

## Chapter 09 지리 정보 분석

# 목차

01 [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

02 [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

# 학습목표

- 수집한 주소 데이터를 정리하고 분석하여 시각화
- 주소를 분석하여 위도와 경도의 GPS 정보를 찾기
- 행정구역을 찾아서 지도에 나타난 정보분석

# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 분석 미리보기

- 목표 : 주소에 대한 지리 정보로 위치를 시각화 맵 생성
- 핵심개념 : 위도/경도 GPS 좌표, 지리정보 분석, 지리정보 시각화
- 데이터 수집
  - 커피매장 주소 데이터 (CoffeeBean.csv)
  - 행정구역 주소체계 데이터 (행정구역\_시구군\_별\_성별\_인구수.xlsx)
- 데이터 준비 및 탐색
  - 데이터 정제 : 주소의 행정구역 이름을 정확한 이름으로 수정
  - 데이터 조합 : 필요한 컬럼을 추출하고 병합

1. 포리움 라이브러리로 생성한 맵



2. 특정 주소의 위치를 시각화한 맵



# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 핵심 개념 이해

### ■ GPS 좌표 구하기

- 특정 위치의 GPS 좌표를 나타내는 지오 데이터는 구글맵이나 네이버 지도 서비스에서 구함
- 오픈 소프트웨어인 Geocoder-Xr사용 가능

### ■ 지리 정보 시각화 라이브러리

- 지오 데이터를 지도에 시각화, 지리 정보 시각화 라이브러리 사용(**folium**)

## ■ 데이터 수집

### • 주소 데이터

- map에 위치를 표시할 주소 데이터

### • 행정구역 주소 체계 데이터

- 국가통계포털 데이터로, 주소 데이터의 행정구역 이름을 정제할 기준
- GPS 위도와 경도의 좌표를 구하기 위해서, 주소 데이터의 행정구역 이름이 정확해야 함,  
-> 데이터 정제 작업 필요

# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 데이터 준비 및 탐색

### ■ 분석할 커피 매장의 주소 데이터 준비하기

#### 1. '주소데이터분석'으로 파일 이름을 변경 후 입력

In [1]:	<pre>import pandas as pd CB = pd.read_csv('CoffeeBean.csv', encoding = 'CP949', index_col = 0, header = 0, engine = 'python') CB.head() #작업 내용 확인용 출력</pre>																								
Out[1]:	<table><tr><th></th><th>store</th><th>address</th><th>phone</th></tr><tr><td>0</td><td>차병원점</td><td>서울시 강남구 논현로 566 강남차병원1층</td><td>02-538-7615</td></tr><tr><td>1</td><td>강남대로점</td><td>서울시 서초구 강남대로 369 1층</td><td>02-588-5778</td></tr><tr><td>2</td><td>청담에스점</td><td>서울시 강남구 압구정로 461 네이처포엠빌딩B108,109호</td><td>02-548-6052</td></tr><tr><td>3</td><td>신사점</td><td>서울시 강남구 도산대로 126</td><td>02-548-2741</td></tr><tr><td>4</td><td>역삼점</td><td>서울시 강남구 논현로 512 지상1,2층</td><td>02-569-8051</td></tr></table>		store	address	phone	0	차병원점	서울시 강남구 논현로 566 강남차병원1층	02-538-7615	1	강남대로점	서울시 서초구 강남대로 369 1층	02-588-5778	2	청담에스점	서울시 강남구 압구정로 461 네이처포엠빌딩B108,109호	02-548-6052	3	신사점	서울시 강남구 도산대로 126	02-548-2741	4	역삼점	서울시 강남구 논현로 512 지상1,2층	02-569-8051
	store	address	phone																						
0	차병원점	서울시 강남구 논현로 566 강남차병원1층	02-538-7615																						
1	강남대로점	서울시 서초구 강남대로 369 1층	02-588-5778																						
2	청담에스점	서울시 강남구 압구정로 461 네이처포엠빌딩B108,109호	02-548-6052																						
3	신사점	서울시 강남구 도산대로 126	02-548-2741																						
4	역삼점	서울시 강남구 논현로 512 지상1,2층	02-569-8051																						

In [1]: CoffeeBean.csv 파일을 CB 객체로 로드하고,  
상위 5개 행의 데이터를 출력하여 head() 확인  
파이썬에서 CSV 파일을 파싱하는 과정에서 문제가 발생할 수 있는데,  
-> 이를 해결하기 위해 engine = 'python' 속성 추가

# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 데이터 준비 및 탐색

### ■ 분석할 커피 매장의 주소 데이터 준비하기

#### 2. 주소 데이터를 행정구역 주소 체계에 맞게 정리하기

In [2]:	<pre>addr = [] for address in CB.address:     addr.append(str(address).split()) addr #작업 내용 확인용 출력</pre>
Out[2]:	<pre>['서울시', '강남구', '논현로', '566', '강남차병원1층'], ['서울시', '서초구', '강남대로', '369', '1층'], ['서울시', '강남구', '압구정로', '461', '네이처포엠빌딩B108,109호'], ... ['제주특별자치도', '제주시', '동문로', '8-1'], ['대전광역시', '서구', '둔산동', '1451,', '둔산타워', '제1층', '제101호']]</pre>

In [2]: for 반복문을 이용하여 각 address 컬럼의 값을 분리하고 `split()` addr 리스트로 만듦.

# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 데이터 준비 및 탐색

### ■ 분석할 커피 매장의 주소 데이터 준비하기

#### 2. 주소 데이터를 행정구역 주소 체계에 맞게 정리하기

In [3]:	<pre>addr2 = [] for i in range(len(addr)):     if addr[i][0] == "서울": addr[i][0] = "서울특별시"     elif addr[i][0] == "서울시": addr[i][0] = "서울특별시"     elif addr[i][0] == "부산시": addr[i][0] = "부산광역시"     elif addr[i][0] == "인천": addr[i][0] = "인천광역시"     elif addr[i][0] == "광주": addr[i][0] = "광주광역시"     elif addr[i][0] == "대전시": addr[i][0] = "대전광역시"     elif addr[i][0] == "울산시": addr[i][0] = "울산광역시"     elif addr[i][0] == "세종시": addr[i][0] = "세종특별자치시"     elif addr[i][0] == "경기": addr[i][0] = "경기도"     elif addr[i][0] == "충북": addr[i][0] = "충청북도"     elif addr[i][0] == "충남": addr[i][0] = "충청남도"     elif addr[i][0] == "전북": addr[i][0] = "전라북도"     elif addr[i][0] == "전남": addr[i][0] = "전라남도"     elif addr[i][0] == "경북": addr[i][0] = "경상북도"     elif addr[i][0] == "경남": addr[i][0] = "경상남도"     elif addr[i][0] == "제주": addr[i][0] = "제주특별자치도"     elif addr[i][0] == "제주도": addr[i][0] = "제주특별자치도"     elif addr[i][0] == "제주시": addr[i][0] = "제주특별자치도"     addr2.append(' '.join(addr[i])) addr2 #작업 내용 확인용 출력</pre>
Out[3]:	<pre>['서울특별시 강남구 논현로 566 강남차병원1층',  '서울특별시 서초구 강남대로 369 1층',  ...,  '제주특별자치도 제주시 동문로 8-1',  '대전광역시 서구 둔산동 1451, 둔산타워 제1층 제101호']</pre>

In [3]: if 조건 검사를 통해 기준에 맞지 않는 이름을 수정하면서 분리했던 항목을 연결하여 `join()` `addr2` 리스트를 만들



# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 데이터 준비 및 탐색

### ■ 분석할 커피 매장의 주소 데이터 준비하기

#### 2. 주소 데이터를 행정구역 주소 체계에 맞게 정리하기

In [4]:	addr2 = pd.DataFrame(addr2, columns = ['address2'])																														
In [5]:	CB2 = pd.concat([CB, addr2], axis = 1 ) CB2.head() #작업 내용 확인용 출력																														
Out[5]:	<table><tr><th></th><th>store</th><th>address</th><th>phone</th><th>address2</th></tr><tr><td>0</td><td>차병원점</td><td>서울시 강남구 논현로 566 강남차병원1층</td><td>02-538-7615</td><td>서울특별시 강남구 논현로 566 강남차병원1층</td></tr><tr><td>1</td><td>강남대로점</td><td>서울시 서초구 강남대로 369 1층</td><td>02-588-5778</td><td>서울특별시 서초구 강남대로 369 1층</td></tr><tr><td>2</td><td>청담에스점</td><td>서울시 강남구 압구정로 461 네이쳐포엠빌딩B108,109호</td><td>02-548-6052</td><td>서울특별시 강남구 압구정로 461 네이쳐포엠빌딩B108,109호</td></tr><tr><td>3</td><td>신사점</td><td>서울시 강남구 도산대로 126</td><td>02-548-2741</td><td>서울특별시 강남구 도산대로 126</td></tr><tr><td>4</td><td>역삼점</td><td>서울시 강남구 논현로 512 지상1,2층</td><td>02-569-8051</td><td>서울특별시 강남구 논현로 512 지상1,2층</td></tr></table>		store	address	phone	address2	0	차병원점	서울시 강남구 논현로 566 강남차병원1층	02-538-7615	서울특별시 강남구 논현로 566 강남차병원1층	1	강남대로점	서울시 서초구 강남대로 369 1층	02-588-5778	서울특별시 서초구 강남대로 369 1층	2	청담에스점	서울시 강남구 압구정로 461 네이쳐포엠빌딩B108,109호	02-548-6052	서울특별시 강남구 압구정로 461 네이쳐포엠빌딩B108,109호	3	신사점	서울시 강남구 도산대로 126	02-548-2741	서울특별시 강남구 도산대로 126	4	역삼점	서울시 강남구 논현로 512 지상1,2층	02-569-8051	서울특별시 강남구 논현로 512 지상1,2층
	store	address	phone	address2																											
0	차병원점	서울시 강남구 논현로 566 강남차병원1층	02-538-7615	서울특별시 강남구 논현로 566 강남차병원1층																											
1	강남대로점	서울시 서초구 강남대로 369 1층	02-588-5778	서울특별시 서초구 강남대로 369 1층																											
2	청담에스점	서울시 강남구 압구정로 461 네이쳐포엠빌딩B108,109호	02-548-6052	서울특별시 강남구 압구정로 461 네이쳐포엠빌딩B108,109호																											
3	신사점	서울시 강남구 도산대로 126	02-548-2741	서울특별시 강남구 도산대로 126																											
4	역삼점	서울시 강남구 논현로 512 지상1,2층	02-569-8051	서울특별시 강남구 논현로 512 지상1,2층																											
In [6]:	CB2.to_csv('CoffeeBean_2.csv', encoding = 'CP949', index = False)																														

In [4]: addr2를 DataFrame 타입으로 변경하고, 컬럼 이름을 address2로 지정

In [5]: CB와 addr2를 옆으로axis=1 결합하여concat() CB2를 만듦

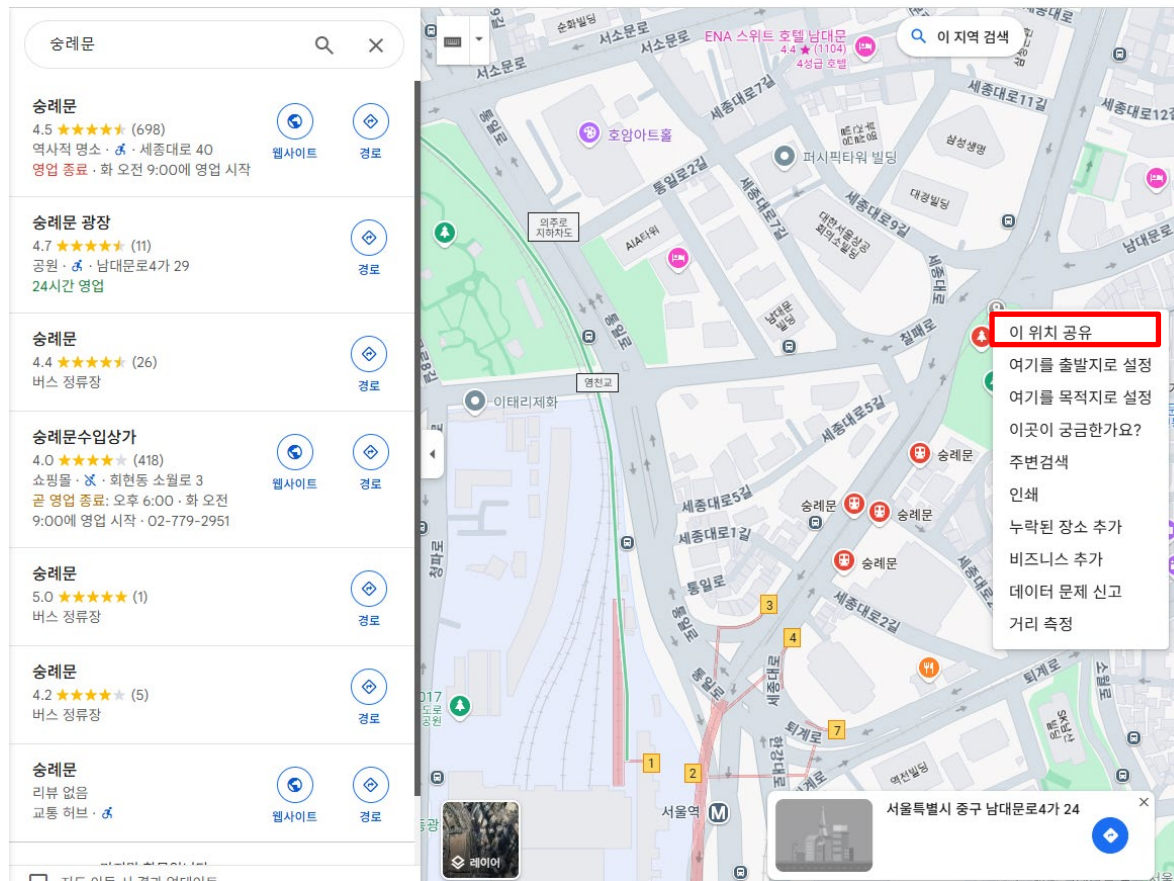
In [6]: 시도 이름이 수정된 데이터를 CB2로 저장. 작업 완료

# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 분석 모델 구축 및 시각화

### ■ 지도 객체 생성하기

1. 구글맵(maps.google.co.kr)에서 우리나라 국보1호인 '송례문'을 검색 → 마커 위에서 마우스 오른쪽 버튼을 클릭하여 [이 위치 공유]를 선택

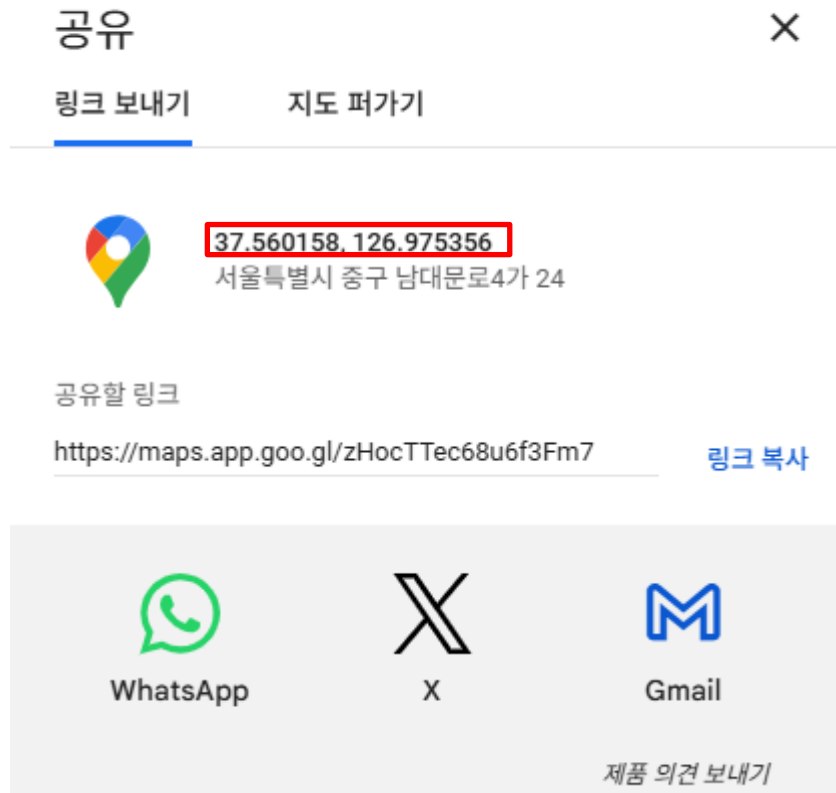


# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 분석 모델 구축 및 시각화

### ■ 지도 객체 생성하기

2. 송례문의 위도와 경도 좌표를 복사



# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 분석 모델 구축 및 시각화

### ■ 지도 객체 생성하기

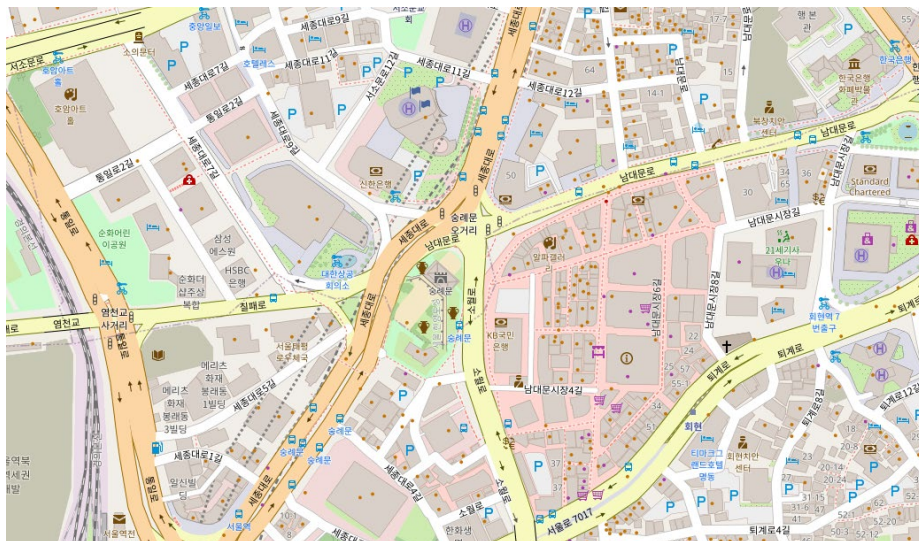
3. 복사한 좌표를 사용한 지도 객체를 생성

In [9]:	map_osm = folium.Map(location = [37.559978, 126.975291], zoom_start = 16)
In [10]:	map_osm.save('map.html')

In [9]: 복사한 송례문의 좌표를 `folium.Map()` 함수의 `location` 속성 값으로 설정  
지도의 크기를 확대하여 `zoom_start = 16` 지도 객체인 `map_osm`을 생성

In[10]: 생성한 지도 객체 `map_osm`를 파일로 저장

4. 저장된 `map.html` 파일을 더블클릭해서 열어 지도 파일을 확인



# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기


## ■ 분석 모델 구축 및 시각화

- [www.geoservice.co.kr](http://www.geoservice.co.kr) 회원가입 후, [변환-지오코딩- 주소를 좌표로 변환]



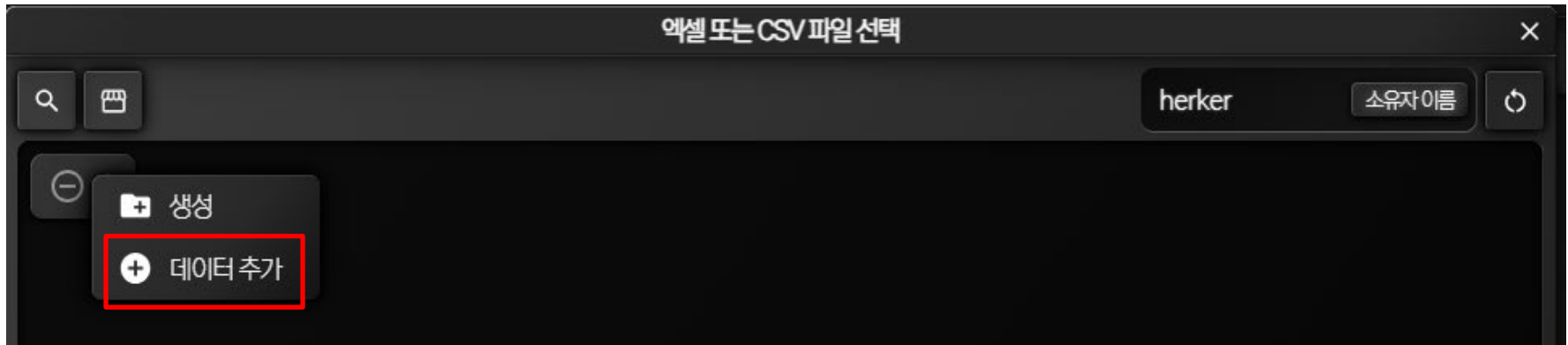
## 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

### ■ 분석 모델 구축 및 시각화

- [지오코딩]창에서 상단 왼쪽  (아카이브에서 데이터 불러오기) 버튼



- [엑셀 또는 CSV 파일선택] 창에서 [데이터 추가]



# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 분석 모델 구축 및 시각화

- [아카이브에 데이터 업로드]창의 [데이터 파일]에 CoffeeBean\_2.csv를 드래그하여 추가, 업로드

아카이브에 데이터 업로드

제목: CoffeeBean\_2.csv

아카이브 폴더: /

데이터 파일: CoffeeBean\_2.csv  
비공인 데이터가 선택되었습니다.

EPSG: [Dropdown]

한글 인코딩: EUC-KR

설명: 설명을 입력하세요.

☒ 공유 데이터

추가 취소



# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 분석 모델 구축 및 시각화

- 업로드된 CoffeeBean\_2.csv를 클릭하여 파일을 열면 깨져나옴,
- 팝업에서 인코딩 문제를 해결

지오코딩 (주소를 좌표로 변환)

주소 필드

☒ 정좌표 ☒ 유사주소좌표 ☒ 대표좌표 ☒ 정제된 주소 ☒ 매칭코드

예) 서울시 주소 머리값

No	store	address	phone	address2
0	e DT	e 211 1	02-3444-9973	U
1		280 7248	02-3412-2326	U
2		566 1	02-538-7615	U
3		369 1	02-588-5778	U
4	□□	Z 159 2	02-6002-3320	U
5	6	6 46 3	02-541-5832	U
6			02-593-5095	U
7	0 完		02-548-6052	U
8	Z		02-548-2741	U
9	6		02-544-6823	U
10			02-569-8051	U
11	黑 Y		02-578-6833	U
12	0 威	Z 716 LEE76 2	02-542-2053	U
13		736-17 B 2	02-3443-2096	U

제가 보기에 한글이 깨져 보이는 것 같습니다.  
정말로 한글이 깨져 보인다면 이를 지금 해결할까요?

예 아니오



# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 분석 모델 구축 및 시각화

- [주소필드]에서 변환할 필드(열)을 [address2]로 선택하고 실행버튼

지오코딩(주소를 좌표로 변환)

주소필드

☒ 정좌표 ☒ 유사주소좌표 ☒ 대표좌표 ☒ 정제된 주소 ☒ 매칭코드 ☒ 예) 서울시 주소머리값

store

address

phone

address2

No	store	address	phone	address2
0	차병원	서울시 강남구 논현로 566 강남차병원1층	02-538-7615	서울특별시 강남구 논현로 566 강남차병원1층
1	강남	서울시 서초구 강남대로 369 1층	02-588-5778	서울특별시 서초구 강남대로 369 1층
2	청담	서울시 강남구 압구정로 461 네이처포엠빌딩B108,109호	02-548-8052	서울특별시 강남구 압구정로 461 네이처포엠빌딩B108,109호
3	신사	서울시 강남구 도산대로 126	02-548-2741	서울특별시 강남구 도산대로 126
4	역삼점	서울시 강남구 논현로 512 지상1,2층	02-569-8051	서울특별시 강남구 논현로 512 지상1,2층
5	청담성당점	서울시 강남구 삼성로 716 LEE76빌딩2층	02-542-2053	서울특별시 강남구 삼성로 716 LEE76빌딩2층
6	영동점	서울 서초구 반포동 736-17 P빌딩 2층	02-3443-2096	서울특별시 서초구 반포동 736-17 P빌딩 2층
7	도곡점	서울시 강남구 언주로 30길 10,112 현대비전21 112호	02-572-2781	서울특별시 강남구 언주로 30길 10,112 현대비전21 112호
8	영동고앞점	서울시 강남구 선릉로 749 1,2층	02-544-3794	서울특별시 강남구 선릉로 749 1,2층
9	압구정시티점	서울시 강남구 도산대로49길 13 1층 17,18호	02-543-5922	서울특별시 강남구 도산대로49길 13 1층 17,18호
10	서초우성점	서울시 서초구 강남대로 51길 1 511 TOWER 1층	02-582-0158	서울특별시 서초구 강남대로 51길 1 511 TOWER 1층
11	논현팍스타워점	서울시 강남구 논현 231-13호 팍스타워지하1층	02-513-3870	서울특별시 강남구 논현 231-13호 팍스타워지하1층
12	삼성오크우드점	서울 강남구 테헤란로87길 46 지하 2층	02-3466-8507	서울특별시 강남구 테헤란로87길 46 지하 2층
13	트레이드타워점	서울시 강남구 영동대로 511 트레이드타워 지하1층	02-803-9994	서울특별시 강남구 영동대로 511 트레이드타워 지하1층
14	삼성봉은사거리점	서울시 강남구 영동대로 607 1,2층	02-3443-5618	서울특별시 강남구 영동대로 607 1,2층
15	서초치파이브점	서울시 서초구 서초동 1685-8호 101~2호,113~4호,121호	02-3477-5580	서울특별시 서초구 서초동 1685-8호 101~2호,113~4호,121호
16	논현동수면센터점	서울시 강남구 논현로 717 1층	02-3443-5575	서울특별시 강남구 논현로 717 1층

# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 분석 모델 구축 및 시각화

- 생성된 지오코딩 파일 이름을 'CB\_geo.zip'으로 입력하고 아카이브에 업로드



아카이브에 데이터 업로드

제목 **CB\_geo.zip**

아카이브 폴더 /

설명

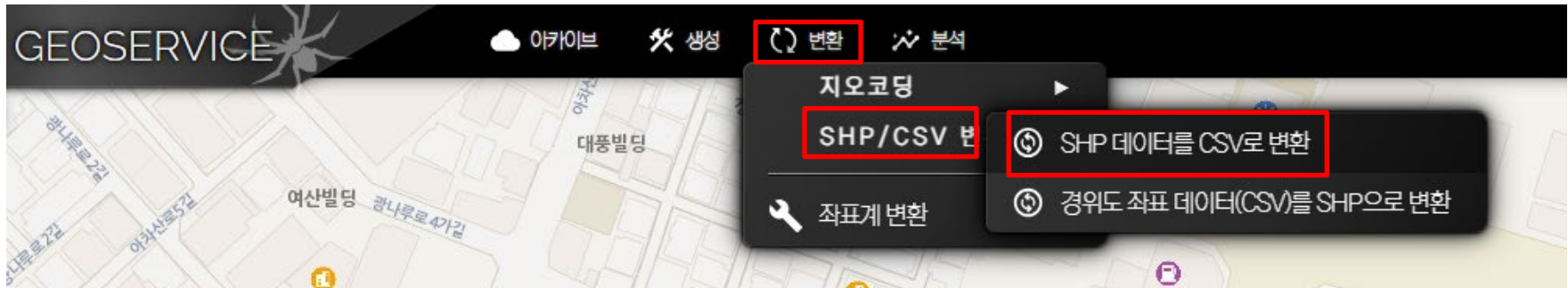
☒ 공유 데이터

**추가** 취소

# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 분석 모델 구축 및 시각화

- [변환]-[SHP/CSV 변환]-[SHP 데이터를 CSV로 변환]



# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 분석 모델 구축 및 시각화

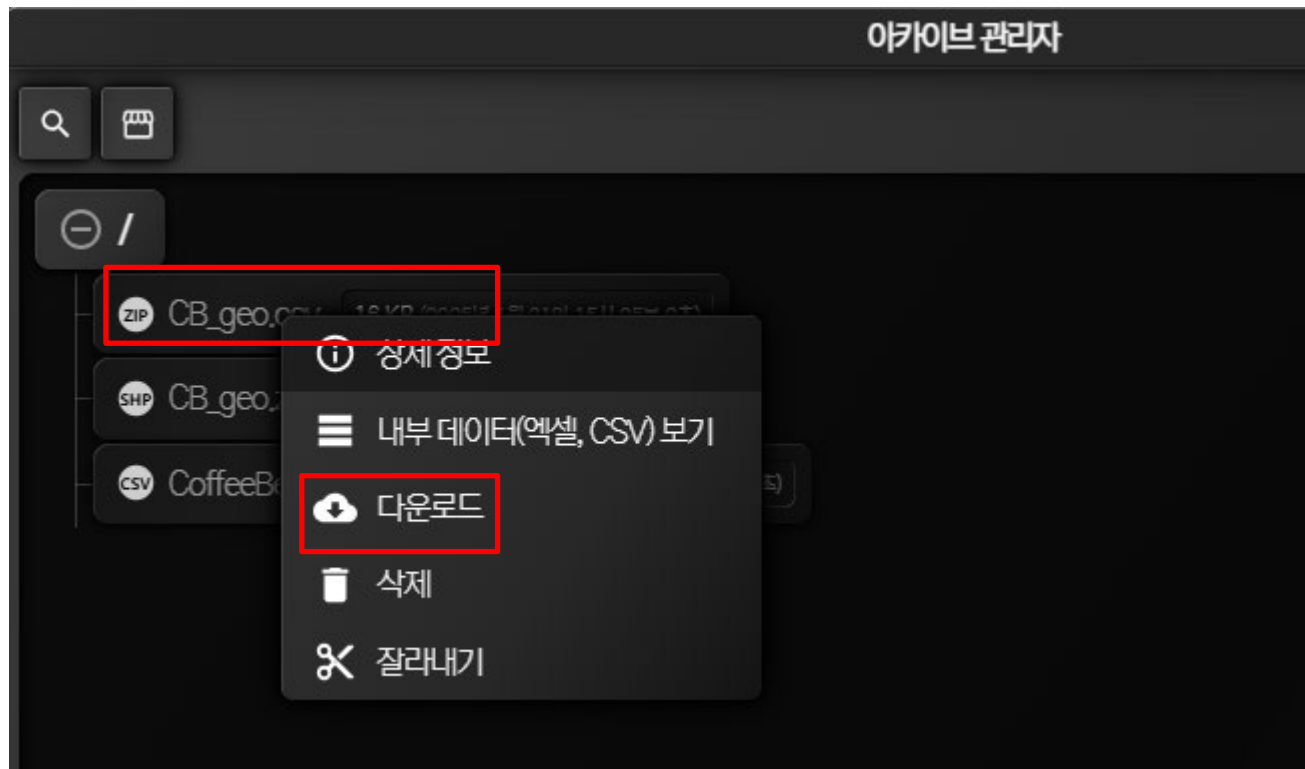
- [SHP데이터를 CSV로 변환] 메뉴에서 'CB\_geo.csv'파일을 아카이브에 업로드



## 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

### ■ 분석 모델 구축 및 시각화

- 변환 작업이 끝난 CB\_geo.csv파일을 [아카이브 관리자]에서 다운로드



# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 분석 모델 구축 및 시각화

### ■ 지도 객체에 커피 매장 위치 표시하기

6. CB\_geo.shp.csv 파일을 열어서 지오데이터 확인 ( \_X: 경도, \_Y: 위도)

	A	B	C	D	E	F	G	H
1	store	address	phone	address2	_GC_TYPE	_CLEANAD	_X	_Y
2	학동역 DT	서울시 강	02-3444-9	서울특별시정	서울특별시	서울특별시	127.0321	37.51469
3	메가박스점	서울 강남	02-6002-3	서울특별시정	서울특별시	서울특별시	127.0588	37.5126
4	강남에스점	서울시 서	02-593-50	서울특별시정	서울특별시	서울특별시	127.0258	37.50046
5	강남대로점	서울시 서	02-588-57	서울특별시정	서울특별시	서울특별시	127.0282	37.49567
6	수서점	서울시 강	02-3412-2	서울특별시정	서울특별시	서울특별시	127.1031	37.48731
7	차병원점	서울시 강	02-538-76	서울특별시정	서울특별시	서울특별시	127.0347	37.50704
8	압구정에스	서울시 강	02-541-58	서울특별시유	서울특별시	서울특별시	127.0366	37.52839
9	청담에스점	서울시 강	02-548-60	서울특별시정	서울특별시	서울특별시	127.0472	37.52525
10	신사점	서울시 강	02-548-27	서울특별시정	서울특별시	서울특별시	127.0224	37.51697
11	압구정역점	서울시 강	02-544-68	서울특별시정	서울특별시	서울특별시	127.0287	37.52396
12	영동점	서울 서초	02-3443-2	서울특별시정	서울특별시	서울특별시	127.0228	37.50679
13	역삼점	서울시 강	02-569-80	서울특별시정	서울특별시	서울특별시	127.0365	37.50235
14	도곡점	서울시 강	02-572-27	서울특별시유	서울특별시	서울특별시	127.0516	37.48734

# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 분석 모델 구축 및 시각화

### ■ 지도 객체에 커피 매장 위치 표시하기

#### 8. 지도에 매장 위치 표시하기

In [11]:	<code>CB_geoData = pd.read_csv('CB_geo.csv', encoding = 'utf-8', engine = 'python')</code>
In [12]:	<code>map_CB = folium.Map(location = [37.560284, 126.975334], zoom_start = 15)</code>
In [13]:	<code>for i, store in CB_geoData.iterrows():     folium.Marker(location = [store['_Y'], store['_X']], popup =                     store['store'], icon = folium.Icon(color = 'red',                     icon = 'star')).add_to(map_CB)</code>
In [14]:	<code>map_CB.save('map_CB.html')</code>
In [15]:	<code>import webbrowser webbrowser.open('map_CB.html')</code>

In [11]: CB\_geo.shp\_2.csv 파일을 CB\_geoData 객체로 로드

In [12]: 지도 객체인 map\_CB를 생성

In [13]: for 반복문을 사용하여 CB\_geoData 객체에 있는 매장 정보를 하나씩 읽음

매장에 대한 마커의 팝업 글자는 매장 이름으로 설정하고 `popup = store['store']`, 마커 모양은 빨간색 별 모양으로  
설정하여 `icon = 'star'` 마커를 만든 뒤 `folium.Marker()` 지도 객체 `map_CB`에 추가 `add_to(map_CB)`.

In [14]: 완성된 지오맵을 저장



# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

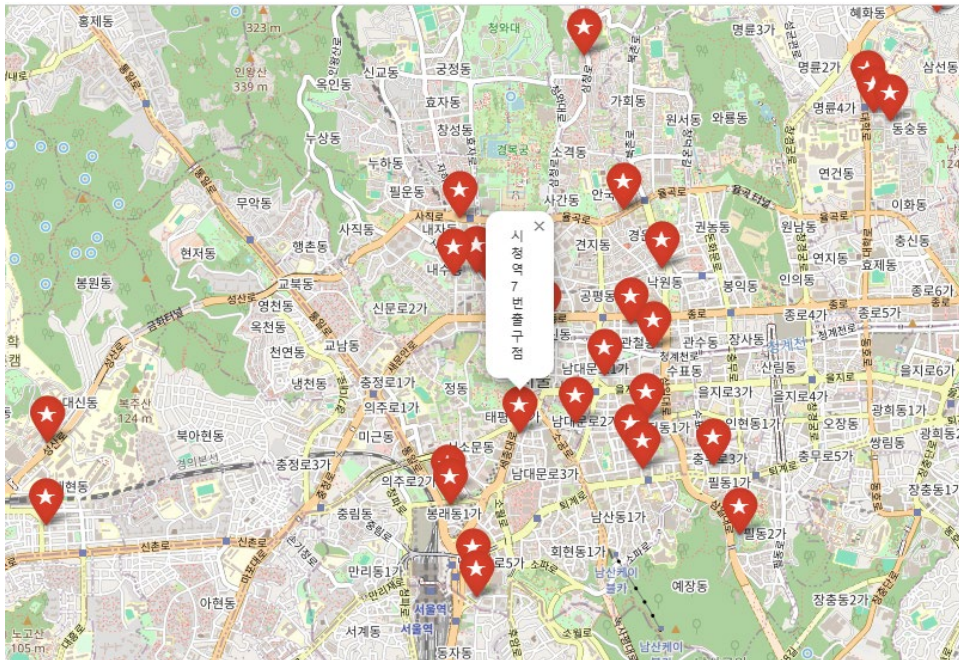
## ■ 분석 모델 구축 및 시각화

- 지도 객체에 커피 매장 위치 표시하기

### 8. 지도에 매장 위치 표시하기

```
In [15]: import webbrowser  
webbrowser.open('map_CB.html')
```

In [15]: 저장한 지오맵 파일을 웹 브라우저에서 열어 확인







# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 데이터 수집

### ■ 주소 데이터 수집하기

– CoffeeBean.csv 파일 준비

### ■ 행정구역 주소 체계 데이터 수집하기

1. 국가통계포털 사이트(<http://kosis.kr>)에서 '행정구역'으로 데이터를 검색

KOSIS 국가통계포털  
Korean Statistical Information Service

국내통계 국제·북한통계 쉽게 보는 통계 온라인간행물 민원안내 서비스 소개

통합검색 행정구역 상세검색

통합검색 통계표 통계시각화콘텐츠 통계분류 온라인간행물 통계설명자료

인기통계 주민등록인구현황 도시계획현황

"행정구역"에 대한 검색결과 20,662건입니다.

통계표 : 20,592건

행정구역(읍면동)별/5세별 주민등록인구(2011년~)

행정구역(시군구)/산업별 고용

# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 데이터 수집

### ■ 행정구역 주소 체계 데이터 수집하기

2. 검색 결과 중에서 '주민등록인구현황:행정구역(시군구)별, 성별 인구수'를 선택

통합검색

통계표

통계시각화콘텐츠

통계분류

온라인간행물

통계설명자료

통계용어

게시물

인기통계

주민등록인구현황

도시계획현황

사업체노동력조사

"행정구역"에 대한 검색결과는 20,556건입니다.

통계표 20,556 건

국내통계

e-지방지표

국제통계

북한통계

국제지역통계

정확도순

20개

가나다순

건수순

기본정렬

주제별(20,004)

인구(1,332)

사회일반(1,789)

범죄·안전(53)

노동(228)

보건(217)

복지(47)

교육·훈련(4)

행정구역(읍면동)별/5세별 주민등록인구(2011년~)

행정안전부, 주민등록인구현황, 2011~2024

행정구역(시군구)/산업별 고용

고용노동부, 사업체노동력조사, 2018~2024

행정구역(시도)/산업별(대분류) 고용

고용노동부, 사업체노동력조사, 2018~2024

행정구역(시군구)별 주민등록세대수

추천 인기

행정안전부, 주민등록인구현황, 1992~2024

행정구역(시군구)별, 성별 인구수

추천 인기

행정안전부, 주민등록인구현황, 1992~2024

행정구역(시군구)별/1세별 주민등록인구

행정안전부, 주민등록인구현황, 2008~2024

# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 데이터 수집

### ■ 행정구역 주소 체계 데이터 수집하기

2. 행정구역이 상위 레벨(시도)만 있고 하위 레벨(군구)은 보이지 않음

1. 행정구역(시군구)별, 성별 ... X 모두 닫기									
1) 행정구역(시군구)별, 성별 인구수									
「주민등록인구현황」 행정안전부 (자료문의처: 044-205-3158) 통계설명자료   온라인간행물									
수록기간 : 월, 년 1992 ~ 2024.12 / 자료갱신일: 2025-01-13 / 주석정보									
시점 증감/증감률 행렬전환 열고정해제 새 탭 열기 화면복사 주소/출처 스크랩 OPENAPI 인쇄 다운로드 조회설정									
행정구역(시군구)별	2024.10			2024.11			2024.12		
	총인구수 (명)	남자인구수 (명)	여자인구수 (명)	총인구수 (명)	남자인구수 (명)	여자인구수 (명)	총인구수 (명)	남자인구수 (명)	여자인구수 (명)
전국	51,238,450	25,511,290	25,727,160	51,226,757	25,504,176	25,722,581	51,217,221	25,498,324	
서울특별시	9,343,304	4,511,734	4,831,570	9,335,495	4,507,664	4,827,831	9,331,828	4,505,355	
2) 부산광역시	3,271,062	1,592,440	1,678,622	3,268,993	1,591,231	1,677,762	3,266,598	1,589,912	
대구광역시	2,364,892	1,160,429	1,204,463	2,364,334	1,160,074	1,204,260	2,363,629	1,159,601	
인천광역시	3,017,928	1,507,849	1,510,079	3,018,589	1,508,189	1,510,400	3,021,010	1,509,243	
광주광역시	1,410,548	696,227	714,321	1,409,624	695,854	713,770	1,408,422	695,224	
대전광역시	1,440,094	717,893	722,201	1,439,687	717,664	722,023	1,439,157	717,291	
울산광역시	1,098,540	565,011	533,529	1,098,312	564,865	533,447	1,098,049	564,888	
3) 세종특별자치시	389,978	194,110	195,868	390,603	194,407	196,196	390,685	194,413	
경기도	13,686,731	6,879,539	6,807,192	13,690,645	6,880,565	6,810,080	13,694,685	6,882,186	
강원특별자치도	1,519,545	764,109	755,436	1,518,700	763,577	755,123	1,517,766	763,025	
충청북도	1,590,704	809,530	781,174	1,591,100	809,697	781,403	1,591,177	809,920	
충청남도	2,136,010	1,095,772	1,040,238	2,136,632	1,096,190	1,040,442	2,136,574	1,096,128	
전북특별자치도	1,740,753	866,902	873,851	1,739,751	866,399	873,352	1,738,690	866,030	
전라남도	1,790,421	903,270	887,151	1,789,610	902,758	886,852	1,788,819	902,380	
경상북도	2,535,971	1,282,754	1,253,217	2,533,979	1,281,907	1,252,072	2,531,384	1,280,547	
경상남도	3,231,132	1,628,096	1,603,036	3,230,071	1,627,652	1,602,419	3,228,380	1,626,926	
제주특별자치도	670,837	335,625	335,212	670,632	335,483	335,149	670,368	335,255	

# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 데이터 수집

### ■ 행정구역 주소 체계 데이터 수집하기

3. 행정구역(시군구)별 탭을 클릭하고, [2 레벨 전체선택]을 체크해서 선택

「주민등록인구현황」 행정안전부 (자료문의처: 044-205-3158) 통계설명자료 온라인간행물

조회조건 부가기능

화면복사 주소/출처 스크랩

조회

인구(명)	남자인구수(명)	여자인구수(명)
226,757	25,504,176	25,72
335,495	4,507,664	4,82
138,577	66,673	7
120,931	58,247	6
204,079	97,628	10
274,287	132,769	14
332,571	159,576	17
339,611	166,095	
379,541	185,591	
422,141	202,309	
283,934	136,992	14
303,468	146,848	15
492,072	235,878	25
461,086	218,310	24
303,453	143,714	15
361,841	168,544	19
431,623	210,547	22
557,095	266,725	29
388,488	190,209	19
225,371	113,304	11

항목

행정구역(시군구)별

검색하기 검색

하위레벨 전체선택 ?

1, 2 레벨 전체해제

접기

- ☒ 전국
- ☒ 서울특별시
  - ☒ 종로구
  - ☒ 중구
  - ☒ 용산구
  - ☒ 성동구
  - ☒ 광진구
  - ☒ 동대문구
  - ☒ 중랑구
  - ☒ 성북구
  - ☒ 강북구
  - ☒ 도봉구

# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 데이터 수집

### ■ 행정구역 주소 체계 데이터 수집하기

4. [시점] 탭을 클릭하고, 시점 목록에서 2025.04을 선택한 뒤 버튼을 클릭

The screenshot shows a web application interface for data collection. At the top, there are two tabs: '조회조건' (Search Conditions) and '부가기능' (Additional Functions). The '조회조건' tab is active. Below the tabs, there is a search button labeled 'Q 조회'. Underneath, there are three expandable sections: '항목' (Items), '행정구역(시군구)별' (By Administrative Area (City/Gun/Gu)), and '시점' (Period). The '시점' section is expanded, showing a date selection interface. It has two buttons: '월' (Month) and '년' (Year). The '월' button is selected, indicated by a green checkmark. Below these buttons, there are two dropdown menus for selecting the start and end dates. The start date is '2025.02' and the end date is '2025.04'. Below the date selection, there is a checkbox labeled '전체선택' (Select All), which is currently unchecked. At the bottom, there is a list of months with checkboxes: '2025.04' is checked with a green checkmark, '2025.03' is unchecked, and '2025.02' is unchecked.

# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 데이터 수집

### ■ 행정구역 주소 체계 데이터 수집하기

5. 파일형태를 'EXCEL'로 선택하고 버튼을 클릭

The screenshot shows a web application interface for downloading population data by administrative district. The main table lists districts and their population for April 2025. A download dialog box is open, allowing the user to choose the file format and options.

행정구역(시군구)별	2025.04 총인구수 (명)	남자인구수 (명)	여자인구수 (명)
전국	51,175,725	25,474,633	25,701,092
서울특별시	9,331,421	4,681,111	4,650,310
충청도	1,384,111	681,111	703,000
충청	119,421	59,111	60,310
충청	203,421	101,111	102,310
2) 성동구	273,421	136,111	137,310
광진구	332,421	166,111	166,310
동대문구	342,421	171,111	171,310
중랑구	377,421	188,111	189,310
성북구	421,421	210,111	211,310
강북구	282,421	141,111	141,310
3) 도봉구	302,421	151,111	151,310
노원구	488,421	244,111	244,310
은평구	459,421	229,111	230,310
서대문구	301,421	150,111	151,310
마포구	360,421	180,111	180,310
양천구	428,421	214,111	214,310
강서구	553,421	276,111	277,310
4) 구로구	386,421	193,111	193,310
금천구	224,421	112,111	112,310
영등포구	372,421	186,111	186,310
동작구	375,421	187,111	188,310
관악구	479,421	239,111	240,310
서초구	407,119	193,142	214,037
강남구	556,822	266,037	290,785
송파구	647,426	310,186	337,240

Download Dialog Box:

- File Type: ☒ EXCEL(xlsx) ☐ EXCEL(xls) ☒ 셀 병합
- CSV (인코딩: ANSI)
- TXT (인코딩: ANSI)
- SDMX(2.0) [DSD (데이터구조) DATA Generic]
- 시점정렬: ☒ 오름차순 ☐ 내림차순
- 소수점: ☐ 수록자료형식과 동일 ☒ 조화화면과 동일

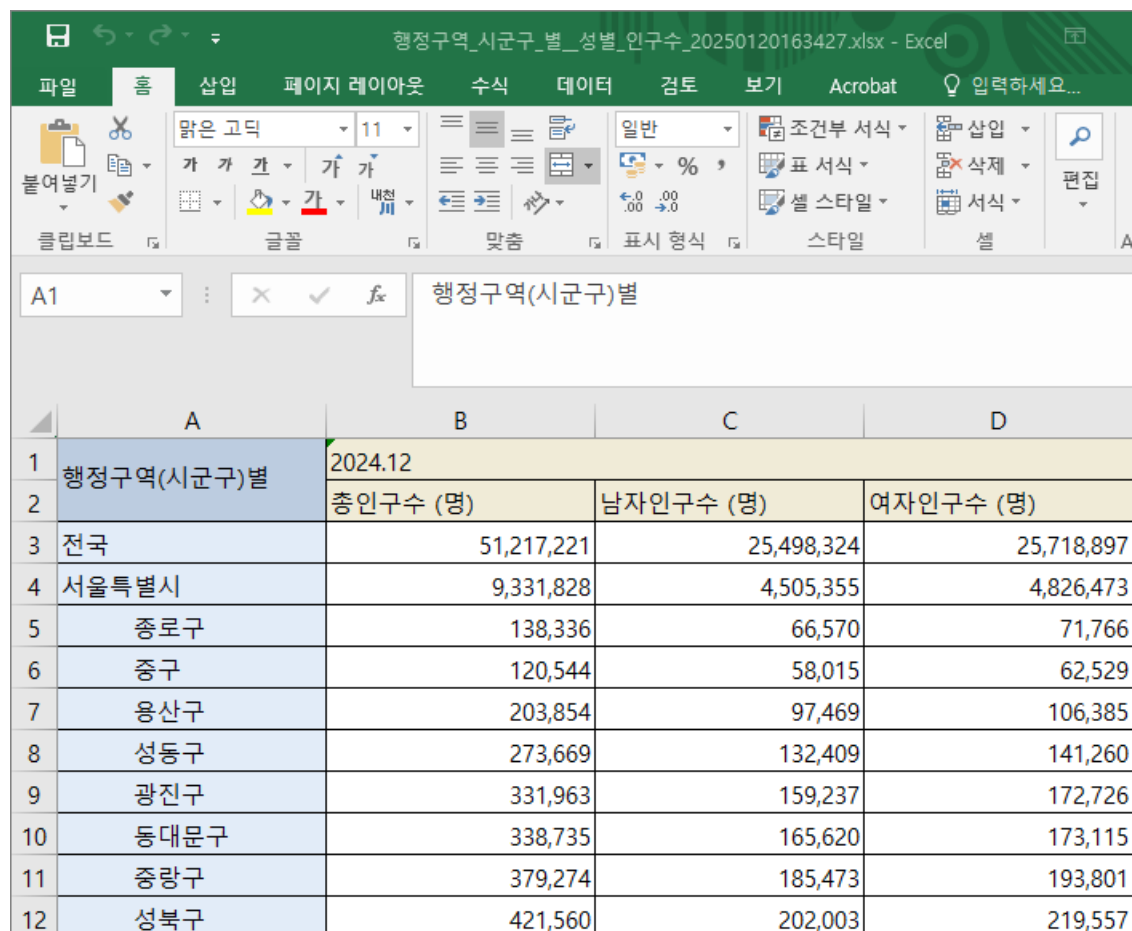
6. 다운로드된 파일은 이름을 '행정구역\_시군구\_별\_성별\_인구수.xlsx'로 수정한 뒤에 작업 폴더로 이동

# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 데이터 준비 및 탐색

### ■ 행정구역 주소 체계 데이터 준비하기

1. 국가통계포털에서 다운로드한 '행정구역\_시군구\_별\_성별\_인구수.xlsx' 파일을 열어서 확인



The screenshot shows an Excel spreadsheet titled '행정구역\_시군구\_별\_성별\_인구수\_20250120163427.xlsx'. The data is organized in a table with columns A, B, C, and D. Column A lists administrative regions, column B shows the total population, column C shows the male population, and column D shows the female population. The data is for December 2024.

	A	B	C	D
1	행정구역(시군구)별	2024.12		
2		총인구수 (명)	남자인구수 (명)	여자인구수 (명)
3	전국	51,217,221	25,498,324	25,718,897
4	서울특별시	9,331,828	4,505,355	4,826,473
5	종로구	138,336	66,570	71,766
6	중구	120,544	58,015	62,529
7	용산구	203,854	97,469	106,385
8	성동구	273,669	132,409	141,260
9	광진구	331,963	159,237	172,726
10	동대문구	338,735	165,620	173,115
11	중랑구	379,274	185,473	193,801
12	성북구	421,560	202,003	219,557

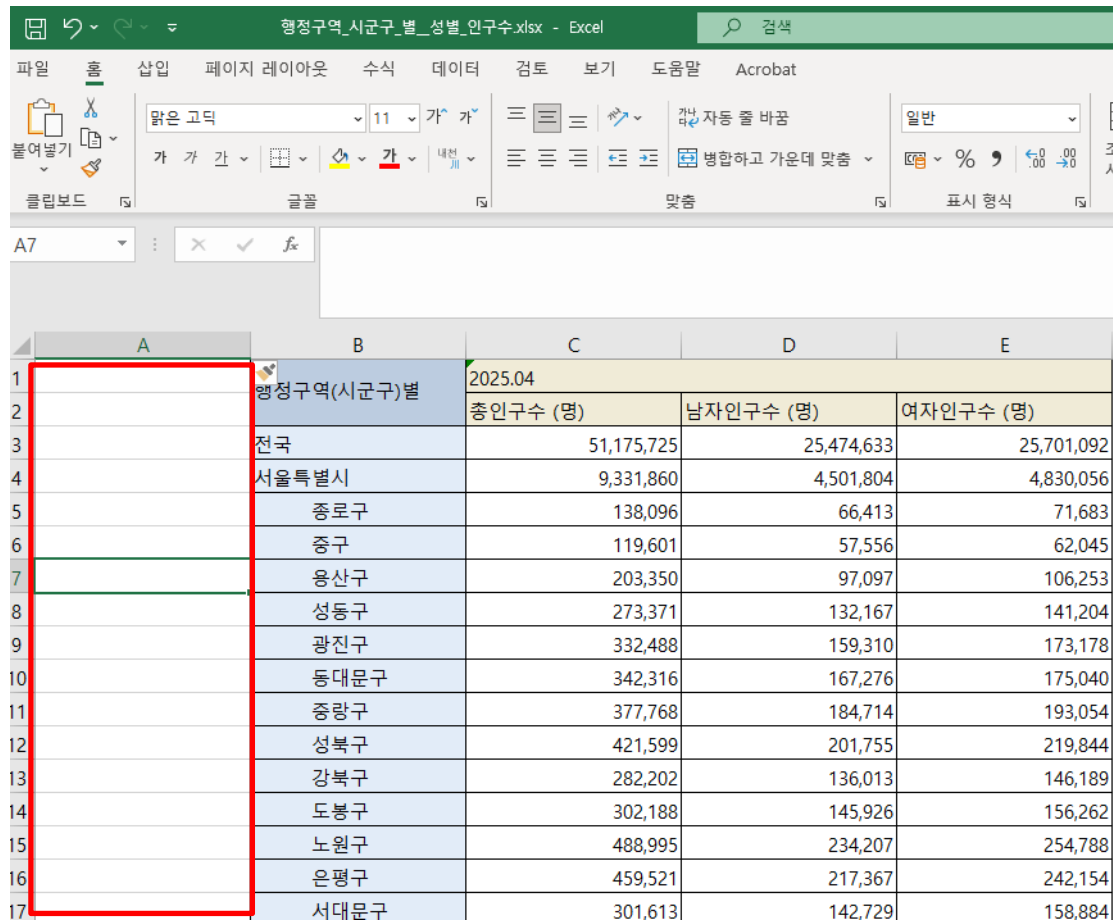


# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 데이터 준비 및 탐색

### ■ 행정구역 주소 체계 데이터 준비하기

2. 엑셀에서 데이터 정리하기 먼저, 왼쪽에 빈 열을 삽입



Excel spreadsheet showing administrative district data (행정구역\_시군구\_별\_성별\_인구수.xlsx). The data is organized in columns A through E. Column A is highlighted with a red box, indicating it is the target for data insertion.

	A	B	C	D	E
1		행정구역(시군구)별	2025.04		
2			총인구수 (명)	남자인구수 (명)	여자인구수 (명)
3		전국	51,175,725	25,474,633	25,701,092
4		서울특별시	9,331,860	4,501,804	4,830,056
5		종로구	138,096	66,413	71,683
6		중구	119,601	57,556	62,045
7		용산구	203,350	97,097	106,253
8		성동구	273,371	132,167	141,204
9		광진구	332,488	159,310	173,178
10		동대문구	342,316	167,276	175,040
11		중랑구	377,768	184,714	193,054
12		성북구	421,599	201,755	219,844
13		강북구	282,202	136,013	146,189
14		도봉구	302,188	145,926	156,262
15		노원구	488,995	234,207	254,788
16		은평구	459,521	217,367	242,154
17		서대문구	301,613	142,729	158,884

# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 데이터 준비 및 탐색

### ■ 행정구역 주소 체계 데이터 준비하기

3. 시 이름을 1 레벨로, 군/구 이름을 2 레벨로 옮기는 작업을 실시  
1 레벨로 분리할 시 이름을 잘라내어 A열에 붙여넣음

행정구역\_시군구\_별\_성별\_인구수\_20250120163427.xlsx - Excel

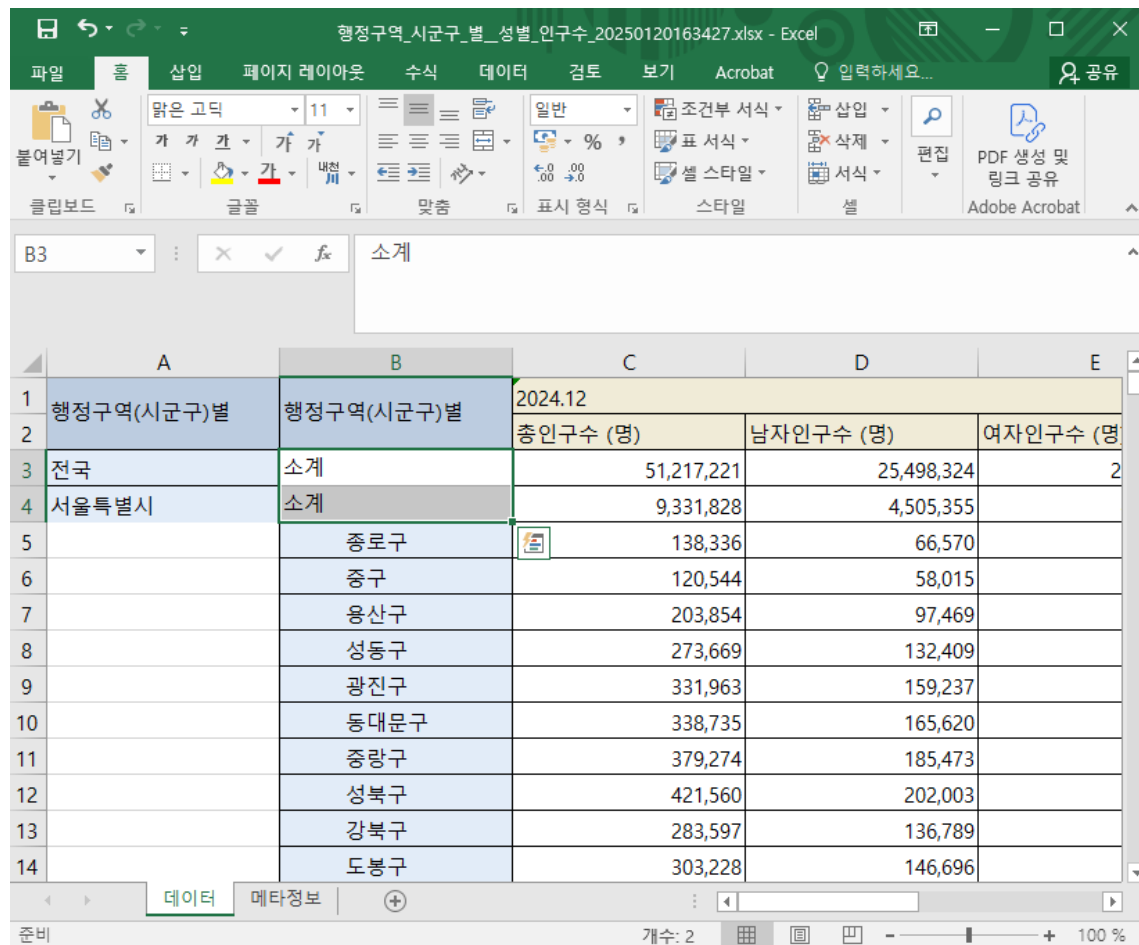
행정구역(시군구)별	행정구역(시군구)별	2024.12		
		총인구수 (명)	남자인구수 (명)	여자인구수 (명)
전국		51,217,221	25,498,324	2
서울특별시		9,331,828	4,505,355	
	종로구	138,336	66,570	
	중구	120,544	58,015	
	용산구	203,854	97,469	
	성동구	273,669	132,409	
	광진구	331,963	159,237	
	동대문구	338,735	165,620	
	중랑구	379,274	185,473	
	성북구	421,560	202,003	
	강북구	283,597	136,789	
	도봉구	303,228	146,696	

# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 데이터 준비 및 탐색

### ■ 행정구역 주소 체계 데이터 준비하기

4. 잘라낸 시 이름 자리에 합계를 나타내는 '소계'를 입력



행정구역(시군구)별	행정구역(시군구)별	2024.12		
		총인구수 (명)	남자인구수 (명)	여자인구수 (명)
전국	소계	51,217,221	25,498,324	2
서울특별시	소계	9,331,828	4,505,355	
	종로구	138,336	66,570	
	중구	120,544	58,015	
	용산구	203,854	97,469	
	성동구	273,669	132,409	
	광진구	331,963	159,237	
	동대문구	338,735	165,620	
	중랑구	379,274	185,473	
	성북구	421,560	202,003	
	강북구	283,597	136,789	
	도봉구	303,228	146,696	

# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 데이터 준비 및 탐색

### ■ 행정구역 주소 체계 데이터 준비하기

5. 엑셀의 채우기 기능을 이용해 빈 자리에 시 이름을 복사

파일홈삽입페이지 레이아웃수식데이터검토보기Acrobat수행할 작업을 알려 주세요.공유

붙여넣기

클립보드

글꼴

맞춤

표시 형식

스타일

일반

조건부 서식

삽입

삭제

서식

Σ

PDF 생성 및 링크 공유

Adobe Acrobat

A4: 서울특별시

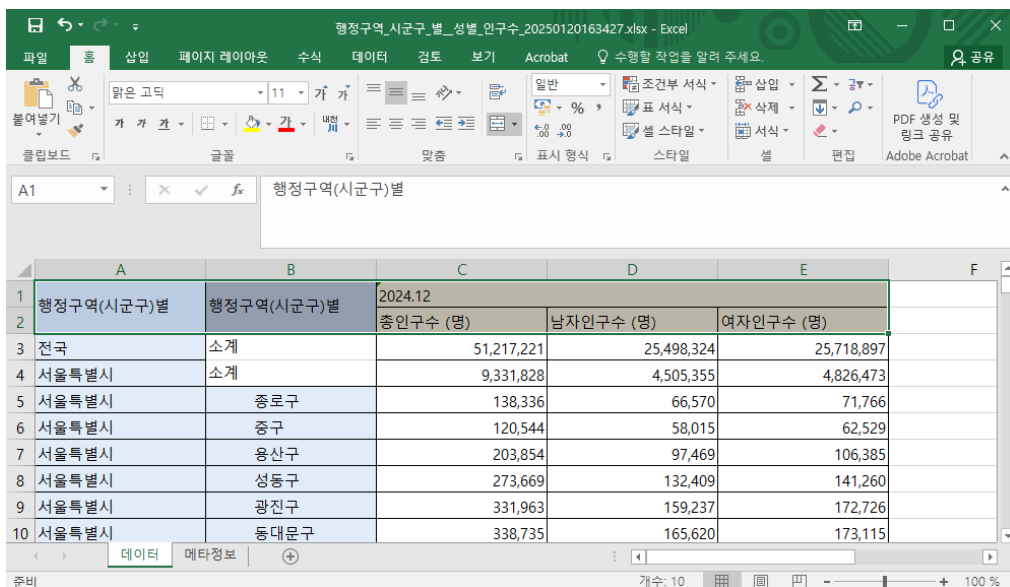
	A	B	C	D	E	F		
4	서울특별시	소계	9,331,828	4,505,355	4,826,473			
5	서울특별시	종로구	138,336	66,570	71,766			
6	서울특별시	중구	120,544	58,015	62,529			
7	서울특별시	용산구	285	경상남도	창녕군	56,085	28,112	27,973
8	서울특별시	성동구	286	경상남도	고성군	48,005	24,312	23,693
9	서울특별시	광진구	287	경상남도	남해군	39,832	19,394	20,438
10	서울특별시	동대문구	288	경상남도	하동군	40,765	20,402	20,363
11	서울특별시	중랑구	289	경상남도	산청군	33,259	16,475	16,784
12	서울특별시	성북구	290	경상남도	함양군	36,131	17,616	18,515
13	서울특별시	강북구	291	경상남도	거창군	59,588	29,322	30,266
14	서울특별시	도봉구	292	경상남도	합천군	40,225	19,764	20,461
15	서울특별시	노원구	293	제주특별자치도	소계	670,368	335,255	335,113
16	서울특별시	은평구	294	제주특별자치도	제주시	488,348	243,675	244,673
			295	제주특별자치도	서귀포시	182,020	91,580	90,440
			296					

# 01. [주소 데이터 분석 + 지오맵] 지리 정보 분석 후 맵 생성하기

## ■ 데이터 준비 및 탐색

### ■ 행정구역 주소 체계 데이터 준비하기

6. 1번째 행을 삭제하고 열 이름을 '행정구역(시군구)별(1)'과 '행정구역(시군구)별(2)'로 변경



	A	B	C	D	E	F
1	행정구역(시군구)별	행정구역(시군구)별	2024.12			
2			총인구수 (명)	남자인구수 (명)	여자인구수 (명)	
3	전국	소계	51,217,221	25,498,324	25,718,897	
4	서울특별시	소계	9,331,828	4,505,355	4,826,473	
5	서울특별시	종로구	138,336	66,570	71,766	
6	서울특별시	중구	120,544	58,015	62,529	
7	서울특별시	용산구	203,854	97,469	106,385	
8	서울특별시	성동구	273,669	132,409	141,260	
9	서울특별시	광진구	331,963	159,237	172,726	
10	서울특별시	동대문구	338,735	165,620	173,115	

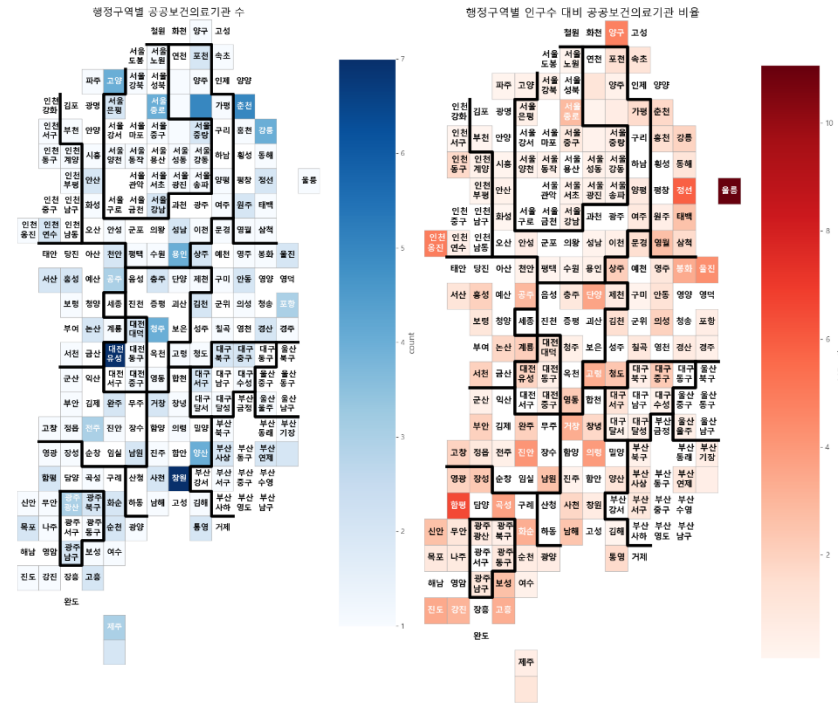
	A	B	C	D	E	
1	행정구역(시군구)별(1)	행정구역(시군구)별(2)	총인구수 (명)	남자인구수 (명)	여자인구수 (명)	
2	전국	소계	51,217,221	25,498,324	25,718,897	
3	서울특별시	소계	9,331,828	4,505,355	4,826,473	
4	서울특별시	종로구	138,336	66,570	71,766	
5	서울특별시	중구	120,544	58,015	62,529	

7. 작업이 끝나면 파일을 '행정구역\_시군구\_별\_성별\_인구수\_2.xlsx'로 저장

## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 분석 미리보기

- 목표 : 행정구역별 공공보건의료기관 수, 인구수 대비 공공보건의료기관 비율
- 핵심개념 : 블록맵
- 데이터 수집
  - 공공보건의료기관현황.csv (공공데이터포털)
  - 행정구역\_시구군\_별\_성별\_인구수\_2.xlsx
- 데이터 준비 및 탐색
  - 행정구역 이름으로 주소 수정
  - 행정구역별 공공보건의료기관 수 집계
  - 행정구역별 인구수 데이터 정리
  - 테이블에 필요한 컬럼 추출 후 테이블 병합



## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 목표설정

1. 행정구역별로 공공보건의료기관 수를 파악
2. 행정구역 별로 인구수 대비 공공보건의료기관 비율을 비교 분석

### ■ 핵심 개념 이해

#### ■ 블록맵

- 구역의 경계선을 단순화한 뒤 블록 형태로 그려서 지도를 나타내는 시각화 기법
- 행정구역별 데이터 크기를 시각화할 때 많이 사용

### ■ 데이터 수집

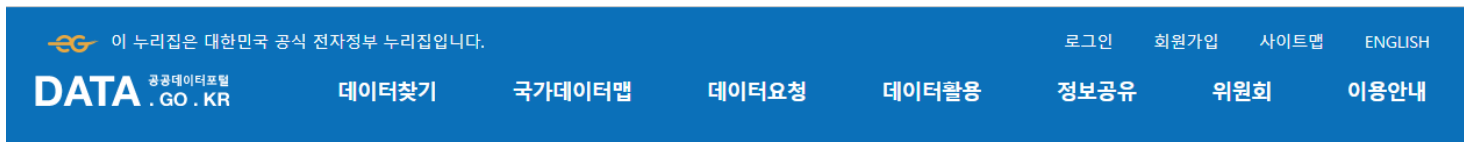
- 전국 공공보건의료기관 현황 데이터
  - 행정구역별 공공보건의료기관 수를 파악, 인구수 대비 공공보건의료기관 비율을 비교 분석
  - 공공데이터포털 사이트에서 다운로드
- 행정구역별 인구수 데이터
  - 행정구역별 인구수 대비 공공보건의료기관 비율 비교 분석
  - 앞의 '행정구역\_시군구\_별\_성별\_인구수\_2.xlsx' 파일을 사용

## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 데이터 수집

#### ■ 전국 공공보건의료기관 현황 데이터 수집하기

1. 공공데이터포털 사이트(www.data.go.kr)에서 '공공보건 의료기관 현황'으로 검색



#### 데이터목록

공공보건의료기관현황 AND 🔍

연관

제공기관별 검색

상세검색

보건의료

국가행정기관

[미리보기](#)

CSV JSON + XML 보건복지부\_전국 지역보건의료기관 현황

전국 지역보건의료기관 현황으로 시도, 시군구, 보건기관 유형, 상위 보건기관명, 보건기관명, 주소, 읍면동명, 도서지역여부, 대표전화번호 등에 대한 정보를 제공합니다.

제공기관 보건복지부 수정일 2024-05-21 조회수 21834 다운로드 8195 주기성 데이터 2 키워드 지역보건의료기관명,주소,대표전화번호

다운로드



## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 데이터 수집

- 전국 공공보건의료기관 현황 데이터 수집하기
- 2. 파일데이터 상세 페이지가 나타나면 버튼을 클릭

데이터 상세



#### 보건복지부\_전국 지역보건의료기관 현황

전국 지역보건의료기관 현황으로 시도, 시군구, 보건기관 유형, 상위 보건기관명, 보건기관명, 주소, 읍면동명, 도서지역여부, 대표전화번호 등에 대한 정보를 제공합니다.



파일데이터

오픈API

추천데이터

공공데이터활용지원센터는 공공데이터포털에 개방되는 3단계 이상의 오픈 포맷 파일데이터를 오픈 API(RestAPI 기반의 JSON/XML)로 자동변환하여 제공합니다. 오픈 API를 활용하기 위해서는 공공데이터포털 회원 가입 및 활용신청이 필요하며, 활용 관련 문의는 공공데이터활용지원센터로 연락주시기 바라며, 데이터 자체에 대한 문의는 아래 제공기관의 관리부서 전화번호로 연락주시기 바랍니다. 파일데이터는 로그인 없이 다운로드를 통해 이용하실 수 있습니다.

CSV 보건복지부\_전국 지역보건의료기관 현황

다운로드

- 2. 다운로드한 파일은 파일명을 '공공보건의료기관현황.csv'로 바꾼 뒤 작업 폴더에 저장

## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 데이터 준비 및 탐색

#### ■ 공공보건의료기관 현황 데이터 준비하기

##### 1. 데이터 파일 확인하기

In [1]:

```
import pandas as pd
pd.set_option('mode.chained_assignment', None) ##원본 데이터프레임이 의도치 않게 수정되는 것을 방지

import numpy as np

data = pd.read_csv('공공보건의료기관현황.csv', index_col = 0,
                  encoding = ' CP949', engine = 'python')
data.head() #작업 확인용 출력
```

Out[1]:

	시군구	기관유형	상위 보건기관명	보건기관명	주소	읍면동명	도서지역 여부	대표 전화번호	
시도									
서울특별시	종로구	보건소	종로구보건소	종로구보건소	서울특별시 종로구 자하문로19길 36 (옥인동, 종로구보건소, 청운효자동자치회관) ...	청운효자동	아니오	02-2148-3514	
서울특별시	중구	보건소	서울중구보건소	서울중구보건소	서울특별시 중구 다산로39길 16 (무학동, 중구보건소)	신당동	아니오	02-3396-6302	
서울특별시	중구	일반보건지소	서울중구보건소	약수보건지소	서울특별시 중구 다산로 92 (신당동, 약수동주민센터)	약수동주민센터	신당동	아니오	02-3396-6964
서울특별시	중구	일반보건지소	서울중구보건소	황학보건지소	서울특별시 중구 난계로11길 52 (황학동, 황학동주민센터)	황학동주민센터	아니오	02-3396-6989	
서울특별시	중구	일반보건지소	서울중구보건소	다산보건지소	서울특별시 중구 동호로15길 50 (신당동, 동사무소여린이집)	신당동	아니오	02-3396-6959	

In [1]: '공공보건의료기관현황.csv' 파일을 data 객체로 로드하고, 상위 다섯 개 행의 데이터를 출력하여 `data.head()` 확인

## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 데이터 준비 및 탐색

- 공공보건의료기관 현황 데이터 준비하기

- 2. 주소 정리하기

In [2]:	<pre># 주소에서 시도, 군구 정보 분리 addr = pd.DataFrame(data['주소'].apply(lambda v: v.split()[:2]).tolist(), columns = ('시도', '군구')) addr.head() #작업 확인용 출력</pre>																		
Out[2]:	<table><tr><th></th><th>시도</th><th>군구</th></tr><tr><td>0</td><td>서울특별시</td><td>종로구</td></tr><tr><td>1</td><td>서울특별시</td><td>중구</td></tr><tr><td>2</td><td>서울특별시</td><td>중구</td></tr><tr><td>3</td><td>서울특별시</td><td>중구</td></tr><tr><td>4</td><td>서울특별시</td><td>중구</td></tr></table>		시도	군구	0	서울특별시	종로구	1	서울특별시	중구	2	서울특별시	중구	3	서울특별시	중구	4	서울특별시	중구
	시도	군구																	
0	서울특별시	종로구																	
1	서울특별시	중구																	
2	서울특별시	중구																	
3	서울특별시	중구																	
4	서울특별시	중구																	

In [2]: data 객체에서 ['주소'] 컬럼의 값을 띄어쓰기를 기준으로 분리하여 split( ), 시군과 군구 정보에 해당하는 0~1번 컬럼[:2]을 추출해서 컬럼 이름을 '시도', '군구'로 나타내고 columns=(' 시도', '군구'), 데이터프레임 객체인 addr을 생성  
생성된 addr 객체의 내용을 출력하여 addr.head( ) 확인

## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 데이터 준비 및 탐색

- 공공보건의료기관 현황 데이터 준비하기

3. 시도 이름에서 잘못된 내용이 있는지 확인

In [3]:	addr['시도'].unique()
Out[3]:	array(['서울특별시', '충청북도', '충청남도', '전라북도', '전라남도', '경상북도', '경상남도', '제주특별자치도', '세종특별자치시', '부산광역시', '대구광역시', '인천광역시', '광주광역시', '대전광역시', '울산광역시', '경기도', '강원도', '강원특별자치도'], dtype=object)

In [3]: addr 객체의 ['시도'] 컬럼 값에서 고유값을 확인 `addr['시도'].unique()`.

## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 데이터 준비 및 탐색

- 공공보건의료기관 현황 데이터 준비하기

- 8. ['군구'] 컬럼에서 정리할 사항이 있는지 탐색

In [16]:	addr['군구'].unique()
Out[16]:	array(['춘천시', '삼척시', '영월군', '원주시', '강릉시', '속초시', '정선군', '수원시', '이천시', '안성시', '의정부시', '포천시', '파주시', '용인시', '평택시', '시흥시', '여주시', '남양주시', '동두천시', '안산시', '부천시', '통영시', '사천시', '창원시', '김해시', '양산시', '거창군', '남해군', '의령군', '포항시', '김천시', '안동시', '울진군', '경주시', '구미시', '영주시', '상주시', '문경시', '경산시', '의성군', '청도군', '고령군', '칠곡군', '봉화군', '울릉군', '부평구', '북구', '순천시', '대덕구', '태백시', '동해시', '화성시', '광산구', '남구', '중구', '아란13길', '서구', '전주시', '진주시', '청주시', '종로구', '성남시', '동구', '화순군', '강동구', '사상구', '달서구', '해운대구', '유성구', '가평군', '양주시', '고양시', '홍천군', '양구군', '청원군', '계룡시', '논산시', '함평군', '양평군', '수성구', '달성군', '연수구', '노원구', '기장군', '공주시', '강북구', '광진구', '나주시', '창녕군', '목포시', '고흥군', '연제구', '동매로', '서초구', '은평구', '중랑구', '강남구', '동작구', '동대문구', '양천구', '성동구', '송파구', '울주군', '계양구', '옹진군', '보성군', '광양시', '영광군', '무안군', '진도군', '강진군', '곡성군', '여수시', '신안군', '장성군', '완주군', '부안군', '정읍시', '남원시', '군산시', '고창군', '진안군', '제주시', '서귀포시', '천안시', '보령시', '서산시', '서천군', '홍성군', '제천시', '충주시', '영동군', '단양군'], dtype=object)

In [16]: addr 객체의 ['군구'] 컬럼의 고유값을 확인

## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 데이터 준비 및 탐색

- 공공보건의료기관 현황 데이터 준비하기
9. 행정구역별 공공보건의료기관의 수 구하기

In [19]:	<pre>addr['시도군구'] = addr.apply(lambda r: r['시도'] + ' ' + r['군구'], axis = 1) addr.head()  #작업 확인용 출력</pre>																								
Out[19]:	<table><tr><th></th><th>시도</th><th>군구</th><th>시도군구</th></tr><tr><td>0</td><td>서울특별시</td><td>종로구</td><td>서울특별시 종로구</td></tr><tr><td>1</td><td>서울특별시</td><td>중구</td><td>서울특별시 중구</td></tr><tr><td>2</td><td>서울특별시</td><td>중구</td><td>서울특별시 중구</td></tr><tr><td>3</td><td>서울특별시</td><td>중구</td><td>서울특별시 중구</td></tr><tr><td>4</td><td>서울특별시</td><td>중구</td><td>서울특별시 중구</td></tr></table>		시도	군구	시도군구	0	서울특별시	종로구	서울특별시 종로구	1	서울특별시	중구	서울특별시 중구	2	서울특별시	중구	서울특별시 중구	3	서울특별시	중구	서울특별시 중구	4	서울특별시	중구	서울특별시 중구
	시도	군구	시도군구																						
0	서울특별시	종로구	서울특별시 종로구																						
1	서울특별시	중구	서울특별시 중구																						
2	서울특별시	중구	서울특별시 중구																						
3	서울특별시	중구	서울특별시 중구																						
4	서울특별시	중구	서울특별시 중구																						

In [19]: [' 시도']와 ['군구'] 컬럼 값을 연결하여 만든 값으로 addr 객체에 새로운 [' 시도군구'] 컬럼을 추가

## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 데이터 준비 및 탐색

- 공공보건의료기관 현황 데이터 준비하기
9. 행정구역별 공공보건의료기관의 수 구하기

In [20]:	<pre>addr['count'] = 0 addr.head() #작업 확인용 출력</pre>																														
Out[20]:	<table><tr><th></th><th>시도</th><th>군구</th><th>시도군구</th><th>count</th></tr><tr><td>0</td><td>서울특별시</td><td>종로구</td><td>서울특별시 종로구</td><td>0</td></tr><tr><td>1</td><td>서울특별시</td><td>중구</td><td>서울특별시 중구</td><td>0</td></tr><tr><td>2</td><td>서울특별시</td><td>중구</td><td>서울특별시 중구</td><td>0</td></tr><tr><td>3</td><td>서울특별시</td><td>중구</td><td>서울특별시 중구</td><td>0</td></tr><tr><td>4</td><td>서울특별시</td><td>중구</td><td>서울특별시 중구</td><td>0</td></tr></table>		시도	군구	시도군구	count	0	서울특별시	종로구	서울특별시 종로구	0	1	서울특별시	중구	서울특별시 중구	0	2	서울특별시	중구	서울특별시 중구	0	3	서울특별시	중구	서울특별시 중구	0	4	서울특별시	중구	서울특별시 중구	0
	시도	군구	시도군구	count																											
0	서울특별시	종로구	서울특별시 종로구	0																											
1	서울특별시	중구	서울특별시 중구	0																											
2	서울특별시	중구	서울특별시 중구	0																											
3	서울특별시	중구	서울특별시 중구	0																											
4	서울특별시	중구	서울특별시 중구	0																											

In [20]: addr 객체에 ['count'] 컬럼을 추가

## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 데이터 준비 및 탐색

- 공공보건의료기관 현황 데이터 준비하기
- 9. 행정구역별 공공보건의료기관의 수 구하기

In [21]:	<pre>addr_group = pd.DataFrame( addr.groupby(['시도', '군구', '시도군구'], as_index = False).count()) addr_group.head() #작업 확인용 출력</pre>																														
Out[21]:	<table><tr><th></th><th>시도</th><th>군구</th><th>시도군구</th><th>count</th></tr><tr><td>0</td><td>강원도</td><td>강릉시</td><td>강원도 강릉시</td><td>19</td></tr><tr><td>1</td><td>강원도</td><td>동해시</td><td>강원도 동해시</td><td>3</td></tr><tr><td>2</td><td>강원도</td><td>삼척시</td><td>강원도 삼척시</td><td>15</td></tr><tr><td>3</td><td>강원도</td><td>양구군</td><td>강원도 양구군</td><td>8</td></tr><tr><td>4</td><td>강원도</td><td>양양군</td><td>강원도 양양군</td><td>11</td></tr></table>		시도	군구	시도군구	count	0	강원도	강릉시	강원도 강릉시	19	1	강원도	동해시	강원도 동해시	3	2	강원도	삼척시	강원도 삼척시	15	3	강원도	양구군	강원도 양구군	8	4	강원도	양양군	강원도 양양군	11
	시도	군구	시도군구	count																											
0	강원도	강릉시	강원도 강릉시	19																											
1	강원도	동해시	강원도 동해시	3																											
2	강원도	삼척시	강원도 삼척시	15																											
3	강원도	양구군	강원도 양구군	8																											
4	강원도	양양군	강원도 양양군	11																											

In [21]: ['시도'], ['군구'], ['시도군구'] 컬럼을 기준으로 그룹을 만듦 `addr.groupby(['시도', '군구', '시도군구'], as_index = False)`.  
그룹별 원소의 개수를 구하여 `count()` ['count'] 컬럼에 저장



## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 데이터 준비 및 탐색

- 공공보건의료기관 현황 데이터 준비하기

#### 10. 데이터 병합에 사용할 인덱스를 설정

In [22]:

addr\_group = addr\_group.set\_index("시도군구")  
addr\_group.head() #작업 확인용 출력

Out[22]:

	시도	군구	count
시도군구			
강원도 강릉시	강원도	강릉시	19
강원도 동해시	강원도	동해시	3
강원도 삼척시	강원도	삼척시	15
강원도 양구군	강원도	양구군	8
강원도 양양군	강원도	양양군	11

In [22]: ['시도군구'] 컬럼을 데이터프레임 병합에 사용할 인덱스로 설정

## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 데이터 준비 및 탐색

#### ■ 행정구역별 인구수 데이터 준비하기

##### 1. 데이터 정리하기

In [23]:	<pre>!pip install openpyxl # 엑셀 파일 불러올 때, 에러시 설치 필요 population = pd.read_excel('행정구역_시군구_별_성별_인구수_2.xlsx') population.head() #작업 확인용 출력</pre>																																								
Out[23]:	<table><thead><tr><th></th><th>행정구역(시군구)별(1)</th><th>행정구역(시군구)별(2)</th><th>총인구수 (명)</th><th>남자인구수 (명)</th><th>여자인구수 (명)</th></tr></thead><tbody><tr><td>0</td><td>전국</td><td>소계</td><td>51175725</td><td>25474633</td><td>25701092</td></tr><tr><td>1</td><td>서울특별시</td><td>소계</td><td>9331860</td><td>4501804</td><td>4830056</td></tr><tr><td>2</td><td>서울특별시</td><td>종로구</td><td>138096</td><td>66413</td><td>71683</td></tr><tr><td>3</td><td>서울특별시</td><td>중구</td><td>119601</td><td>57556</td><td>62045</td></tr><tr><td>4</td><td>서울특별시</td><td>용산구</td><td>203350</td><td>97097</td><td>106253</td></tr></tbody></table>						행정구역(시군구)별(1)	행정구역(시군구)별(2)	총인구수 (명)	남자인구수 (명)	여자인구수 (명)	0	전국	소계	51175725	25474633	25701092	1	서울특별시	소계	9331860	4501804	4830056	2	서울특별시	종로구	138096	66413	71683	3	서울특별시	중구	119601	57556	62045	4	서울특별시	용산구	203350	97097	106253
	행정구역(시군구)별(1)	행정구역(시군구)별(2)	총인구수 (명)	남자인구수 (명)	여자인구수 (명)																																				
0	전국	소계	51175725	25474633	25701092																																				
1	서울특별시	소계	9331860	4501804	4830056																																				
2	서울특별시	종로구	138096	66413	71683																																				
3	서울특별시	중구	119601	57556	62045																																				
4	서울특별시	용산구	203350	97097	106253																																				
In [24]:	<pre>population = population.rename(columns = {'행정구역(시군구)별(1)': '시도', '행정구역(시군구)별(2)': '군구'}) population.head() #작업 확인용 출력</pre>																																								
Out[24]:	<table><thead><tr><th></th><th>시도</th><th>군구</th><th>총인구수 (명)</th><th>남자인구수 (명)</th><th>여자인구수 (명)</th></tr></thead><tbody><tr><td>0</td><td>전국</td><td>소계</td><td>51175725</td><td>25474633</td><td>25701092</td></tr><tr><td>1</td><td>서울특별시</td><td>소계</td><td>9331860</td><td>4501804</td><td>4830056</td></tr><tr><td>2</td><td>서울특별시</td><td>종로구</td><td>138096</td><td>66413</td><td>71683</td></tr><tr><td>3</td><td>서울특별시</td><td>중구</td><td>119601</td><td>57556</td><td>62045</td></tr><tr><td>4</td><td>서울특별시</td><td>용산구</td><td>203350</td><td>97097</td><td>106253</td></tr></tbody></table>						시도	군구	총인구수 (명)	남자인구수 (명)	여자인구수 (명)	0	전국	소계	51175725	25474633	25701092	1	서울특별시	소계	9331860	4501804	4830056	2	서울특별시	종로구	138096	66413	71683	3	서울특별시	중구	119601	57556	62045	4	서울특별시	용산구	203350	97097	106253
	시도	군구	총인구수 (명)	남자인구수 (명)	여자인구수 (명)																																				
0	전국	소계	51175725	25474633	25701092																																				
1	서울특별시	소계	9331860	4501804	4830056																																				
2	서울특별시	종로구	138096	66413	71683																																				
3	서울특별시	중구	119601	57556	62045																																				
4	서울특별시	용산구	203350	97097	106253																																				

In [23]: '행정구역\_시군구\_별\_성별\_인구수\_2.xlsx' 파일을 population 객체로 로드하고, 출력하여 확인

In [24]: rename() 함수를 사용하여 컬럼 이름을 변경

## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 데이터 준비 및 탐색

#### ■ 행정구역별 인구수 데이터 준비하기

2. ['군구'] 컬럼에 포함되어 있는 왼쪽 띄어쓰기 공백을 제거

['시도군구'] 컬럼을 만들고 addr\_group과 병합하기 위해 인덱스로 설정

In [25]:	population['군구'] = population['군구'].str.strip() # pandas 메서드 바뀜																																										
In [26]:	population['시도군구'] = population.apply(lambda r: r['시도'] + ' ' + r['군구'], axis = 1) population.head()    #작업 확인용 출력																																										
Out[26]:	<table><tr><th></th><th>시도</th><th>군구</th><th>총인구수 (명)</th><th>남자인구수 (명)</th><th>여자인구수 (명)</th><th>시도군구</th></tr><tr><td>0</td><td>전국</td><td>소계</td><td>51217221</td><td>25498324</td><td>25718897</td><td>전국 소계</td></tr><tr><td>1</td><td>서울특별시</td><td>소계</td><td>9331828</td><td>4505355</td><td>4826473</td><td>서울특별시 소계</td></tr><tr><td>2</td><td>서울특별시</td><td>종로구</td><td>138336</td><td>66570</td><td>71766</td><td>서울특별시 종로구</td></tr><tr><td>3</td><td>서울특별시</td><td>중구</td><td>120544</td><td>58015</td><td>62529</td><td>서울특별시 중구</td></tr><tr><td>4</td><td>서울특별시</td><td>용산구</td><td>203854</td><td>97469</td><td>106385</td><td>서울특별시 용산구</td></tr></table>		시도	군구	총인구수 (명)	남자인구수 (명)	여자인구수 (명)	시도군구	0	전국	소계	51217221	25498324	25718897	전국 소계	1	서울특별시	소계	9331828	4505355	4826473	서울특별시 소계	2	서울특별시	종로구	138336	66570	71766	서울특별시 종로구	3	서울특별시	중구	120544	58015	62529	서울특별시 중구	4	서울특별시	용산구	203854	97469	106385	서울특별시 용산구
	시도	군구	총인구수 (명)	남자인구수 (명)	여자인구수 (명)	시도군구																																					
0	전국	소계	51217221	25498324	25718897	전국 소계																																					
1	서울특별시	소계	9331828	4505355	4826473	서울특별시 소계																																					
2	서울특별시	종로구	138336	66570	71766	서울특별시 종로구																																					
3	서울특별시	중구	120544	58015	62529	서울특별시 중구																																					
4	서울특별시	용산구	203854	97469	106385	서울특별시 용산구																																					

In [25]: ['군구'] 컬럼의 문자열 앞뒤에 포함된 띄어쓰기 공백을 모두 제거 `strip()`

In [26]: ['시도']와 ['군구'] 컬럼 값을 연결하여 새로운 ['시도군구'] 컬럼에 추가

## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 데이터 준비 및 탐색

- 행정구역별 인구수 데이터 준비하기

2. ['군구'] 컬럼에 포함되어 있는 왼쪽 띄어쓰기 공백을 제거

['시도군구'] 컬럼을 만들고 addr\_group과 병합하기 위해 인덱스로 설정

In [27]:	population = population[population.군구 != '합계'] population = population[population.군구 != '소계']																																										
In [28]:	population = population.set_index("시도군구") population.head() #작업 확인용 출력																																										
Out[28]:	<table><tr><th></th><th>시도</th><th>군구</th><th>총인구수 (명)</th><th>남자인구수 (명)</th><th>여자인구수 (명)</th></tr><tr><td colspan="6">시도군구</td></tr><tr><td>서울특별시 종로구</td><td>서울특별시</td><td>종로구</td><td>138336</td><td>66570</td><td>71766</td></tr><tr><td>서울특별시 중구</td><td>서울특별시</td><td>중구</td><td>120544</td><td>58015</td><td>62529</td></tr><tr><td>서울특별시 용산구</td><td>서울특별시</td><td>용산구</td><td>203854</td><td>97469</td><td>106385</td></tr><tr><td>서울특별시 성동구</td><td>서울특별시</td><td>성동구</td><td>273669</td><td>132409</td><td>141260</td></tr><tr><td>서울특별시 광진구</td><td>서울특별시</td><td>광진구</td><td>331963</td><td>159237</td><td>172726</td></tr></table>		시도	군구	총인구수 (명)	남자인구수 (명)	여자인구수 (명)	시도군구						서울특별시 종로구	서울특별시	종로구	138336	66570	71766	서울특별시 중구	서울특별시	중구	120544	58015	62529	서울특별시 용산구	서울특별시	용산구	203854	97469	106385	서울특별시 성동구	서울특별시	성동구	273669	132409	141260	서울특별시 광진구	서울특별시	광진구	331963	159237	172726
	시도	군구	총인구수 (명)	남자인구수 (명)	여자인구수 (명)																																						
시도군구																																											
서울특별시 종로구	서울특별시	종로구	138336	66570	71766																																						
서울특별시 중구	서울특별시	중구	120544	58015	62529																																						
서울특별시 용산구	서울특별시	용산구	203854	97469	106385																																						
서울특별시 성동구	서울특별시	성동구	273669	132409	141260																																						
서울특별시 광진구	서울특별시	광진구	331963	159237	172726																																						

In [27]: ['군구'] 컬럼 값이 '합계', '소계' 행은 필요 없으므로 제외

In [28]: ['시도군구'] 컬럼을 데이터프레임 병합에 사용할 인덱스로 설정

## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 데이터 준비 및 탐색

#### ■ 행정구역별 인구수 데이터 준비하기

#### 3. addr\_group과 population을 인덱스 기준으로 병합

- 필요한 데이터를 하나의 데이터프레임으로 정리할 수 있음

In [29]:

addr\_population\_merge = pd.merge(addr\_group,population, how = 'inner', left\_index = True, right\_index = True)  
addr\_population\_merge.head() #작업 확인용 출력

Out[29]:

시도_x 군구_x			count	시도_y 군구_y			총인구수 (명)	남자인구수 (명)	여자인구수 (명)
강원특별자치도 고성군	강원특별자치도	고성군	14	강원특별자치도	고성군		26999	13841	13158
강원특별자치도 동해시	강원특별자치도	동해시	1	강원특별자치도	동해시		87675	44495	43180
강원특별자치도 속초시	강원특별자치도	속초시	1	강원특별자치도	속초시		80933	39931	41002
강원특별자치도 영월군	강원특별자치도	영월군	14	강원특별자치도	영월군		36721	18990	17731
강원특별자치도 철원군	강원특별자치도	철원군	13	강원특별자치도	철원군		40497	20922	19575

In [29]: addr\_group과 population을 내부병합으로 병합

## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 데이터 준비 및 탐색

#### ■ 행정구역별 인구수 데이터 준비하기

#### 3. addr\_group과 population을 인덱스 기준으로 병합

- 필요한 데이터를 하나의 데이터프레임으로 정리할 수 있음

In [30]:	<pre>local_MC_Population = addr_population_merge[['시도_x', '군구_x', 'count', '총인구수 (명)']] local_MC_Population.head() #작업 확인용 출력</pre>																																										
Out[30]:	<table><tr><th colspan="2"></th><th>시도_x</th><th>군구_x</th><th>count</th><th>총인구수 (명)</th></tr><tr><th colspan="2">시도군구</th><th></th><th></th><th></th><th></th></tr><tr><td>강원특별자치도</td><td>고성군</td><td>강원특별자치도</td><td>고성군</td><td>14</td><td>26999</td></tr><tr><td>강원특별자치도</td><td>동해시</td><td>강원특별자치도</td><td>동해시</td><td>1</td><td>87675</td></tr><tr><td>강원특별자치도</td><td>속초시</td><td>강원특별자치도</td><td>속초시</td><td>1</td><td>80933</td></tr><tr><td>강원특별자치도</td><td>영월군</td><td>강원특별자치도</td><td>영월군</td><td>14</td><td>36721</td></tr><tr><td>강원특별자치도</td><td>철원군</td><td>강원특별자치도</td><td>철원군</td><td>13</td><td>40497</td></tr></table>			시도_x	군구_x	count	총인구수 (명)	시도군구						강원특별자치도	고성군	강원특별자치도	고성군	14	26999	강원특별자치도	동해시	강원특별자치도	동해시	1	87675	강원특별자치도	속초시	강원특별자치도	속초시	1	80933	강원특별자치도	영월군	강원특별자치도	영월군	14	36721	강원특별자치도	철원군	강원특별자치도	철원군	13	40497
		시도_x	군구_x	count	총인구수 (명)																																						
시도군구																																											
강원특별자치도	고성군	강원특별자치도	고성군	14	26999																																						
강원특별자치도	동해시	강원특별자치도	동해시	1	87675																																						
강원특별자치도	속초시	강원특별자치도	속초시	1	80933																																						
강원특별자치도	영월군	강원특별자치도	영월군	14	36721																																						
강원특별자치도	철원군	강원특별자치도	철원군	13	40497																																						

In [30]: 필요한 컬럼 4개만 추출하여 local\_MC\_Population 객체를 생성

## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 데이터 준비 및 탐색

#### ■ 행정구역별 인구수 데이터 준비하기

#### 3. addr\_group과 population을 인덱스 기준으로 병합

- 필요한 데이터를 하나의 데이터프레임으로 정리할 수 있음

In [31]:

```
#컬럼 이름 변경
local_MC_Population = local_MC_Population.rename(columns = {'시도_x': '시도', '군구_x': '군구','총인구수 (명)': '인구수'})
MC_count = local_MC_Population['count']
local_MC_Population['MC_ratio'] = MC_count.div(local_MC_Population['인구수'], axis = 0)*100000
local_MC_Population.head() #작업 확인용 출력
```

Out[31]:

	시도	군구	count	인구수	MC_ratio
시도군구					
강원특별자치도 고성군	강원특별자치도	고성군	14	26999	51.853772
강원특별자치도 동해시	강원특별자치도	동해시	1	87675	1.140576
강원특별자치도 속초시	강원특별자치도	속초시	1	80933	1.235590
강원특별자치도 영월군	강원특별자치도	영월군	14	36721	38.125323
강원특별자치도 철원군	강원특별자치도	철원군	13	40497	32.101143

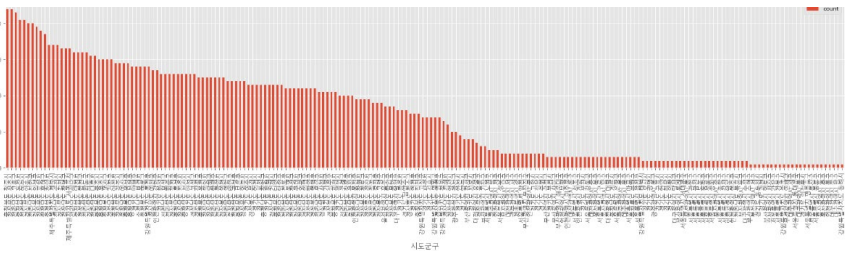
In [31]: 인구수 대비 공공보건의료기관 비율을 구하여 local\_MC\_Population의 ['MC\_ratio'] 컬럼에 추가

## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 분석 모델 구축 및 시각화

#### ■ 바 차트 그리기

##### 1. 행정구역별 공공보건의료기관 수에 대한 바 차트를 그리기

In [32]:	<pre>from matplotlib import pyplot as plt from matplotlib import rcParams, style style.use('ggplot')  from matplotlib import font_manager, rc font_name = font_manager.FontProperties(fname = "c:/Windows/Fonts/malgun.ttf").get_name() rc('font', family = font_name)</pre>
In [33]:	<pre>MC_ratio = local_MC_Population[['count']] MC_ratio = MC_ratio.sort_values('count', ascending = False) plt.rcParams["figure.figsize"] = (25, 5) MC_ratio.plot(kind = 'bar', rot = 90) plt.show()</pre>
Out[33]:	

In [33]: local\_MC\_Population 객체의 ['count'] 컬럼 값을 오름차순으로 정렬하여, 행정 구역별 공공보건의료기관 수에 대한 바 차트를 그림

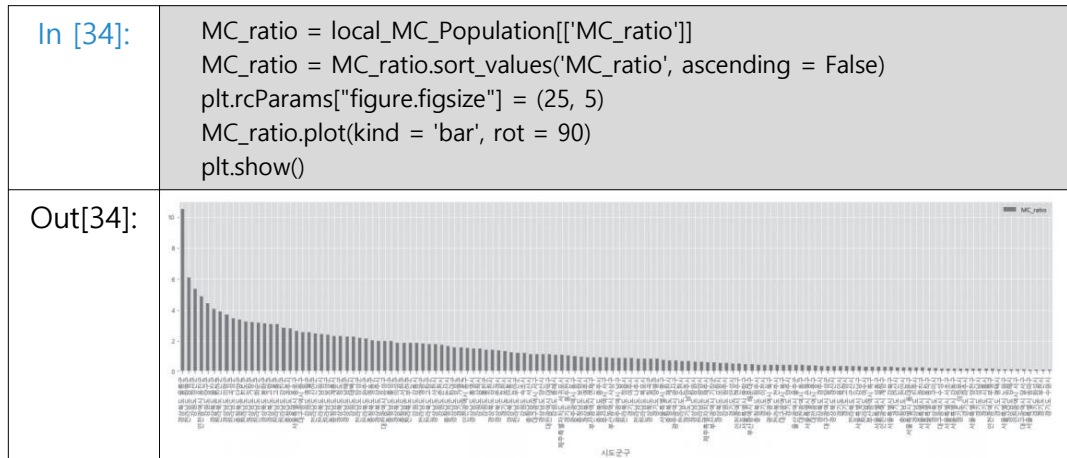


## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 분석 모델 구축 및 시각화

#### ■ 바 차트 그리기

#### 2. 행정구역별로 인구수 대비 공공보건의료기관 비율에 대한 바 차트를 그리기



In [34]: local\_MC\_Population 객체의 ['MC\_ratio'] 컬럼 값을 오름차순으로 정렬하여, 행정구역별로 인구수 대비 공공보건의료기관 비율에 대한 바 차트를 그

## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 분석 모델 구축 및 시각화

#### ■ 블록맵으로 시각화하기

##### 1. 데이터 준비하기

In [35]:	<pre>import os path = os.getcwd()</pre>																																																
In [36]:	<pre>data_draw_korea = pd.read_csv('data_draw_korea.csv',                                 index_col = 0, encoding = 'UTF-8', engine = 'python')  data_draw_korea.head() #작업 확인용 출력</pre>																																																
Out[36]:	<table><tr><th></th><th>인구수</th><th>shortName</th><th>x</th><th>y</th><th>면적</th><th>광역시도</th><th>행정구역</th></tr><tr><td>0</td><td>202520</td><td>강릉</td><td>11</td><td>4</td><td>1040.07</td><td>강원도</td><td>강릉시</td></tr><tr><td>1</td><td>25589</td><td>고성(강원)</td><td>9</td><td>0</td><td>664.19</td><td>강원도</td><td>고성군</td></tr><tr><td>2</td><td>86747</td><td>동해</td><td>11</td><td>5</td><td>180.01</td><td>강원도</td><td>동해시</td></tr><tr><td>3</td><td>63986</td><td>삼척</td><td>11</td><td>8</td><td>1185.80</td><td>강원도</td><td>삼척시</td></tr><tr><td>4</td><td>76733</td><td>속초</td><td>9</td><td>1</td><td>105.25</td><td>강원도</td><td>속초시</td></tr></table>		인구수	shortName	x	y	면적	광역시도	행정구역	0	202520	강릉	11	4	1040.07	강원도	강릉시	1	25589	고성(강원)	9	0	664.19	강원도	고성군	2	86747	동해	11	5	180.01	강원도	동해시	3	63986	삼척	11	8	1185.80	강원도	삼척시	4	76733	속초	9	1	105.25	강원도	속초시
	인구수	shortName	x	y	면적	광역시도	행정구역																																										
0	202520	강릉	11	4	1040.07	강원도	강릉시																																										
1	25589	고성(강원)	9	0	664.19	강원도	고성군																																										
2	86747	동해	11	5	180.01	강원도	동해시																																										
3	63986	삼척	11	8	1185.80	강원도	삼척시																																										
4	76733	속초	9	1	105.25	강원도	속초시																																										

In [35]: 현재 사용 중인 폴더(디렉토리)의 경로를 구하여 path에 저장

In [36]: data\_draw\_korea.csv 파일을 로드하여 data\_draw\_korea 객체에 저장하고, 출력하여 내용을 확인

## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 분석 모델 구축 및 시각화

#### ■ 블록맵으로 시각화하기

##### 2. 행정구역 이름 매핑하기

	<pre>#2023년 6월 이후에 변경된 행정구역명으로 수정하기 addr_aliases = { '강원도':'강원특별자치도', '전라북도':'전북특별자치도' } data_draw_korea['광역시도'] = data_draw_korea['광역시도'].apply(lambda v: addr_aliases.get(v, v)) data_draw_korea.head() #작업 확인용 출력</pre>																																																								
In [37]:	<pre>data_draw_korea['시도군구'] = data_draw_korea.apply(lambda r: r['광역시도'] + ' ' + r['행정구역'], axis = 1)</pre>																																																								
In [38]:	<pre>data_draw_korea = data_draw_korea.set_index("시도군구") data_draw_korea.head() #작업 확인용 출력</pre>																																																								
Out[38]:	<table><tr><th></th><th>인구수</th><th>shortName</th><th>x</th><th>y</th><th>면적</th><th>광역시도</th><th>행정구역</th></tr><tr><th colspan="8">시도군구</th></tr><tr><td>강원특별자치도 강릉시</td><td>202520</td><td>강릉</td><td>11</td><td>4</td><td>1040.07</td><td>강원특별자치도</td><td>강릉시</td></tr><tr><td>강원특별자치도 고성군</td><td>25589</td><td>고성(강원)</td><td>9</td><td>0</td><td>664.19</td><td>강원특별자치도</td><td>고성군</td></tr><tr><td>강원특별자치도 동해시</td><td>86747</td><td>동해</td><td>11</td><td>5</td><td>180.01</td><td>강원특별자치도</td><td>동해시</td></tr><tr><td>강원특별자치도 삼척시</td><td>63986</td><td>삼척</td><td>11</td><td>8</td><td>1185.80</td><td>강원특별자치도</td><td>삼척시</td></tr><tr><td>강원특별자치도 속초시</td><td>76733</td><td>속초</td><td>9</td><td>1</td><td>105.25</td><td>강원특별자치도</td><td>속초시</td></tr></table>		인구수	shortName	x	y	면적	광역시도	행정구역	시도군구								강원특별자치도 강릉시	202520	강릉	11	4	1040.07	강원특별자치도	강릉시	강원특별자치도 고성군	25589	고성(강원)	9	0	664.19	강원특별자치도	고성군	강원특별자치도 동해시	86747	동해	11	5	180.01	강원특별자치도	동해시	강원특별자치도 삼척시	63986	삼척	11	8	1185.80	강원특별자치도	삼척시	강원특별자치도 속초시	76733	속초	9	1	105.25	강원특별자치도	속초시
	인구수	shortName	x	y	면적	광역시도	행정구역																																																		
시도군구																																																									
강원특별자치도 강릉시	202520	강릉	11	4	1040.07	강원특별자치도	강릉시																																																		
강원특별자치도 고성군	25589	고성(강원)	9	0	664.19	강원특별자치도	고성군																																																		
강원특별자치도 동해시	86747	동해	11	5	180.01	강원특별자치도	동해시																																																		
강원특별자치도 삼척시	63986	삼척	11	8	1185.80	강원특별자치도	삼척시																																																		
강원특별자치도 속초시	76733	속초	9	1	105.25	강원특별자치도	속초시																																																		

In [37]: ['광역시도']와 ['행정구역'] 컬럼 값을 연결하여 새로운 ['시도군구'] 컬럼으로 추가

In [38]: ['시도군구'] 컬럼을 데이터프레임 병합에 사용할 인덱스로 설정

## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 분석 모델 구축 및 시각화

- 블록맵으로 시각화하기

#### 2. 행정구역 이름 매핑하기

In [39]:

```
data_draw_korea_MC_Population_all = pd.merge(data_draw_korea,local_MC_Population, how = 'outer',  
left_index = True, right_index = True)
```

```
data_draw_korea_MC_Population_all.head()
```

Out[39]:

	인구수_x	shortName	x	y	면적	광역시도	행정구 역	시도	군구	count	인구수 _y	MC_ratio
시도군구												
강원특별자치도 강릉시	202520.0	강릉	11.0	4.0	1040.07	강원특별자치도	강릉시	NaN	NaN	NaN	NaN	NaN
강원특별자치도 고성군	25589.0	고성(강원)	9.0	0.0	664.19	강원특별자치도	고성군	강원특별자치도	고성군	14.0	26999.0	51.853772
강원특별자치도 동해시	86747.0	동해	11.0	5.0	180.01	강원특별자치도	동해시	강원특별자치도	동해시	1.0	87675.0	1.140576
강원특별자치도 삼척시	63986.0	삼척	11.0	8.0	1185.80	강원특별자치도	삼척시	NaN	NaN	NaN	NaN	NaN
강원특별자치도 속초시	76733.0	속초	9.0	1.0	105.25	강원특별자치도	속초시	강원특별자치도	속초시	1.0	80933.0	1.235590

In [39]: data\_draw\_korea와 local\_MC\_Population을 외부병합으로how='outer' 병합

## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 분석 모델 구축 및 시각화

- 블록맵으로 시각화하기

- 3. 블록맵으로 시각화하기

```
In [40]: BORDER_LINES = [  
    (3, 2), (5, 2), (5, 3), (9, 3), (9, 1)], # 인천  
    (2, 5), (3, 5), (3, 4), (8, 4), (8, 7), (7, 7), (7, 9), (4, 9), (4, 7), (1, 7)], # 서울  
    (1, 6), (1, 9), (3, 9), (3, 10), (8, 10), (8, 9), (9, 9), (9, 8), (10, 8), (10, 5), (9, 5), (9, 3)], # 경기도  
    (9, 12), (9, 10), (8, 10)], # 강원도  
    (10, 5), (11, 5), (11, 4), (12, 4), (12, 5), (13, 5), (11, 8), (11, 9), (10, 9), (10, 8)], # 충청북도  
    (14, 4), (15, 4), (15, 6)], # 대전시  
    (14, 7), (14, 9), (13, 9), (13, 11), (13, 13)], # 경상북도  
    (14, 8), (16, 8), (16, 10), (15, 10), (15, 11), (14, 11), (14, 12), (13, 12)], # 대구시  
    (15, 11), (16, 11), (16, 13)], # 울산시  
    (17, 1), (17, 3), (18, 3), (18, 6), (15, 6)], # 전라북도  
    (19, 2), (19, 4), (21, 4), (21, 3), (22, 3), (22, 2), (19, 2)], #광주시  
    (18, 5), (20, 5), (20, 6)], # 전라남도  
    (16, 9), (18, 9), (18, 8), (19, 8), (19, 9), (20, 9), (20, 10)], #부산시  
]
```

In [40]: 블록맵의 행정구역 경계선을 그리기 위해 행정구역의 블록 위치인 x, y 데이터를 정의

## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 분석 모델 구축 및 시각화

#### ■ 블록맵으로 시각화하기

#### 3. 블록맵으로 시각화하기

In [41]:

```
def draw_blockMap(blockedMap, targetData, title, color):
    whitelabelmin = (max(blockedMap[targetData]) -
                    min(blockedMap[targetData])) * 0.25 +
                    min(blockedMap[targetData])
    datalabel = targetData
    vmin = min(blockedMap[targetData])
    vmax = max(blockedMap[targetData])
    mapdata = blockedMap.pivot(index = 'y', columns = 'x', values = targetData)
    masked_mapdata = np.ma.masked_where(np.isnan(mapdata), mapdata)

    plt.figure(figsize = (8, 13))
    plt.title(title)
    plt.pcolor(masked_mapdata, vmin = vmin, vmax = vmax, cmap =
              color, edgecolor = '#aaaaaa', linewidth = 0.5)
    #지역 이름 표시
    for idx, row in blockedMap.iterrows():
        annocolor = 'white' if row[targetData] > whitelabelmin else
        'black'
        #광역시는 구 이름이 겹치는 경우가 많아서 시단위 이름도 같이 표시
        if row['광역시도'].endswith('시') and not row['광역시도'].
        startswith('세종'):
            dispname = '{}\n{}'.format(row['광역시도'][:2], row['행정구
            역'][:1])
            if len(row['행정구역']) <= 2:
                dispname += row['행정구역'][-1]
            else:
                dispname = row['행정구역'][-1]
        #서대문구, 서귀포시 같이 이름이 3자 이상이면 작은 글자로 표시
```

## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 분석 모델 구축 및 시각화

#### ■ 블록맵으로 시각화하기

#### 3. 블록맵으로 시각화하기

```
In [41]: if len(dispname.splitlines()[-1]) >= 3:
          fontsize, linespacing = 9.5, 1.5
          else:
          fontsize, linespacing = 11, 1.2
          plt.annotate(dispname, (row['x']+0.5, row['y']+0.5), weight = 'bold',
          fontsize = fontsize, ha = 'center', va = 'center', color = annocolor,
          linespacing = linespacing)

          #시도 경계를 그린다.
          for path in BORDER_LINES:
          ys, xs = zip(*path)
          plt.plot(xs, ys, c = 'black', lw = 4)
          plt.gca().invert_yaxis()
          #plt.gca().set_aspect(1)
          plt.axis('off')

          cb = plt.colorbar(shrink = 1, aspect = 10)
          cb.set_label(datalabel)
          plt.tight_layout()
          plt.savefig('..장_data' + 'blockMap_' + targetData + '.png')
          plt.show()
```

In [41]: 블록맵의 블록에 데이터를 매핑하고 색을 표시하여 블록맵을 그린 뒤 저장하는 함수를 정의

## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 분석 모델 구축 및 시각화

- 블록맵으로 시각화하기

#### 4. 행정구역별 공공보건의료기관 수를 블록맵으로 시각화

```
In [42]: draw_blockMap(data_draw_korea_MC_Population_all, 'count', '행정구역  
별 공공보건의료기관 수', 'Blues')
```

In [42]: data\_draw\_korea\_MC\_Population\_all 객체의 ['count'] 컬럼 값에 대해 Blues 색상 스펙트럼을 사용하여 블록맵 작성

#### 5. 인구수 대비 공공보건의료기관 비율을 블록맵으로 시각화

```
In [43]: draw_blockMap(data_draw_korea_MC_Population_all, 'MC_ratio', '행정구  
역별 인구수 대비 공공보건의료기관 비율', 'Reds')
```

In [43]: data\_draw\_korea\_MC\_Population\_all 객체의 ['MC\_ratio'] 컬럼 값에 대해 Reds 색상 스펙트럼을 사용하여 블록맵 작성



## 02. [행정구역별 데이터 분석 + 블록맵] 행정구역별 의료기관 현황 분석하기

### ■ 분석 모델 구축 및 시각화

#### ■ 블록맵으로 시각화하기 - 결과

