

시와 데이터 기초

텍스트 시각화

오늘 수업은

- ❖ 워드클라우드의 이해
- ❖ 기본적인 워드클라우드 실습
- ❖ 추가속성 설정하기
- ❖ 한글 텍스트를 이용한 워드클라우드 생성
- ❖ [실습 내용]
 - 위키피디아의 [세종대왕]과 미곡생산량을 활용한 텍스트 시각화분석
- ❖ folium 이해하기
- ❖ folium 을 활용한 Map 그리기
- ❖ 지도에 Marker 추가하기
- ❖ 지도에 MarkerCluster 추가하기
- ❖ [실습 내용]
 - 서울시 공영주차장 위치 표시하기

워드클라우드란?

❖ 워드클라우드(WordCloud)

- 텍스트에서 단어들을 분석하여 **인기도** 및 **중요도에** 따라 글자 색상, 크기 및 굵기의 형태로 **시각적 이미지**로 표현하는 것
 - 인기도 및 중요도는 빈도수로 표현될 수 있음



WordCloud 준비하기

❖ 설치하기

```
pip install wordcloud
```

❖ 코랩(colab)에는 이미 설치되어 있음

❖ 라이브러리 선언

- WordCloud는 matplotlib 라이브러리를 기반으로 하고 있으므로 함께 선언

```
1 from wordcloud import WordCloud  
2 import matplotlib.pyplot as plt
```

1. 라이브러리 설치 및 한글 폰트 설치

❖ 자연어 처리를 위한 라이브러리 설치하기

```
pip install konlpy
```

❖ Colab 한글 폰트 설치

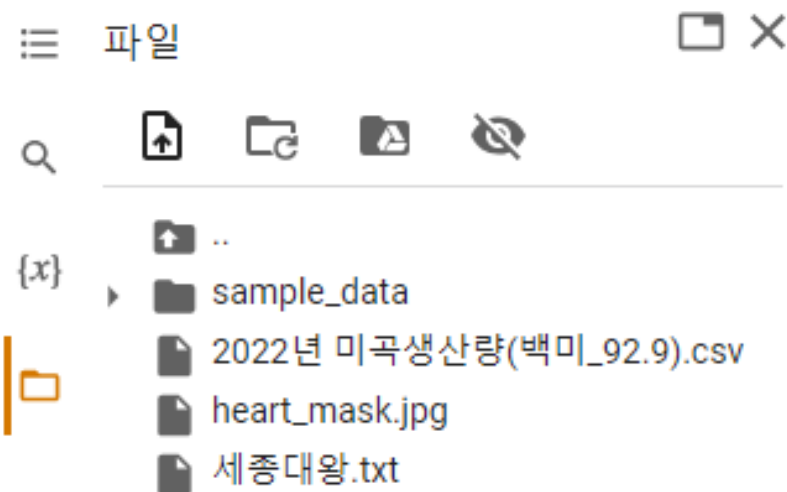
```
!sudo apt-get install -y fonts-nanum
```

```
!sudo fc-cache -fv
```

```
!rm ~/.cache/matplotlib -rf
```

2. 텍스트 파일과 이미지 파일 불러오기

❖ 텍스트 파일과 이미지 파일 업로드하기



❖ 명령어를 통하여 파일 업로드 하기

```
from google.colab import files
files.upload()
```

3. 필요한 라이브러리들 로딩하기

❖워드 클라우드 생성에 필요한 라이브러리들(wordcloud, STOPWORDS, matplotlib)

```
from wordcloud import WordCloud  
from wordcloud import STOPWORDS  
import matplotlib.pyplot as plt
```

❖마스크 이미지 처리를 위해 필요한 라이브러리들(Image, numpy)

```
from PIL import Image  
import numpy as np
```

4. 관련 파일들 open 하고 저장하기

❖ 텍스트 파일을 열어 읽어오기

- 파일변수명 = `open('파일명', 'r', encoding= '인코딩방식')`
- 텍스트변수명 = 파일변수명.`read()`
 - 파일에 저장되어 있는 본문을 읽음

❖ CSV 파일 읽어오기

- 변수명 = `pd.read_csv(파일명, encoding= '인코딩방식' , skiprows=[행번호])`

❖ 마스크 이미지 파일 읽어와서 array로 변경하기

- 마스크변수명 = `np.array(Image.open('마스크이미지파일명'))`
 - 마스크 이미지를 읽어와서 Array 자료형으로 변환

5. 자연어 처리하기

- ❖ 한글 자연어처리를 위한 konlpy 라이브러리 로딩하기

```
from konlpy.tag import Okt
```

- ❖ Okt 라이브러리를 사용하기 위한 객체 생성
 - 변수명 = Okt()
- ❖ 문자열에서 명사만 추출하기
 - 변수명.nouns(문자열)

6. 워드 클라우드 속성 지정하기

❖ 불용어 처리를 위한 단어 추가하기

- 변수명 = **STOPWORDS.union**({'단어들'})
 - 단어들: 불용어로 처리하고자 하는 단어들을 쉼표(,)로 분리하여 기술

❖ 워드 클라우드 속성 설정하기

- 변수명 = **WordCloud**(width= 정수, height= 정수,
stopwords= 불용어변수명,
font_path = 폰트경로,
mask = 마스크변수명,
background_color= '배경색')

6. 워드 클라우드 생성하기

❖ 문자열을 활용하여 워드 클라우드 생성하기

- 변수명 `.generate(문자열변수명)`

❖ 딕셔너리 데이터를 활용하여 워드클라우드 생성하기

- 변수명 `.generate_from_frequencies(dictionary변수명)`

7. 워드클라우드 출력 및 눈금 숨기기

❖ 워드클라우드 이미지 출력하기

- `plt.imshow(변수명)`

❖ x, y축 눈금 숨기기

- `plt.axis('off')` : x, y축에 표시되는 눈금 제거

지도 시각화



❖ leaflet.js 기반으로 만들어진 지도 시각화에 특화된 라이브러리

- 자바 스트림트 기반으로 만들어져서 웹에서 출력 용이

❖ 무료로 사용 가능

❖ 지도 생성, 마커 표시, 행정구역 경계선 및 색상 표현, html파일로 내보내기등 다양한 기능 제공

❖ 다양한 plugin을 제공함으로 좀더 복잡하고 시각적인 표현 가능

❖ folium 설치

- colab 의 경우 이미 설치되어 있음

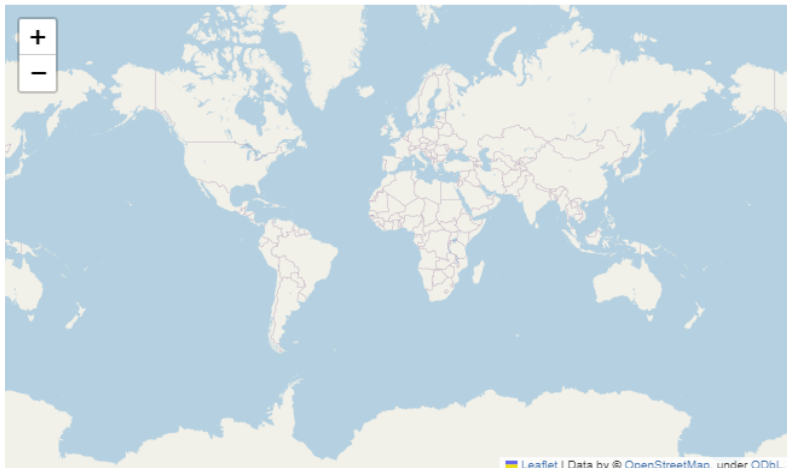
```
pip install folium
```

❖ folium 선언

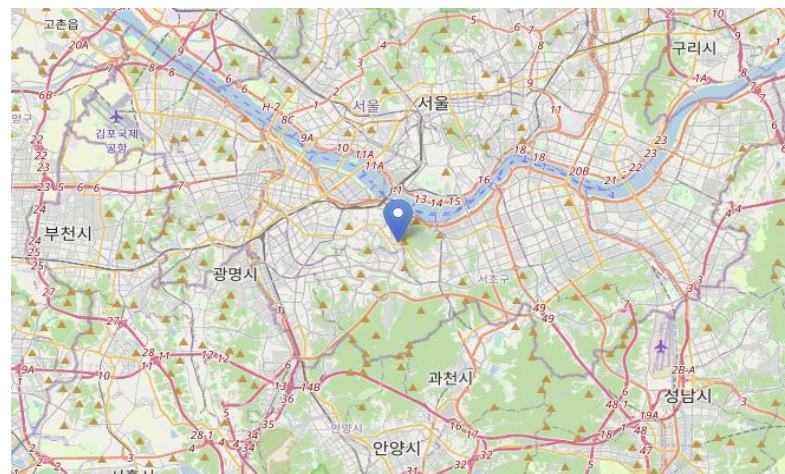
```
import folium
```

folium을 이용한 지도 시각화 순서

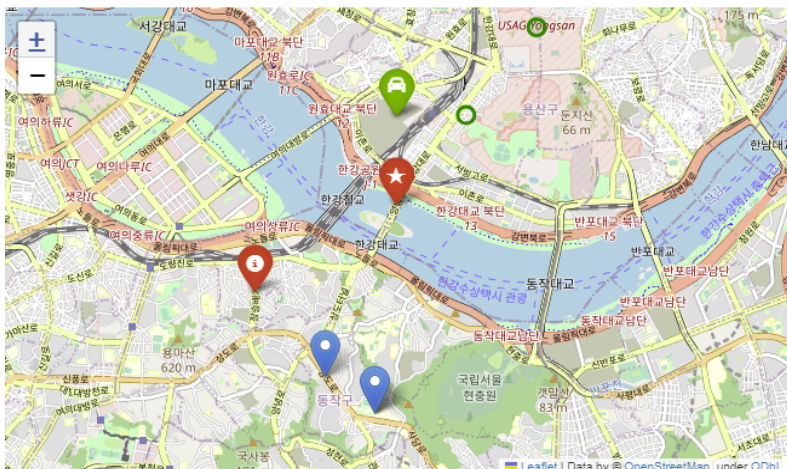
① folium 을 활용한 Map 그리기



② 지도에 Marker 표시하기



③ 지도에 여러 Marker 표시하기



④ MarkerCluster로 표현하기



지도 그리기

❖ folium.Map(속성들)

- 지도를 그려주는 객체
- `location=[위도, 경도]` : 지도의 중심 좌표를 지리 좌표계인 [위도, 경도] 또는(위도, 경도)로 나타냄
- `zoom_start = 정수` : 지도를 처음 그릴때 확대 정도
- `zoom_control=True` : zoom in/out 버튼 표시 여부
- `control_scales=False` : 스케일 컨트롤 버튼 표시 여부
- `tiles= '스타일'` : 지도 스타일을 지정

지도상에 Marker와 MarkerCluster 표시하기

❖ 지도상에 일반 Marker 표시하기

- `folium.Marker(속성들).add_to(MarkerCluster변수명)`
 - `location=[위도, 경도]` : marker가 표시되는 위치정보(위도,경도)
 - `popup= '문자열'` : marker를 클릭했을때 나타나는 메시지(문자열)
 - `tooltip= '문자열'` : marker에 마우스를 올렸을때 나타나는 메시지(문자열)
 - `icon= '모양'` : marker의 모양을 설정

❖ MarkerCluster 표현하기

- `변수명 = MarkerCluster().add_to(지도변수명)`

❖ DataFrame의 각 행의 정보 읽어오기

- `df.iterrow()` : DataFrame에 저장된 데이터를 index 단위로 읽어옴

서울시 공영주차장 위치 표시하기


서울시 공영주차장 위치 표시하기

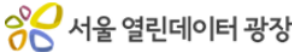
❖ 서울시 공영주차장 안내 정보

■ 서울시 공영주차장 안내 정보.csv

제공되는 데이터 파일은 공공데이터를 수정한 파일임(중복항목 제거, 누락된 위도/경도를 삽입)

■ <http://data.seoul.go.kr/dataList/OA-13122/S/1/datasetView.do>

 출산에서 육아까지 '몽땅정보 만능키' 누리집 운영 개시 서울소식 응답소 정보공개 분야별정보

 공공데이터 통계 서울빅데이터 소식&참여 이용안내 로그인 | 회원가입 | 사이트맵


데이터셋

Home > 공공데이터 > 공공데이터

Q 상세 검색 통합 검색

☐ 결과 내 재검색

공공데이터


교통

활용사례(갤러리) 등록

URL 복사

목록 이동

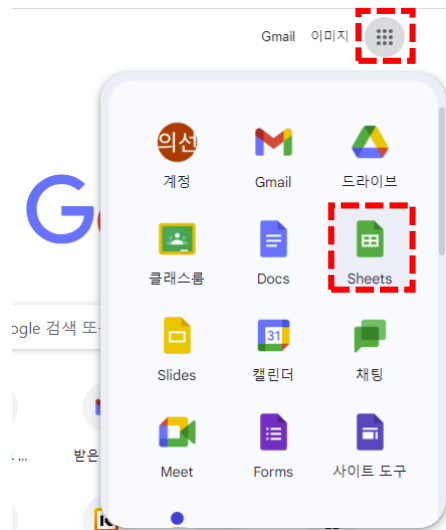
서울시 공영주차장 안내 정보

공영 노상, 노외 주차장 운영 정보를 안내하고 있으며, 주차장 여건에 따라 실제 정보와 다를 수 있습니다.
※ 기타 사항에 대하여는 원본 주관부서인 교통기획관 주차계획과(이종운주무관 02-2133-2356) 또는 시스템운영(070-4477-6913)로 문의 바랍니다.

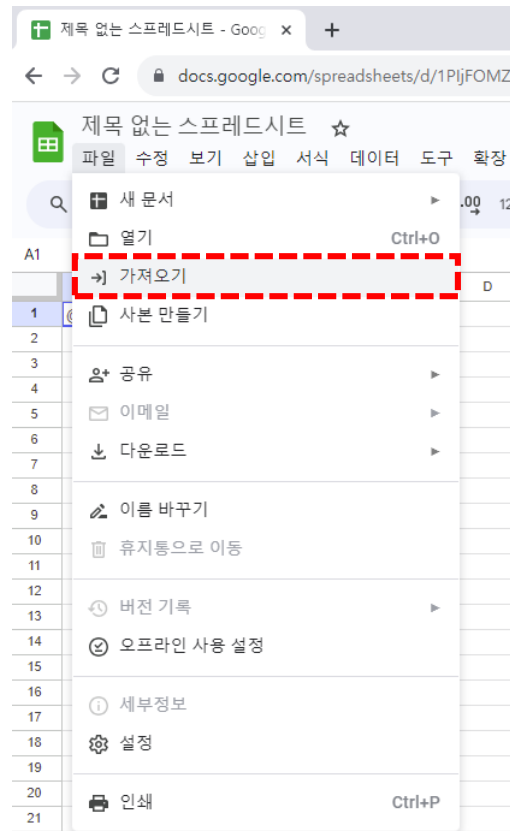
서울시 공영주차장의 위도/경도 찾기

❖ 구글 sheet에서 제공하는 Geocode by Awesome Table를 활용

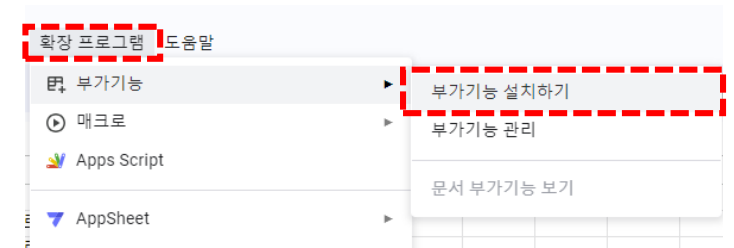
① 구글의 Sheets 실행



② 파일 가져오기



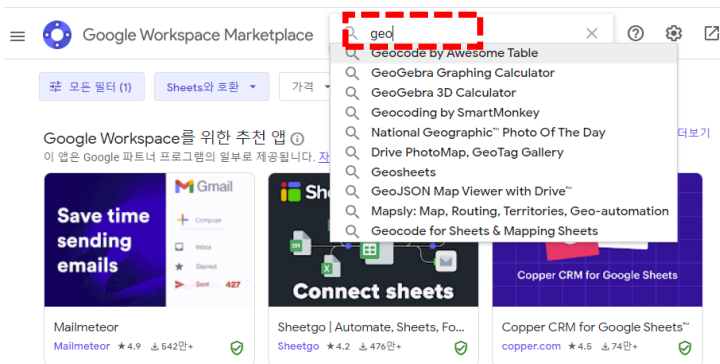
③ 부가기능 설치하기



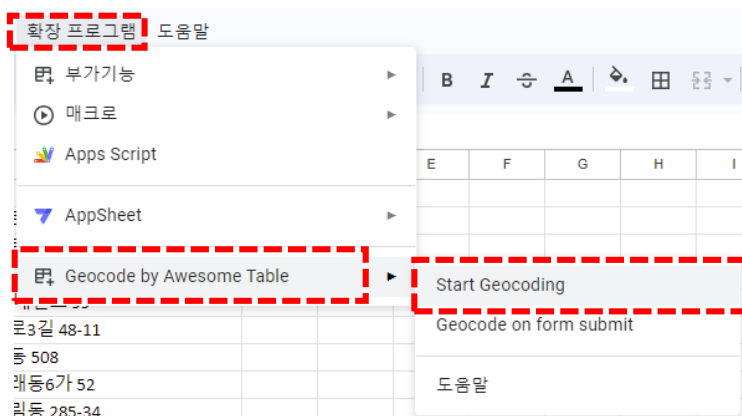
서울시 공영주차장의 위도/경도 찾기

❖ 구글 sheet에서 제공하는 Geocode by Awesome Table를 활용

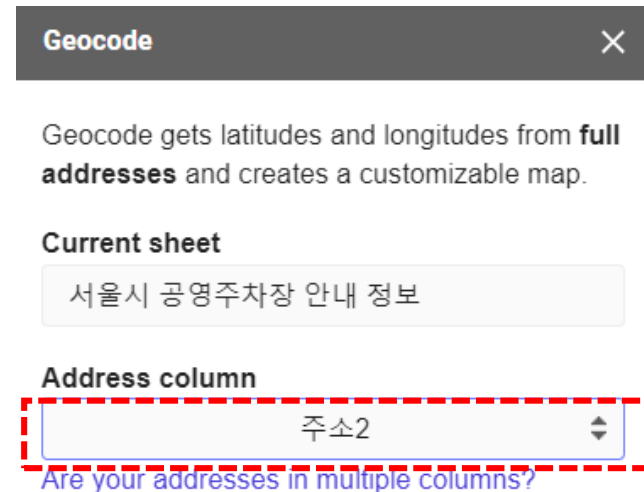
④ Geocode 검색



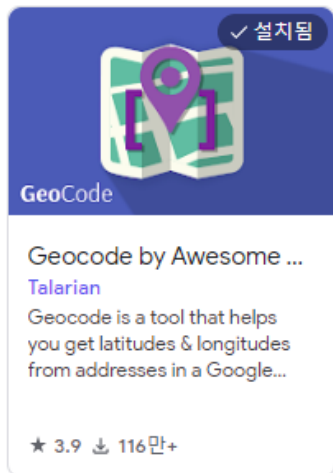
⑤ Start Geocoding 실행



⑥ 테이블의 주소열 선택



⑤ Geocode by Awesome Table 설치



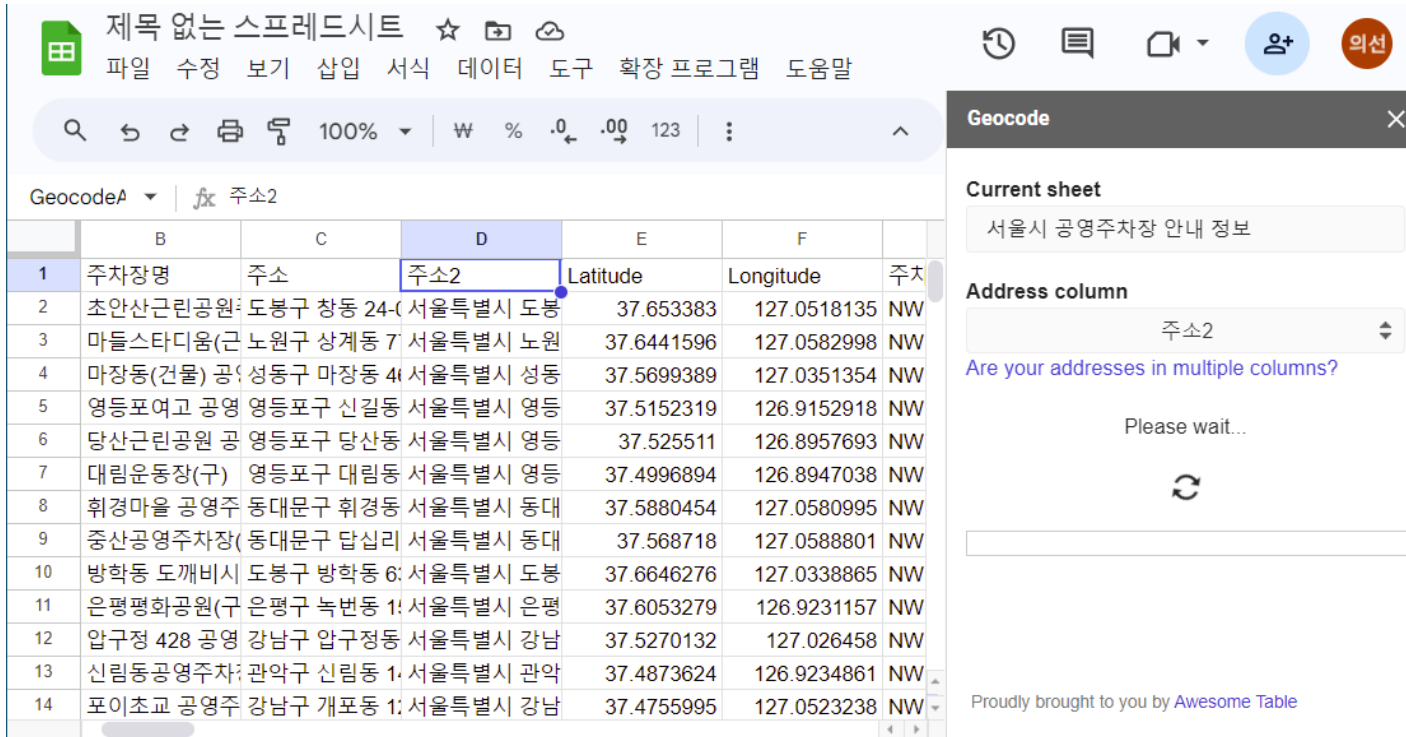
Geocode!

Proudly brought to you by Awesome Table

서울시 공영주차장의 위도/경도 찾기

❖ 구글 sheet에서 제공하는 Geocode by Awesome Table를 활용

⑦ 위도/경도 자동 생성



GeocodeA | 주소2

	B	C	D	E	F	
1	주차장명	주소	주소2	Latitude	Longitude	주차
2	초안산근린공원	도봉구 창동 24-1	서울특별시 도봉	37.653383	127.0518135	NW
3	마들스타디움(근	노원구 상계동 7	서울특별시 노원	37.6441596	127.0582998	NW
4	마장동(건물) 공	성동구 마장동 4	서울특별시 성동	37.5699389	127.0351354	NW
5	영등포여고 공영	영등포구 신길동	서울특별시 영등	37.5152319	126.9152918	NW
6	당산근린공원 공	영등포구 당산동	서울특별시 영등	37.525511	126.8957693	NW
7	대림운동장(구)	영등포구 대림동	서울특별시 영등	37.4996894	126.8947038	NW
8	회경마을 공영주	동대문구 회경동	서울특별시 동대	37.5880454	127.0580995	NW
9	중산공영주차장	동대문구 답십리	서울특별시 동대	37.568718	127.0588801	NW
10	방학동 도개비시	도봉구 방학동 6	서울특별시 도봉	37.6646276	127.0338865	NW
11	은평평화공원(구	은평구 녹번동 1	서울특별시 은평	37.6053279	126.9231157	NW
12	압구정 428 공영	강남구 압구정동	서울특별시 강남	37.5270132	127.026458	NW
13	신림동공영주차	관악구 신림동 1	서울특별시 관악	37.4873624	126.9234861	NW
14	포이초교 공영주	강남구 개포동 1	서울특별시 강남	37.4755995	127.0523238	NW

Current sheet: 서울시 공영주차장 안내 정보

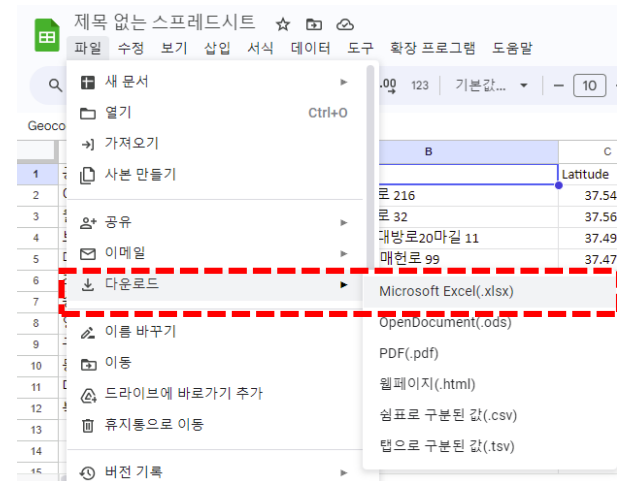
Address column: 주소2

Are your addresses in multiple columns?

Please wait...

Proudly brought to you by [Awesome Table](#)

⑧ 파일 다운로드



File menu options:

- 새 문서
- 열기
- 가져오기
- 다른 만들기
- 공유
- 이메일
- 다운로드
- 이름 바꾸기
- 이동
- 드라이브에 바로가기 추가
- 휴지통으로 이동
- 버전 기록

Download options:

- Microsoft Excel(.xlsx)
- OpenDocument(.ods)
- PDF(.pdf)
- 웹페이지(.html)
- 실제로 구분된 값(.csv)
- 탭으로 구분된 값(.tsv)

1.데이터 읽어오기

❖서울시 공영주차장 파일 읽어오기

```
1 #서울시 공영주차장 안내 정보(경도위도 포함).xlsx 화일을 읽고 상위 5개 데이터 확인하기
2 #변수명 = pd.read_excel('파일경로명')
3 #변수명.head()
4 import pandas as pd
5
6 data = pd.read_excel('/content/서울시 공영주차장 안내 정보(경도위도 포함).xlsx')
7 data.head()
```

	주차장코드	주차장명	주소	주차장종류	주차장종류명	운영구분	운영구분명	전화번호	주차현황정보제공여부	주차현황정보제공여부명	...	기본주차시간(분단위)	추가단위요금	추가시간(분단위)	버스기본주차요금	버스기본주차시간(분단위)	버스추가단위요금	버스추가시간(분단위)	일최대요금	위도	경도
0	1668761	공유 15-10(구)	서울특별시 성동구 성수동2가 843-0	NS	노상주차장	3	시간제 + 거주자 주차장	02-2204-7970	1	현재~20분 이내 연계데이터 존재(현재 주차대수 표현)	...	30.0	900.0	30.0	0.0	0.0	0.0	0.0	10000.0	3.754099e+10	127.050764
1	1372873	도봉산역 공영주차장(시)	서울특별시 도봉구 도봉동 288-19	NW	노외주차장	1	시간제 주차장	02)2290-6047	1	현재~20분 이내 연계데이터 존재(현재 주차대수 표현)	...	5.0	170.0	5.0	0.0	0.0	0.0	0.0	0.0	3.768972e+01	127.044839
2	173831	도봉산 공영주차장(시)	서울특별시 도봉구 도봉동 282-26	NW	노외주차장	1	시간제 주차장	070-7119-0059	1	현재~20분 이내 연계데이터 존재(현재 주차대수 표현)	...	5.0	170.0	5.0	0.0	0.0	0.0	0.0	0.0	3.768764e+01	127.040660
3	1580171	도봉2동 6 공영주차장(구)	서울특별시 도봉구 도봉동 59-1	NW	노외주차장	2	거주자 우선 주차장	02-2091-4215	0	미연계중	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.768304e+01	127.046240
4	1453816	수락산디자인서울거리(구)	서울특별시 노원구 상계동 996-2	NS	노상주차장	1	시간제 주차장	NaN	0	미연계중	...	5.0	0.0	0.0	0.0	0.0	0.0	0.0	19800.0	3.767574e+01	127.056222

5 rows × 39 columns

2. 데이터 정보 확인하기

```
1 #행열개수, 데이터 타입, 열의 개수 등 데이터 정보 확인하기
2 #변수명.info()
3 data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1557 entries, 0 to 1556
Data columns (total 39 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   주차장코드                               1557 non-null   int64
1   주차장명                                 1557 non-null   object
2   주소                                     1557 non-null   object
3   주차장 종류                             1557 non-null   object
4   주차장 종류명                             1557 non-null   object
5   운영구분                                 1557 non-null   int64
6   운영구분명                             1557 non-null   object
7   전화번호                                 838 non-null    object
8   주차현황 정보 제공여부                 1557 non-null   int64
9   주차현황 정보 제공여부명               1557 non-null   object
10  총 주차면                               1380 non-null   float64
11  유무료구분                             1557 non-null   object
12  유무료구분명                             1557 non-null   object
13  야간무료개방여부                       1557 non-null   object
14  야간무료개방여부명                     1557 non-null   object
15  평일 운영 시작시각(HHMM)               1557 non-null   int64
16  평일 운영 종료시각(HHMM)               1557 non-null   int64
17  주말 운영 시작시각(HHMM)               1557 non-null   int64
18  주말 운영 종료시각(HHMM)               1557 non-null   int64
19  공휴일 운영 시작시각(HHMM)             1557 non-null   int64
20  공휴일 운영 종료시각(HHMM)             1557 non-null   int64
21  최종데이터 동기화 시간                 1557 non-null   datetime64[ns]
22  토요일 유, 무료 구분                   1557 non-null   object
23  토요일 유, 무료 구분명                 1557 non-null   object
24  공휴일 유, 무료 구분                   1557 non-null   object
25  공휴일 유, 무료 구분명                 1557 non-null   object
26  월 정기권 금액                         1156 non-null   float64
27  노상 주차장 관리그룹번호               757 non-null    float64
28  기본 주차 요금                         1344 non-null   float64
29  기본 주차 시간(분 단위)                 1343 non-null   float64
30  추가 단위 요금                         1284 non-null   float64
31  추가 단위 시간(분 단위)                 1284 non-null   float64
32  버스 기본 주차 요금                     1354 non-null   float64
33  버스 기본 주차 시간(분 단위)           1354 non-null   float64
34  버스 추가 단위 시간(분 단위)           1353 non-null   float64
35  버스 추가 단위 요금                     1353 non-null   float64
36  일 최대 요금                           1028 non-null   float64
37  위도                                    1557 non-null   float64
38  경도                                    1557 non-null   float64
dtypes: datetime64[ns](1), float64(14), int64(9), object(15)
memory usage: 474.5+ KB
```


3. 지도와 Marker 표시하기

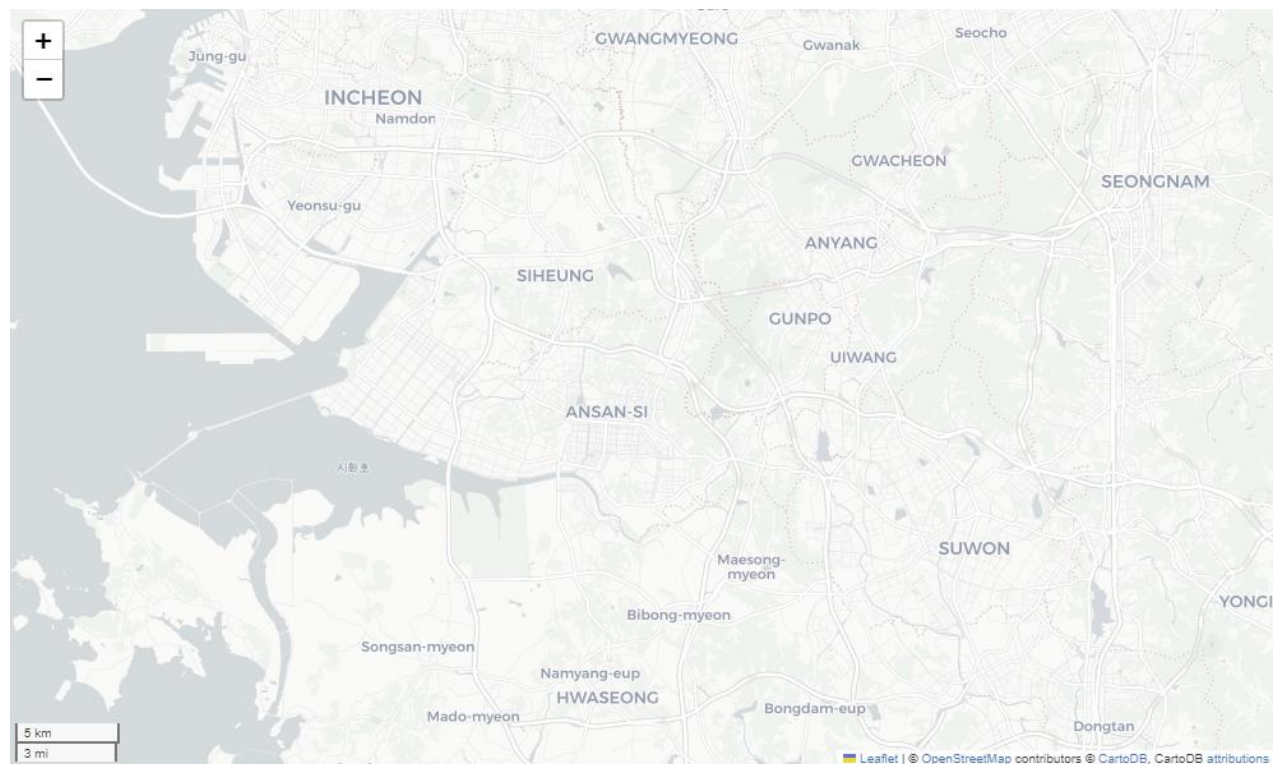
❖ folium 라이브러리 선언하기

```
1 #지도를 생성하기 위한 folium 라이브러리 선언하기
2 #import folium
3 import folium
```

❖ 지도 생성하기

```
parkingMap = folium.Map(location=[37.4965, 126.9572],
                          zoom_start=11,
                          control_scale=True,
                          tiles='cartodbpositron')
```

parkingMap



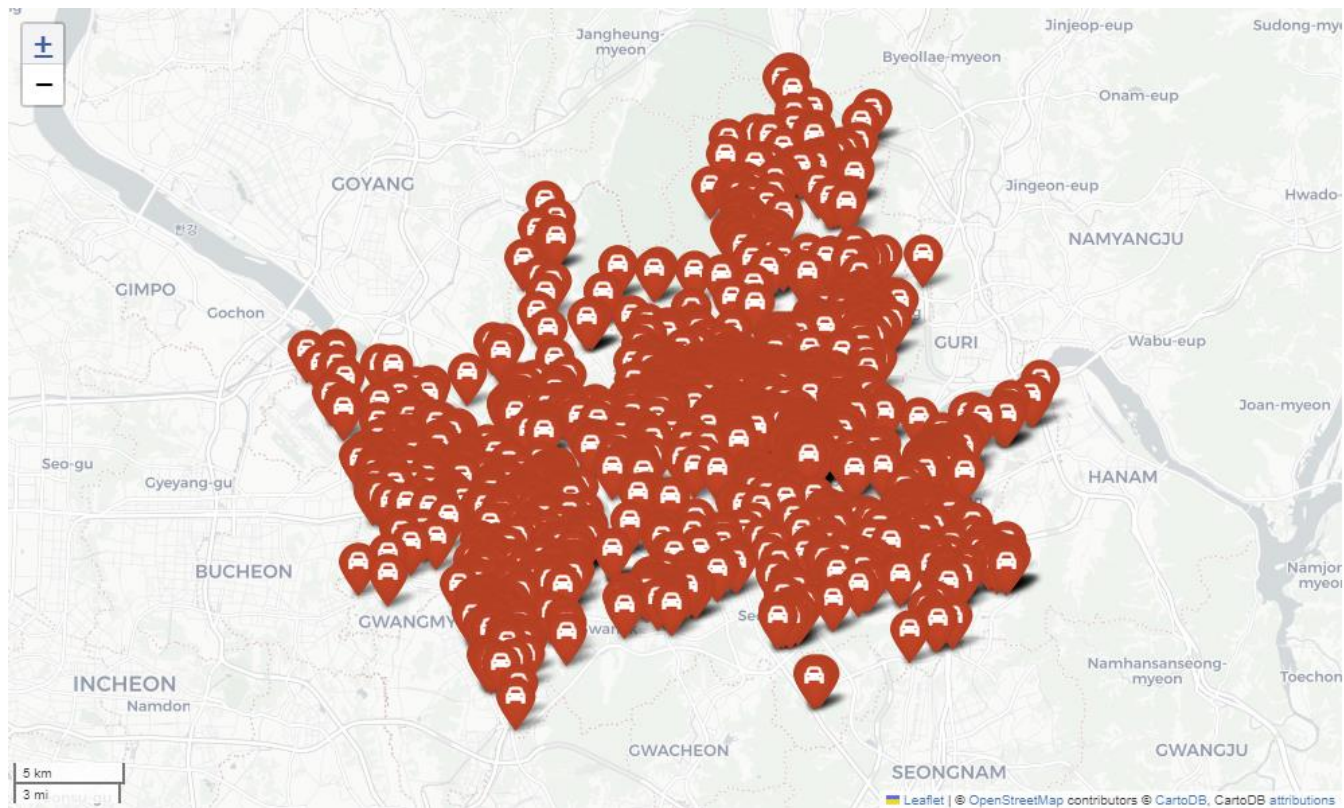
3. 지도와 Marker 표시하기

❖ 지도에 Marker 표시하기

```
for i, row in data.iterrows():
    주차장명 = row['주차장명']
    위도 = row['위도']
    경도 = row['경도']

    mk = folium.Marker([위도, 경도],
                       popup=folium.Popup(주차장명, max_width=200),
                       icon=folium.Icon(color='red', icon='car', prefix='fa'))
    mk.add_to(parkingMap)

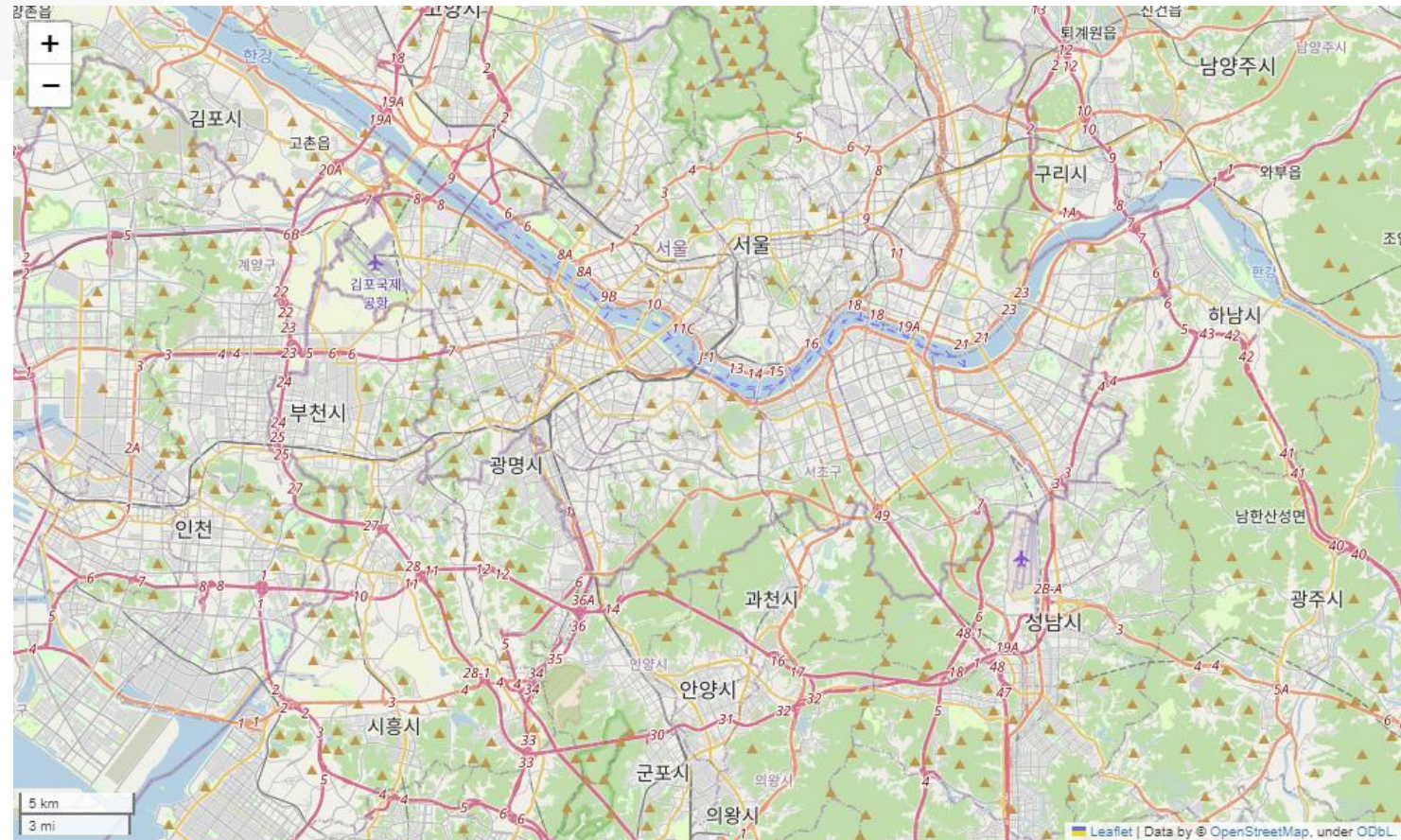
parkingMap
```



4. 지도에 MarkerCluster와 Marker 표시하기

❖ Marker와 MarkerCluster를 표시할 지도 생성하기

```
1 #Marker와 MarkerCluster 를 표시할 지도 생성하기
2 parkingMap2 = folium.Map(location=[37.4965,126.9572],
3                             zoom_start=11,
4                             control_scale=True)
5 parkingMap2
```



4. 지도에 MarkerCluster와 Marker 표시하기

❖ MarkerCluster 생성하고 지도에 추가하기

```
1 #MarkerCluster 라이브러리 선언하기
2 #from folium.plugins import MarkerCluster
3 from folium.plugins import MarkerCluster
4
5 #MarkerCluster 를 생성하고 지도에 추가하기
6 #변수명 = MarkerCluster().add_to(지도변수명)
7 mc = MarkerCluster().add_to(parkingMap2)
```


4. 지도에 MarkerCluster와 Marker 표시하기

❖ Marker들을 생성하고 MarkerCluster에 추가하기

```
1 #folium.Marker(속성들).add_to(MarkerCluster변수명)
2
3 for i, row in data.iterrows():
4     주차장명 = row['주차장명']
5     위도 = row['위도']
6     경도 = row['경도']
7
8     mk = folium.Marker([위도, 경도],
9                        popup=folium.Popup(주차장명, max_width=200))
10    mk.add_to(mc)
11
12 parkingMap2
```

