2['비율']==np.inf].sort_values('아동수', ascending=False).copy()
d2 = d2[(d2['아동수']) >= (d1['비율'].iloc[-1])]
d2.index = range(d2.shape[0])

plt.figure(figsize = (15,10))
g2 = sns.barplot(data=d2, x='아동수', y='행정구역', palette='pastel')
g2.set_xlim(left=380, right=480)
g2.set_title('소중한 아이들을 위한 소아과가 필요해요', fontdict={'fontsize': 24, 'color':'blue'}, pad=17)

for idx in range(len(d2)):
    text = d2['아동수'].iloc[idx].astype(str) + '명의 소아과 : 단 0곳'
    if d2['아동수'].iloc[idx] > 440 :
        plt.text(d2['아동수'].iloc[idx]-35, idx+0.2, text, size=25)
    else :
        plt.text(d2['아동수'].iloc[idx]+2, idx+0.2, text, size=25)

plt.savefig('data/4_8_그래프.png', bbox_inches='tight')
plt.show()
```



결론

소아과 제안 지역

연구의 시사점

연구의 한계점

종류 : 지도

대상 : 소아과 제안지역 (총17개 지역)

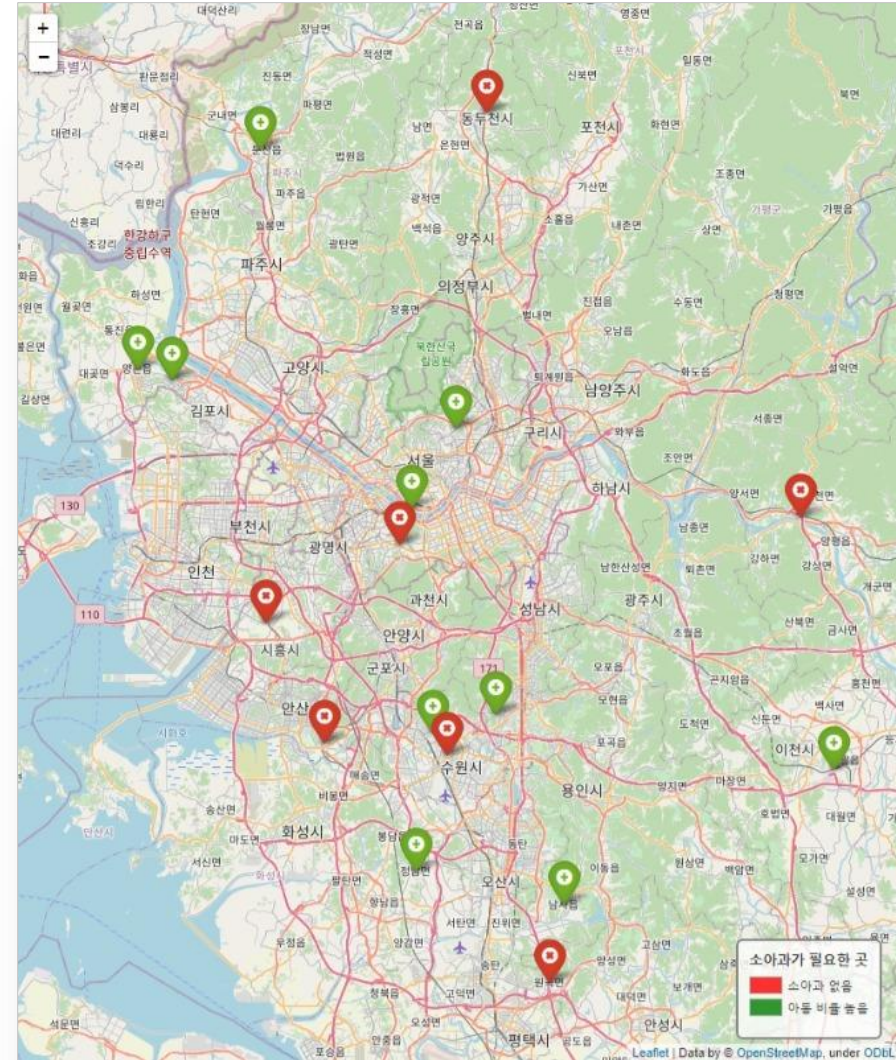
```
# d1 = 비율 높은 상위 10곳 d2 = 소아과가 없는 곳중 d1의 비율보다 아동이 많은 곳
# d3 = d1 + d2
d1 = df_2.copy()
d1 = d1[d1['비율']!=np.inf].sort_values('비율', ascending=False).head(10)
d1['구분'] = '아동 비율 높음'
d2 = df_2[df_2['비율']==np.inf].sort_values('아동수', ascending=False).copy()
d2 = d2[(d2['아동수']) >= (d1['비율'].iloc[-1])]
d2['구분'] = '소아과 없음'
d3 = pd.concat([d1, d2])
d3['읍_아_소'] = d3[['읍면동', '아동수']].astype(str).apply(lambda x : ' / 아동'.join(x), axis=1)
d3['읍_아_소'] = d3[['읍_아_소', '소아과']].astype(str).apply(lambda x : ' / 소아과'.join(x), axis=1)
d3.index = range(d3.shape[0])

import folium
lat_mean = d3['위도'].mean()
long_mean = d3['경도'].mean()
min_lon, max_lon = long_mean, long_mean
min_lat, max_lat = lat_mean, lat_mean

m = folium.Map(location=[lat_mean, long_mean], zoom_start=9,
               max_bounds=True, min_lat=min_lat, max_lat=max_lat, min_lon=min_lon, max_lon=max_lon)

for i in d3.index:
    lat = d3.loc[i, '위도']
    long = d3.loc[i, '경도']
    popup = d3.loc[i, '읍_아_소']
    if d3.loc[i, '구분'] == '소아과 없음':
        folium.Marker(location=[lat, long], tooltip=tooltip, icon=folium.Icon(icon="remove-sign", color='red')).add_to(m)
    elif d3.loc[i, '구분'] == '아동 비율 높음':
        folium.Marker(location=[lat, long], tooltip=tooltip, icon=folium.Icon(icon="plus-sign", color='green')).add_to(m)

from branca.element import Template, MacroElement
```




동두천시소요동
 서울시중랑동
 수원시매산동
 시흥시신현동
 안산시안산동
 안성시원곡면
 양평군옥천면
 김포시양촌읍
 김포시장기동
 서울시돈암동
 서울시이촌동
 수원시정자동
 용인시남사읍
 용인시성북동
 이천시부발읍
 파주시문산읍
 화성시정남면

1 아이와
가야하는 곳

2 자주
가야하는 곳

3 급히
가야하는 곳

4 가까이
더 가까이

 소중한
아이들을 위한
소아과



자료마다 **다른 주소방식**을 사용함



내과, 등 다른 **진료과목**도 소아과 목록에 포함



소아과 목록의 **작성시점**의 확인이 불가함



‘읍면동’이 다름에도 동일한 **도보생활권**인 지역 다수

Q & A

