

Geopandas를 이용한 공간 데이터 시각화

Jaehwan Jung¹

Aug 19th, 2025

¹Department of Statistics and Data Science, University of Seoul, S. Korea

GeoPandas 기초

공간 데이터

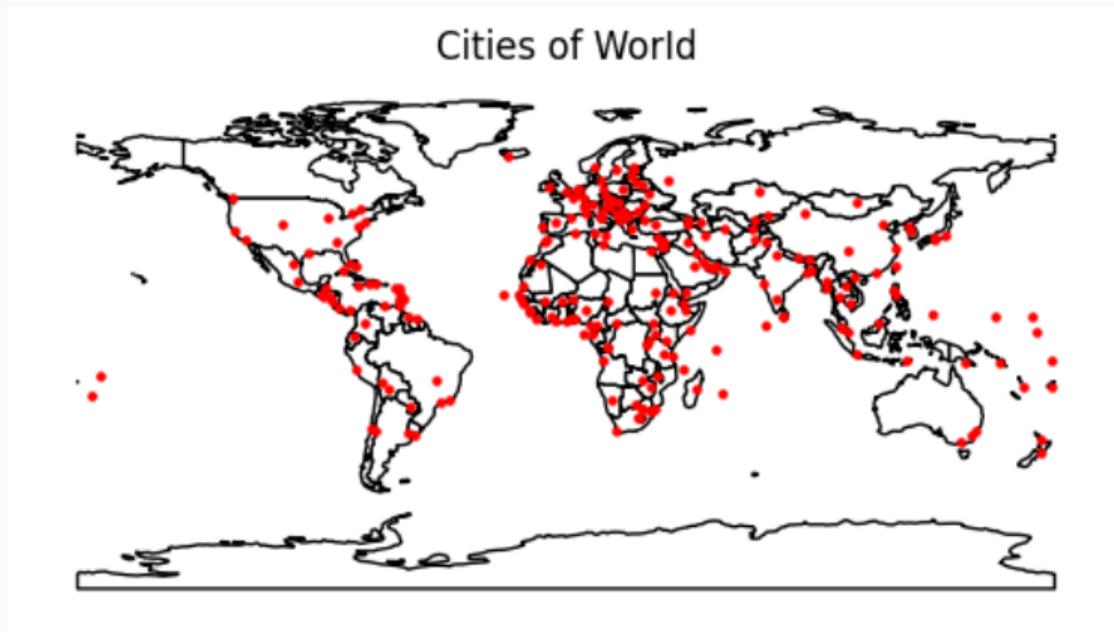
튜토리얼

데이터와 좌표계

GeoPandas 기초

	정류소아이디	모바일아이디	정류소명	영문명	시도	구군	동	경도	위도	경유노선수	경유노선
0	7361000423	10423.0	정리	Jeong-ri	대구광역시	군위군	군위읍	128.560200	36.251510	1	군위3
1	7361000424	10424.0	정리	Jeong-ri	대구광역시	군위군	군위읍	128.560072	36.251480	2	군위2+군위3
2	7361000426	10426.0	정리	Jeong-ri	대구광역시	군위군	군위읍	128.562187	36.246557	2	군위2+군위3
3	7361000428	10428.0	사직1리	sajik1ri	대구광역시	군위군	군위읍	128.556720	36.229607	1	군위4
4	7361000429	10429.0	사직1리	sajik1ri	대구광역시	군위군	군위읍	128.556858	36.229608	1	군위4

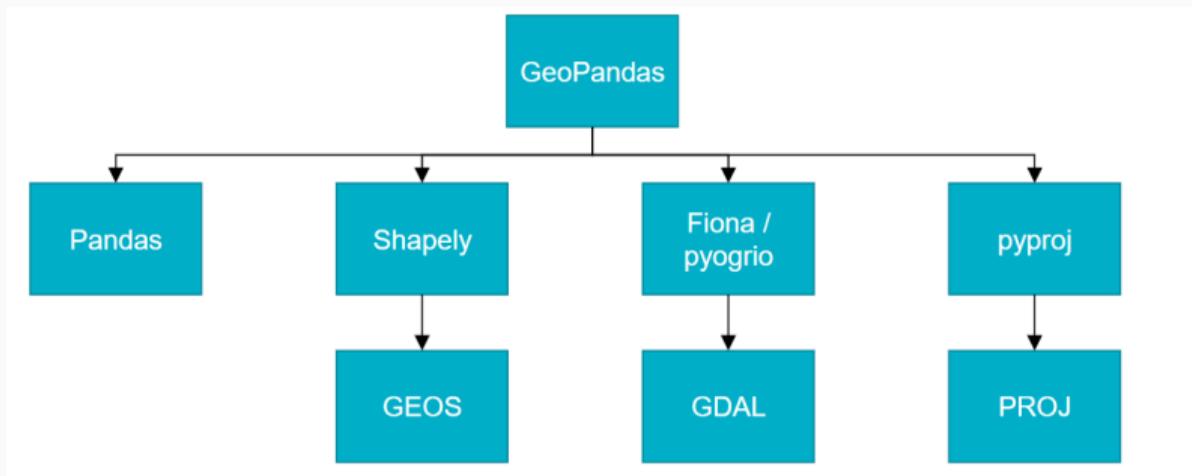
- 기존 데이터와 다른점은?
- 위도(Lat), 경도(Lon) 등 데이터 분석.



- 각 국가의 모양을 그리는 방법은?
- 주요 도시의 point를 찍는 방법은?

Geopandas

- 지리공간 데이터를 Python에서 더 쉽게 다룰 수 있도록 지원.
- Pandas를 확장하여 지리 객체와 공간 연산을 처리 가능하게 함.
- Pandas, GEOS, Shapely, GDAL, PyProj, Rtree 등 여러 라이브러리의 기능을 결합.



- Pandas: GeoPandas의 GeoDataFrame은 Pandas의 DataFrame을 확장한 형태로, 일반 데이터 컬럼 + geometry 컬럼을 함께 다룸.
- Shapely + GEOS: 파이썬에서 점(Point), 선(LineString), 면(Polygon) 등 도형 객체 생성 및 기하 연산 지원함.
- Fiona / pyogrio: 공간 데이터 파일 입출력 라이브러리.
- Pyproj: 좌표계 변환 라이브러리.

공간 데이터

V-WORLD 디지털트윈국토

지도조회 다운로드 오픈API 서비스 활용지원 소통공간 기업정보

브이월드로 만나는 디지털 트윈국토

통합검색

- 지도조회
- 분석 및 시뮬레이션
- 인허가 분석
- 오픈API
- 공간정보 다운로드
- 기업정보

방문자	다운로드	API호출
오늘 26,435 명	오늘 208 건	오늘 1,223,509 건

<https://www.vworld.kr/>

읍면동 행정구역 경계 파일 다운로드

다운로드 공간정보 다운로드

홈 > 다운로드 > 공간정보 다운로드

국토관리 지역개발 > 경계 행정구역_읍면동(법정동)

유형 다운로드 | 기관 국토교통부 | 다운로드 47,933 | 문의 23 | 조회 59,214

· 설명 법정동(읍면동) 경계도면
· 라이선스 CC BY-NC-ND

↓ 컬럼 정의서 다운로드

데이터 리소스

파일명 조회 10건씩

선택 0 건 ↓ 선택다운로드 총 18 건 (2 / 2 page)

<input checked="" type="checkbox"/>	SHP	LSMD_ADM_SECT_UMD_부산.zip	용량 776,040 byte	구분 데이터	기준일 2025-06	갱신일 2025-06-30	↓ 다운로드
<input checked="" type="checkbox"/>	SHP	LSMD_ADM_SECT_UMD_대전.zip	용량 788,566 byte	구분 데이터	기준일 2025-06	갱신일 2025-06-30	↓ 다운로드
<input checked="" type="checkbox"/>	SHP	LSMD_ADM_SECT_UMD_대구.zip	용량 968,606 byte	구분 데이터	기준일 2025-06	갱신일 2025-06-30	↓ 다운로드

'LSMD_ADM_SECT_UMD_대구.zip' 파일 선택 다운로드.

시군구 행정구역 경계 파일 다운로드

다운로드 공간정보 다운로드

홈 > 다운로드 > 공간정보 다운로드

국토관리 지역개발 > 경계 행정구역시군구_경계

유형 다운로드 | 기관 국토교통부 | 다운로드 83,875 | 문의 45 | 조회 78,674 상세 ^

- 설명 법정동(시군구) 경계도면
- 라이선스 CC BY-NC-ND ^

↓ 웹툰 정의서 다운로드  

데이터 리소스 ^

파일명 10건씩 ^

선택 0 건 총 18 건 (1 / 2 page)

○ 등록된 데이터가 없습니다.

1 2

‘LARD_ADM_SECT_SGG_대구.zip’ 파일 선택 다운로드.

이 누리집은 대한민국 공식 전자정부 누리집입니다.

로그인 회원가입 사이트맵 ENGLISH

DATA .GO.KR 데이터찾기 국가데이터맵 데이터요청 기업 공공데이터 문제해결 지원센터 데이터활용 정보공유 이용안내

어떤 공공데이터를 찾으시나요?

인기검색어 1. 기상청

검색조건 분류체계 서비스유형 확장자

① 검색도움말 콘텐츠 바로가기

테마별 **컨텐츠리뷰** 국가중정데이터별 재공기관유형별

교육 국토관리 공공행정 재정금융 산업고용 사회복지 식품건강 문화관광

<https://www.data.go.kr/>

데이터 상세

[URL 복사](#)

대구광역시_시내버스 정류소 위치정보

2024년 9월 24일 기준으로 작성된 대구광역시 내에 운행 중인 시내버스 정류소의 전체 위치정보입니다. 정류소ID, 모바일ID, 정류소명(한글), 정류소명(영문), 행정구역, GPS(X-Y좌표), 경유노선수, 경유노선 등의 내용이 포함된 CSV 파일형식의 자료입니다. 시민들의 대중교통 이용 편의를 높이기 위해 제공되는 자료로, 정류소별 정보를 체계적으로 정리하여 이용자에게 정확한 정보를 전달드리고자 합니다. 추가적으로 궁금한 사항이나 의견이 있으시면 대구광역시 교통정보운영과 교통운영팀(☎053-803-6861)으로 연락해주시기 바랍니다.



0



0



관심

[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

대구광역시_시내버스 정류소 위치정보



다운로드

<https://www.data.go.kr/data/15050946/fileData.do>

데이터 상세

[URL 복사](#)

대구광역시_전기차 충전기 구축 현황

대구광역시에서 직접 구축하여 운영하고 있는 전기자동차 충전시설의 현황이며 충전소 명칭, 충전소 내의 충전기 아이디, 충전기 타입(급속, 완속), 설치 위치, 이용 가능 시간 등의 내용을 포함합니다. 또한 다음 어느 하나에 해당하는 경우에는 충전 요금의 50%를 감면합니다.(장애인 소유의 자동차, 국가유공자 소유의 자동차, 고엽제후유증환자 소유의 자동차, 병역명문가로 선정된 예우대상자 소유의 자동차). 참고로 대구광역시 관내에 설치 운영 중인 민간 충전시설의 현황은 환경부에서 운영하는 웹사이트인 무공해차통합누리집에서 확인할 수 있습니다.

[👍 1](#)[👎 0](#)[📄 관심](#)[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](#)[대구광역시_전기차 충전기 구축 현황](#)[다운로드](#)

<https://www.data.go.kr/data/15086278/fileData.do>



서비스명 *	<input type="text" value="Geocoding"/>
서비스분류 *	<input type="text" value="교육"/>
서비스유형 *	<input checked="" type="checkbox"/> 웹사이트 <input checked="" type="checkbox"/> APP(모바일, 솔루션 등) <input checked="" type="checkbox"/> 기타(QGIS, ArcGIS 등)
서비스URL	<input type="text" value="예) www.test.com/map.html"/>
서비스설명 *	<div><p>- 개요 : geocoding 교육을 위한 API 요청 - 서비스대상 : 교육용 - 목적 : Python을 이용한 geocoding 교육</p></div> <p>76 / 3000</p>
활용API *	<input checked="" type="checkbox"/> 3D 지도 API <input checked="" type="checkbox"/> 2D 지도 API <input checked="" type="checkbox"/> 배경지도 API <input checked="" type="checkbox"/> WMS/WFS API <input checked="" type="checkbox"/> WMTS/TMS API <input checked="" type="checkbox"/> 2D데이터 API <input checked="" type="checkbox"/> 지오코더 API <input checked="" type="checkbox"/> 검색 API <input checked="" type="checkbox"/> 이미지 API <input checked="" type="checkbox"/> 범례 API <input checked="" type="checkbox"/> 2D 모바일 API <input checked="" type="checkbox"/> 3D 모바일 API <input checked="" type="checkbox"/> 3D데스크톱 API <input checked="" type="checkbox"/> 국가중점 API
사용기관 *	<input type="text" value="민간"/> <input type="text" value="서울시립대학교"/>
활용사례공개	<input checked="" type="checkbox"/> 공개동의

vworld(브이월드) API 요청 페이지.

인증키 관리

[인증키 발급하기](#)

○ 오픈API를 사용하기 위해 발급받은 인증키를 관리할 수 있습니다.

- URL이 다른 여러개의 사이트 개발을 하시기 위해서는, 각각 별도의 오픈API 인증키 발급을 받으시기 바랍니다.
- 개발키 : 개발을 목적으로하며 유효기간은 6개월, 최대 3회 연장가능합니다.
- 운영키 : 서비스 운영을 목적으로하며 유효기간은 2년, 유효기간 연장 신청을 통해 관리자 심사 후 연장가능합니다.

※ 인증키 연장은 만료일 이전에 가능합니다.

총 1 건

개발키	A06352C6	[REDACTED]	1DDC84CF8332	Geocoding	발급일 2025-08-13	만료일 2026-02-13	● 개발
-----	----------	------------	--------------	-----------	----------------	----------------	------

geocoding 등 다양한 정보를 수집하기 위한 인증키 관리.

튜토리얼

- Colab 링크
- 라이브러리의 버전을 고려하여 설치.
- pip 명령어는 터미널(terminal)에서 직접 하거나 python script에 !를 붙여 사용).

```
1 !pip install geopandas==0.14.4 geodatasets==2024.8.0
2 !pip install mapclassify
```

- 라이브러리 불러오기.

```
1 import os
2 import pandas as pd
3 import geopandas as gpd
4 import geodatasets
5 import matplotlib.pyplot as plt
```

- 현재 설치된 라이브러리 버전 확인.
- 터미널에서도 확인 가능 → pip list (또는 가상환경에선 conda list).

```
1 print("geopandas:", gpd.__version__)
2 print("geodatasets:", geodatasets.__version__)
```

기본 내장 데이터 불러오기

- Geopandas의 기본 내장 데이터 불러오기.
- head() 함수를 이용한 변수명 및 데이터 확인.

```
1 countries = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
2 countries.head()
3
4 cities = gpd.read_file(gpd.datasets.get_path('naturalearth_cities'))
5 cities.head()
```

	pop_est	continent	name	iso_a3	gdp_md_est	geometry
0	889953.0	Oceania	Fiji	FJI	5496	MULTIPOLYGON (((180.00000 -16.06713, 180.00000...
1	58005463.0	Africa	Tanzania	TZA	63177	POLYGON ((33.90371 -0.95000, 34.07262 -1.05982...
2	603253.0	Africa	W. Sahara	ESH	907	POLYGON ((-8.66559 27.65643, -8.66512 27.58948...
3	37589262.0	North America	Canada	CAN	1736425	MULTIPOLYGON (((-122.84000 49.00000, -122.9742...
4	328239523.0	North America	United States of America	USA	21433226	MULTIPOLYGON (((-122.84000 49.00000, -120.0000...

- 세계지도 그리기.

```
1 ax = countries.plot()  
2 plt.show()
```

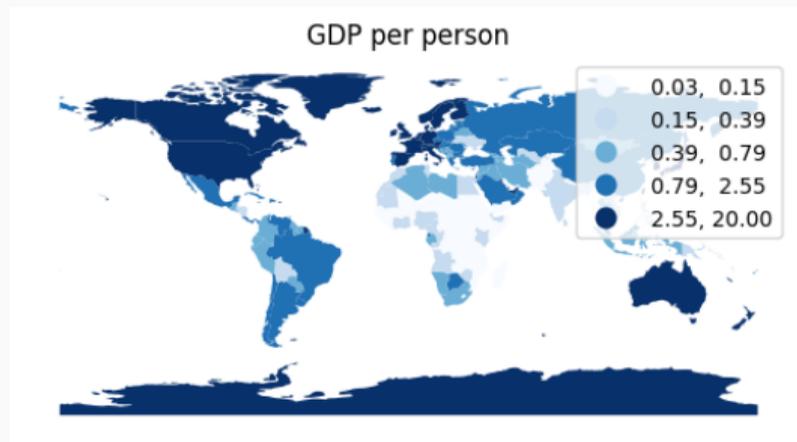
- 세계지도 그리기 (대륙).

```
1 ax = countries.plot(column="continent", legend=True, categorical=True)  
2 ax.set_title("World Map")  
3 ax.set_axis_off()  
4 plt.show()
```



- 1인당 GDP 계산 및 시각화.

```
1 countries['gdp_per_cap'] = countries['gdp_md_est'] / countries['pop_est'] *100
2 ax = countries.plot(column='gdp_per_cap', legend=True, scheme='quantiles', cmap="
    Blues", k=5)
3 ax.set_axis_off()
4 ax.set_title("GDP per person")
5 plt.show()
```



데이터와 좌표계

실습 데이터 다운로드

- 브이월드에서 행정구역(시군구/읍면동) 경계 데이터 다운로드.
- 공공데이터포털에서 전기차 충전소, 시내버스 정류소 위치 데이터 다운로드.
- API 인증키 받기.

좌표계 변환

- epsg를 이용한 좌표계 설정.
- shp 파일 vs gpkg 파일.
- gpkg 파일 저장.
- 경계 데이터 시각화.

지도 위에 point 그리기

- 전기차 충전소, 시내버스 정류소의 위치 정보를 geometry 타입으로 변환.
- 주소만 주어진 경우 geocoder를 이용하여 위치 정보 생성.
- 지도 위에 충전소/정류소 위치 시각화.

- 대구행정구역.

```
1 !wget --no-check-certificate "https://drive.google.com/uc?export=download&id=12
   rE4QyeW1XHZD6A_mJsEPU_ld33aUrf4" -O LSMD_ADM_SECT_UMD_daegu.zip
2 !wget --no-check-certificate "https://drive.google.com/uc?export=download&id=1
   IWRQ0tQ0mn1JV_0xhkC1YtZ050Hz7Fzd" -O LARD_ADM_SECT_SGG_daegu.zip
```

- 압축풀기.

```
1 !unzip LSMD_ADM_SECT_UMD_daegu.zip -d LSMD_ADM_SECT_UMD_daegu
2 !unzip LARD_ADM_SECT_SGG_daegu.zip -d LARD_ADM_SECT_SGG_daegu
```

- 대구 시내버스, 전기차 정보.

```
1 !wget --no-check-certificate "https://drive.google.com/uc?export=download&id=1
   mSeLfekQT98fanB19uZTEq2KCKE-0hWr" -O "daegu_20250513.csv"
2 !wget --no-check-certificate "https://drive.google.com/uc?export=download&id=1
   yEM2ynUw8eSw3wH8RElxVoYvEvd4MpH9" -O "daegu_20240924.csv"
```

- 경로 확인 및 설정.

```
1 print(os.getcwd())
2 path = "./LARD_ADM_SECT_SGG_daegu"
3 files = os.listdir(path)
4 files
```

```
/content
['LARD_ADM_SECT_SGG_27_202505.prj',
'LARD_ADM_SECT_SGG_27_202505.shp',
'LARD_ADM_SECT_SGG_27_202505.cpg',
'LARD_ADM_SECT_SGG_27_202505.dbf',
'LARD_ADM_SECT_SGG_27_202505.qix',
'LARD_ADM_SECT_SGG_27_202505.shx']
```

대구광역시 시군구 데이터 불러오기 (2)

- 데이터 로드 (시군구).

```
1  fname = [v for v in files if v.split(".")[1] == "shp"] [0]
2  sgg= gpd.read_file(os.path.join(path, fname), encoding="EUC-KR")
3  sgg
```

	ADM_SECT_C	SGG_NM	SGG_OID	COL_ADM_SE	geometry
0	27110	대구광역시 중구	1.0	27110	POLYGON ((343043.528 365694.160, 343058.479 36...
1	27140	대구광역시 동구	1.0	27140	POLYGON ((352845.179 381127.667, 352916.234 38...
2	27170	대구광역시 서구	1.0	27170	POLYGON ((340143.350 367423.207, 340132.078 36...
3	27200	대구광역시 남구	NaN	27200	POLYGON ((342915.592 363290.726, 342983.715 36...
4	27230	대구광역시 북구	689.0	27230	POLYGON ((345552.450 377111.749, 345554.918 37...
5	27260	대구광역시 수성구	1.0	27260	POLYGON ((349090.018 365825.403, 349109.202 36...
6	27290	대구광역시 달서구	1.0	27290	POLYGON ((337326.259 364423.689, 337340.372 36...
7	27710	대구광역시 달성군	3.0	27710	MULTIPOLYGON (((334062.134 360343.608, 334068...
8	27720	대구광역시 군위군	34.0	27720	POLYGON ((330344.249 415327.087, 330465.736 41...

- 데이터 시각화.

```
1 ax = sgg.plot(column="SGG_NM", categorical=True)
2 ax.set_title("Daegu(SGG)")
3 ax.set_axis_off()
4 plt.show()
```

- `.to_crs()`를 이용하여 좌표계 변환.
- `shp` 파일을 `gpkg` 파일로 변환하여 저장.

```
1 sgg = sgg.to_crs(epsg = 5179)
2 sgg.to_file("./daegu_ssg_5179.gpkg", driver="GPKG", encoding="UTF-8")
```

대구시 법정동 파일 좌표계 변환하여 저장하기

- 대구시 읍면동(UMD) 파일 로드.
- 시각화 및 좌표계 지정 (`set_crs()` 활용 및 `epsg=5186`).
- 좌표계 변환 (`epsg=5179`).
- GPKG 형식으로 저장.
- 파일명: 대구시_법정동_5179.



특정 좌표 데이터 불러오기 (전기차 충전소)

- 전기차 충전기 csv 데이터 불러오기.

```
1 path = "./"  
2 fname = "daegu_car_20250513.csv"  
3 df = pd.read_csv(os.path.join(path, fname), encoding="CP949")  
4 df.head()
```

	충전소명	주소	ID	충전기 형태	충전기 용량
0	(RS)대구시_효목2동	대구 동구 화랑로25길 45(효목동, 효목2동행정복지센터)	1	3Mode	100
1	(RS)대구시_효목1동 행정복지센터	대구 동구 화랑로41길 46(효목동, 효목1동행정복지센터)	1	3Mode	50
2	(RS)대구시_황금1동 행정복지센터	대구 수성구 청호로57길 10(황금동, 황금1동 행정복지센터)	1	3Mode	50
3	(RS)대구시_현풍읍행정복지센터	대구 달성군 현풍읍 비슬로134길 157(부리, 현풍읍사무소) 주차장	1	AC완속(싱글)	7
4	(RS)대구시_해안동행복복지센터	대구 동구 방촌로29길 46(방촌동, 해안동행정복지센터) 주차장	1	3Mode	50

특정 좌표 데이터 불러오기 (시내버스 정류소)

- 시내버스 csv 데이터 불러오기.

```
1 path = "./"  
2 fname = "daegu_bus_20240924.csv"  
3 df = pd.read_csv(os.path.join(path, fname), encoding="CP949")  
4 df.head()
```

	정류소아이디	모바일아이디	정류소명	영문명	시도	구군	동	경도	위도	경유노선수	경유노선
0	7361000423	10423.0	정리	Jeong-ri	대구광역시	군위군	군위읍	128.560200	36.251510	1	군위3
1	7361000424	10424.0	정리	Jeong-ri	대구광역시	군위군	군위읍	128.560072	36.251480	2	군위2+군위3
2	7361000426	10426.0	정리	Jeong-ri	대구광역시	군위군	군위읍	128.562187	36.246557	2	군위2+군위3
3	7361000428	10428.0	사직1리	sajik1ri	대구광역시	군위군	군위읍	128.556720	36.229607	1	군위4
4	7361000429	10429.0	사직1리	sajik1ri	대구광역시	군위군	군위읍	128.556858	36.229608	1	군위4

- geommetry 생성 및 GeoDataFrame 변환.

```
1 geometry = gpd.points_from_xy(df['lon'], df['lat'])
2 gdf = gpd.GeoDataFrame(df, geometry = geometry, crs=4326)
3 gdf.head()
```

- GPKG 변환 및 저장.

```
1 path = "./"
2 f = "daegu_bus_20240924.gpkg"
3 gdf.to_file(os.path.join(path, f), driver="GPKG", encoding="UTF-8")
```

	정류소아이디	모바일아이디	정류소명	영문명	시도	구군	동	경도	위도	경유노선수	경유노선	geometry
0	7361000423	10423.0	정리	Jeong-ri	대구광역시	군위군	군위를	128.560200	36.251510	1	군위3	POINT (128.56020 36.25151)
1	7361000424	10424.0	정리	Jeong-ri	대구광역시	군위군	군위를	128.560072	36.251480	2	군위2+군위3	POINT (128.56007 36.25148)
2	7361000426	10426.0	정리	Jeong-ri	대구광역시	군위군	군위를	128.562187	36.246557	2	군위2+군위3	POINT (128.56219 36.24656)
3	7361000428	10428.0	사직1리	sajik1ri	대구광역시	군위군	군위를	128.556720	36.229607	1	군위4	POINT (128.55672 36.22961)
4	7361000429	10429.0	사직1리	sajik1ri	대구광역시	군위군	군위를	128.556858	36.229608	1	군위4	POINT (128.55686 36.22961)

- 전기차 충전소는 도로명 주소는 존재하지만 좌표계 정보가 없음.
- Geometry 정보를 얻기 위해 주소명으로부터 위치 정보를 가져와야함.

```
1 path = "./"  
2 fname = "daegu_car_20250513.csv"  
3 df = pd.read_csv(os.path.join(path, fname), encoding="CP949")  
4 df.head()
```

address = '대구광역시 수성구 파동 274'

```
1 geolocator = Nominatim(user_agent="geoapi")
2 location = geolocator.geocode(address)
3 if location:
4     print(location)
5     print([location.longitude, location.latitude])
6 else:
7     print("No address")
```

파동, 수성구, 대구광역시, 대한민국
[128.6153796, 35.8147288]

- 지오코더(Geocoder)를 이용한 좌표 계산.

```
1  from geopy.geocoders import Nominatim
2  import pandas as pd
3  import numpy as np
4  import time
5
6  geolocator = Nominatim(user_agent="geoapi")
7
8  def clean_address(addr):
9      return addr.split(',') [0].strip()
10
11 def geocoder(address):
12     try:
13         location = geolocator.geocode(address)
14         if location:
15             return pd.Series([location.longitude, location.latitude])
16     except:
17         return pd.Series([np.nan, np.nan])
18     return pd.Series([np.nan, np.nan])
```

- 주소 클린 후 좌표 변환.

```
1 df["address_clean"] = df["address"].apply(clean_address)
2
3 coords = []
4 for addr in df["address_clean"]:
5     coords.append(geocoder(addr))
6     time.sleep(1)
7
8 df[["x", "y"]] = pd.DataFrame(coords, index=df.index)
9 result = df
```

	충전소명	주소	ID	충전기 형태	충전기 용량	주소_clean	x	y
0	(RS)대구시_효목2동	대구 동구 화랑로25길 45(효목동, 효목2동행정복지센터)	1	3Mode	100	대구 동구 화랑로25길 45	128.641323	35.878114
1	(RS)대구시_효목1동 행정복지센터	대구 동구 화랑로41길 46(효목동, 효목1동행정복지센터)	1	3Mode	50	대구 동구 화랑로41길 46	128.644967	35.881647
2	(RS)대구시_황금1동 행정복지센터	대구 수성구 청호로57길 10(황금동, 황금1동 행정복지센터)	1	3Mode	50	대구 수성구 청호로57길 10	128.638140	35.843972
3	(RS)대구시_현풍읍행정복지센터	대구 달성군 현풍읍 비슬로134길 157(부리, 현풍읍사무소) 주차장	1	AC완속(싱글)	7	대구 달성군 현풍읍 비슬로134길 157	128.447662	35.697475
4	(RS)대구시_해안동행정복지센터	대구 동구 방촌로29길 46(방촌동, 해안동행정복지센터) 주차장	1	3Mode	50	대구 동구 방촌로29길 46	128.671607	35.882121

- Geometry 생성 및 좌표계 설정.

```
1 geometry = gpd.points_from_xy(result['x'], result['y'])
2 gdf = gpd.GeoDataFrame(result, geometry = geometry, crs=4326)
3 gdf.head()
```

	충전소명	주소	ID	충전기 형태	충전기 용량	주소_clean	x	y	geometry
0	(RS)대구시_효목2동	대구 동구 화랑로25길 45(효목동, 효목2동행정복지센터)	1	3Mode	100	대구 동구 화랑로25길 45	128.641323	35.878114	POINT (128.64132 35.87811)
1	(RS)대구시_효목1동 행정복지센터	대구 동구 화랑로41길 46(효목동, 효목1동행정복지센터)	1	3Mode	50	대구 동구 화랑로41길 46	128.644967	35.881647	POINT (128.64497 35.88165)
2	(RS)대구시_황금1동 행정복지센터	대구 수성구 청호로57길 10(황금동, 황금1동 행정복지센터)	1	3Mode	50	대구 수성구 청호로57길 10	128.638140	35.843972	POINT (128.63814 35.84397)
3	(RS)대구시_현풍음행정복지센터	대구 달성군 현풍읍 비슬로134길 157(부리, 현풍읍사무소) 주차장	1	AC완속(싱글)	7	대구 달성군 현풍읍 비슬로134길 157	128.447662	35.697475	POINT (128.44766 35.69747)
4	(RS)대구시_해안동행정복지센터	대구 동구 방촌로29길 46(방촌동, 해안동행정복지센터) 주차장	1	3Mode	50	대구 동구 방촌로29길 46	128.671607	35.882121	POINT (128.67161 35.88212)

- GPKG 형태로 저장.

```
1 path = "./"  
2 f = "daegu_car_20250513.gpkg"  
3 gdf.to_file(os.path.join(path, f), driver="GPKG", encoding="UTF-8")
```

- 시군구/법정동 별로 충전소 분포 확인.

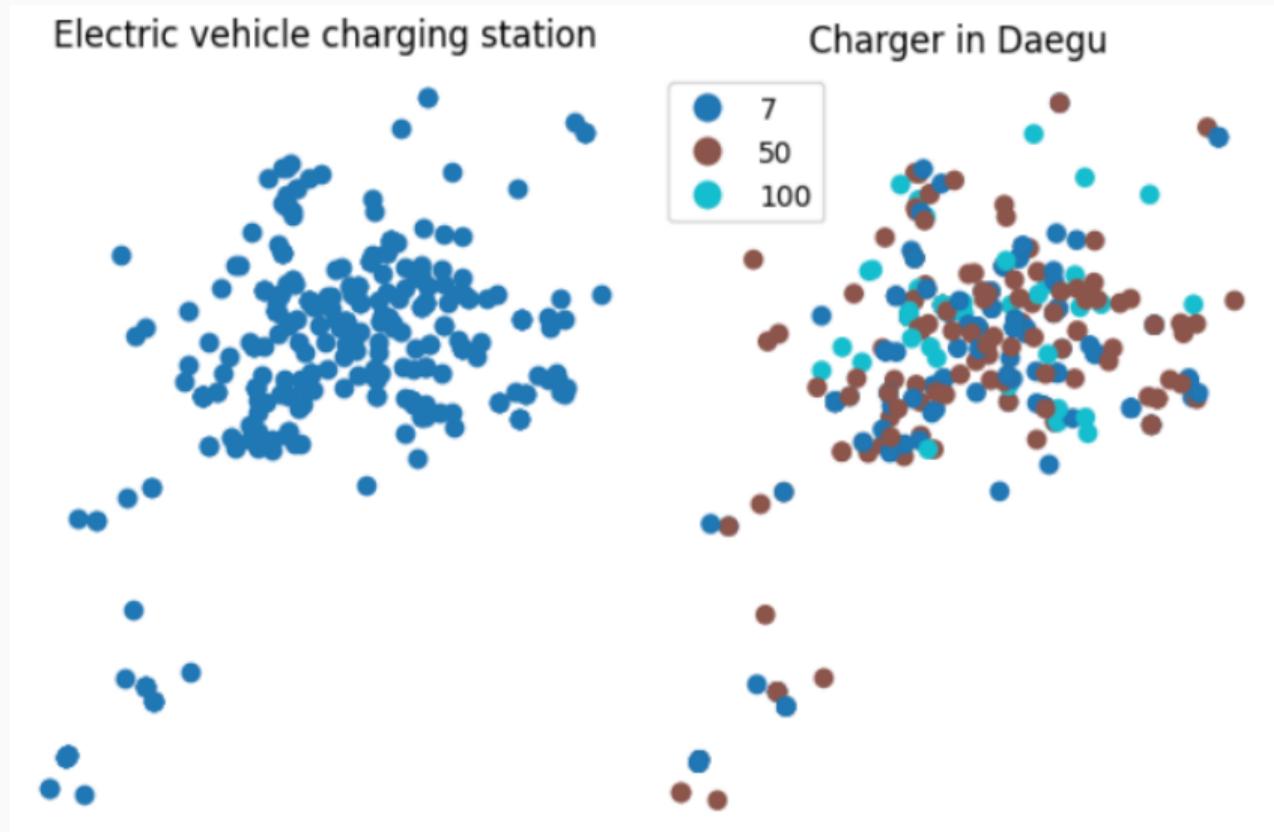
```
1 path = "./"  
2 data = gpd.read_file(os.path.join(path, "daegu_car_20250513.gpkg"))  
3 sgg = gpd.read_file(os.path.join(path, "daegu_sgg_5179.gpkg"))  
4 bjd = gpd.read_file(os.path.join(path, "daegu_bjd_5179.gpkg"))  
5 data.head(5)
```

- 데이터 시각화 (지도 없이 오직 전기차 충전기 위치만).

```
1 ax = data.plot()  
2 ax.set_title("Electric vehicle charging station")  
3 ax.set_axis_off()  
4 plt.show()
```

- 데이터 시각화 (용량별).

```
1 ax = data.plot(column="size", legend=True, categorical=True)  
2 ax.set_title("Charger in Daegu")  
3 ax.set_axis_off()  
4 plt.show()
```



좌표 불일치 문제

- 데이터 시각화 (시군구 + 충전소).
- 시군구 데이터와 충전소 데이터의 좌표계가 맞지 않음.

```
1 base = sgg.plot(color='white', edgecolor="k")
2 ax = data.plot(ax= base)
3 ax.set_title("Charger in Daegu")
4 ax.set_axis_off()
5 plt.show()
```



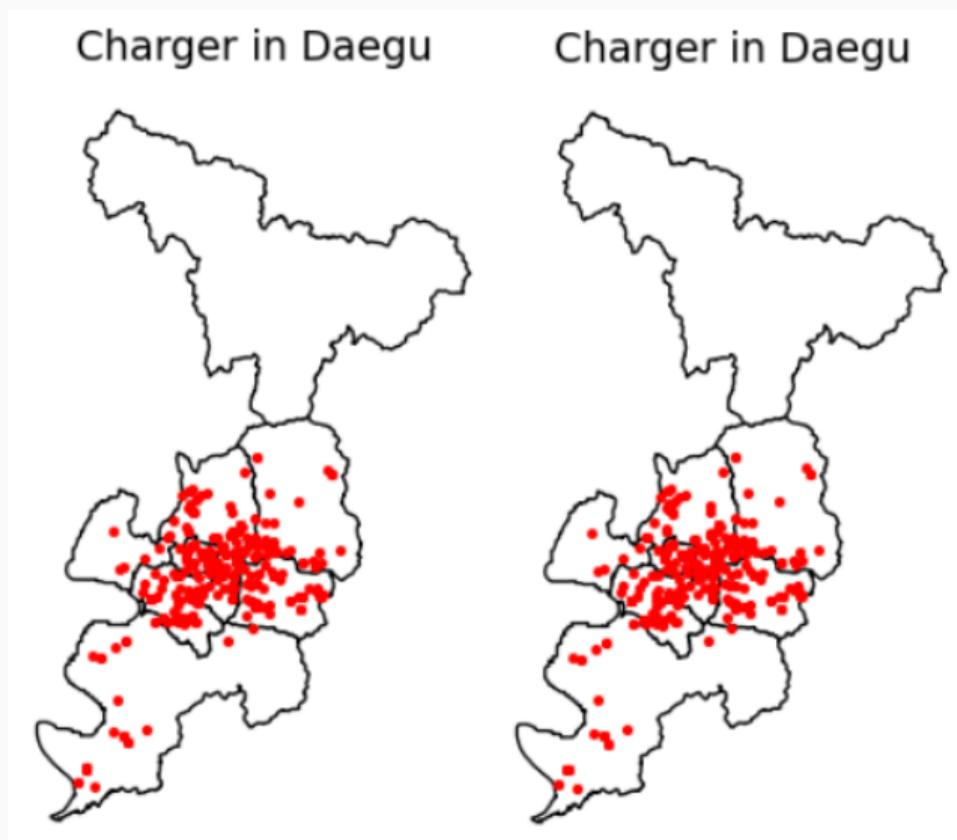
- sgg를 WGS84로 변환.

```
1 sgg_wgs84 = sgg.to_crs(epsg=4326)
2 base = sgg_wgs84.plot(color='white', edgecolor='k')
3 ax = data.plot(ax=base, color='red', markersize=7)
4 ax.set_title("Charger in Daegu", fontsize=14)
5 ax.set_axis_off()
6 plt.show()
```

- data를 EPSG:5179로 변환.

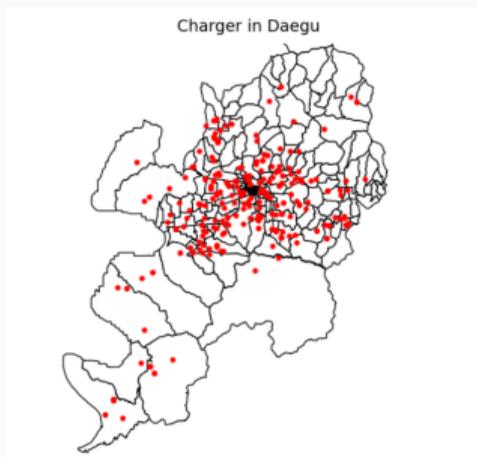
```
1 data_5179 = data.to_crs(epsg=5179)
2 base = sgg.plot(color='white', edgecolor='k')
3 ax = data_5179.plot(ax=base, color='red', markersize=7)
4 ax.set_title("Charger in Daegu", fontsize=14)
5 ax.set_axis_off()
6 plt.show()
```

좌표 EPSG 변환 및 동기화 (시각화)



- 시군구대신 법정동으로 시각화.

```
1 fig, ax = plt.subplots(figsize=(12, 12))
2 base = bjd.plot(ax=ax, color='white', edgecolor='k')
3 data_5179.plot(ax=base, color='red', markersize=14)
4 ax.set_title("Charger in Daegu", fontsize=14)
5 ax.set_axis_off()
6 plt.show()
```



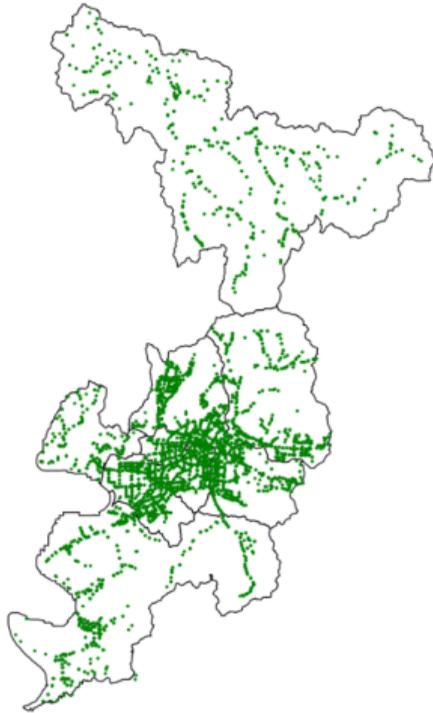
- 시군구/법정동 별로 시내버스 정류장 분포 확인.

```
1 path = "./"  
2 data = gpd.read_file(os.path.join(path, "daegu_bus_20240924.gpkg"))  
3 data.head(5)
```

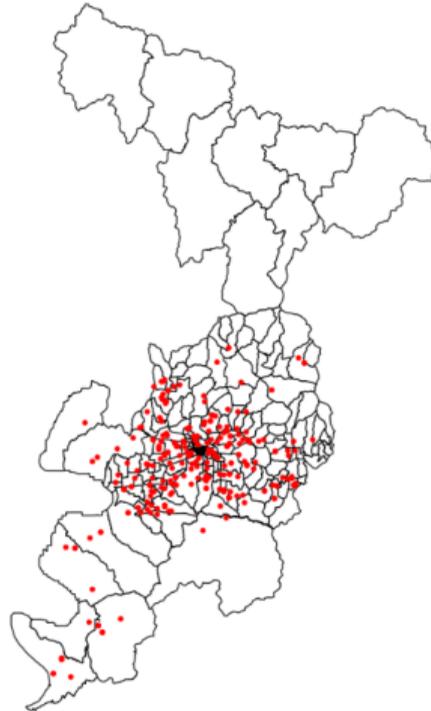
- data를 EPSG:5179로 변환.

```
1 data_5179 = data.to_crs(epsg=5179)  
2 fig, ax = plt.subplots(figsize=(12, 12))  
3 base = sgg.plot(ax=ax, color='white', edgecolor='k')  
4 data_5179.plot(ax=base, color='green', markersize=3)  
5 ax.set_title("Charger in Daegu", fontsize=14)  
6 ax.set_axis_off()  
7 plt.show()
```

Charger in Daegu



Charger in Daegu



본 강의자료는 대구빅데이터활용센터(<https://dipbigdata.kr/>)에서 제공하는 *GeoPandas* 공간분석 강의자료를 참고하여 작성하였습니다.