

Good District to Live in Seoul



LA TROBE
UNIVERSITY

Jiwoo Kwak
Jisoo Kim

1. Introduction

There is a livability score in Australia, but there is no index of this concept in Korea. Therefore, I would like to calculate the score of each autonomous district in Seoul with factors such as transportation, education, parks, and industrial complexes and we will call this 'salubrious score'. With this calculated score, I'm trying to find the best place to live in. And we will also try to find out which factors affect housing prices in Seoul the most.

2. Data

- south korea geofabrik data

<https://www.geofabrik.de/data/download.html>

- seoul district data

<https://data.seoul.go.kr/dataList/10112/S/2/datasetView.do>

- seoul apartement's price data

<http://buking.kr/rank.php?m=mg&si=%EC%84%9C%EC%9A%B8&st=md&gi=3>

- subway number data

<https://www.data.go.kr/data/15081868/fileData.do?recommendDataYn=Y>

so we use the data below:

school → education

commercial(shopping mall, grocery store, cafe, workplace) → food, convenience

park → natural environment

public transport station (bus&subway) → transport

housing price(apartement's price per 3m² by District in Seoul): we usually use "Pyeong" for unit of area and floorspace.

- Data type of the tables

normal_bus_stop / commercial / school / subway	
char	자치구
float	normalized_num
geom	geom

salubrious_score	
char	자치구
float	salubrious_num
geom	geom

seoul_house_price	
char	자치구
float	price
geom	geom

3. Errors and Solutions

When I tried to join the seoul_district table and subway table, one row with 'Gangnam' was dissapeared. So the code below is how I solved this problem. First, I found that there was some difference in the word 'Gangnam' in two tables. It might be the unshown bit problem. So I updated the data.

```
select trim(자치구),length(trim(자치구)) from korea.seoul_district
except
select trim(자치구),length(trim(자치구)) from korea.subway ss;
update korea.subway
set 자치구='Gangnam'
where 자치구 like '%Gangn%';
```

4. Source Code and Result

-- SUBJECT: Calculate the salubrious score by elements(school, commercial, park, bus_stop)
-- and compare with house price

-- 1. create seoul district

-- method: we created seoul district data by grouping data by '자치구' which is the autonomous region

update korea.seoul

```
set wkb_geometry st_transform(st_setsrid(wkb_geometry, 2097), 4326);
```

```
create table korea.seoul_district as
```

```
select s1.자치구, st_union(s1.wkb_geometry) geom
```

from korea.seoul s1

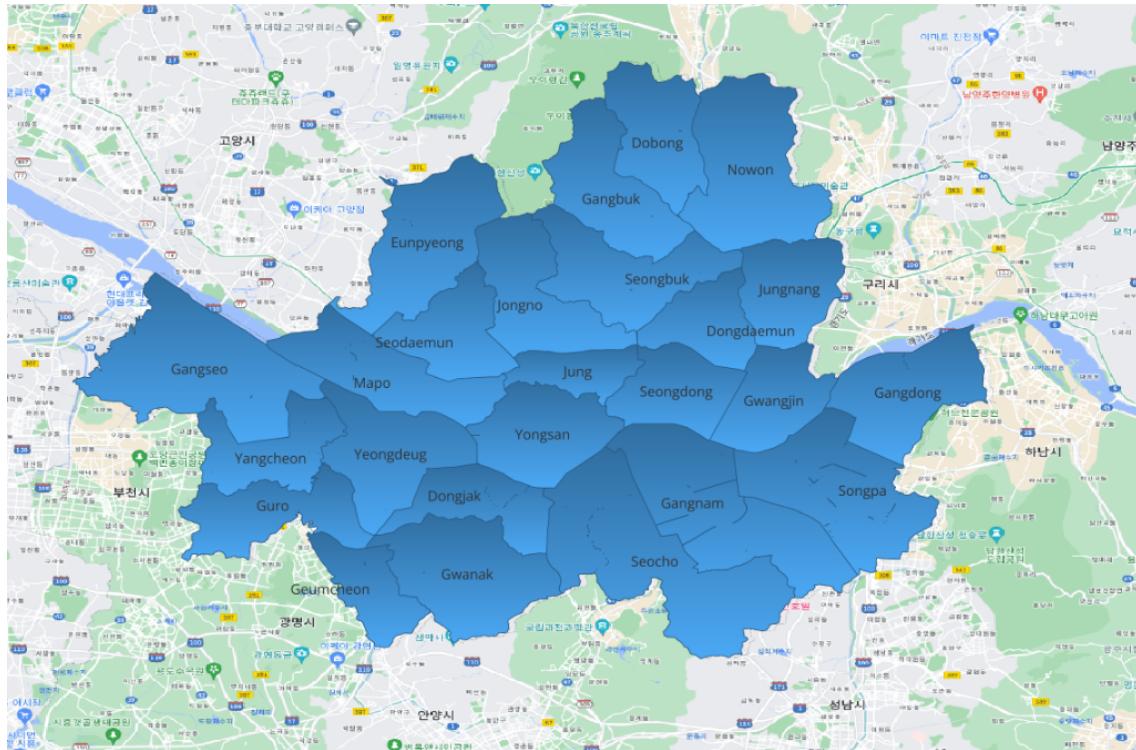
join korea.seoul s2

on s1.자치구s2.자치구

and s1.ogc_fid!= s2.o

group by s1.자치구;

A small map in the bottom right corner showing the area around the school. It includes labels for '서울대학교' (Seoul National University), '한강' (Han River), and '성동구' (Seongdong-gu). A green arrow points towards the school's location.



```
-- 2. SELECT DATA  
-- we used given korea data for school, commercial, park, bus stop  
and we found the data which contains number of subway stations  
-- method: we created table by selecting data which type is school  
from building table
```

```
-- select data which is school
```

```
create table korea.school as
```

```
select *
```

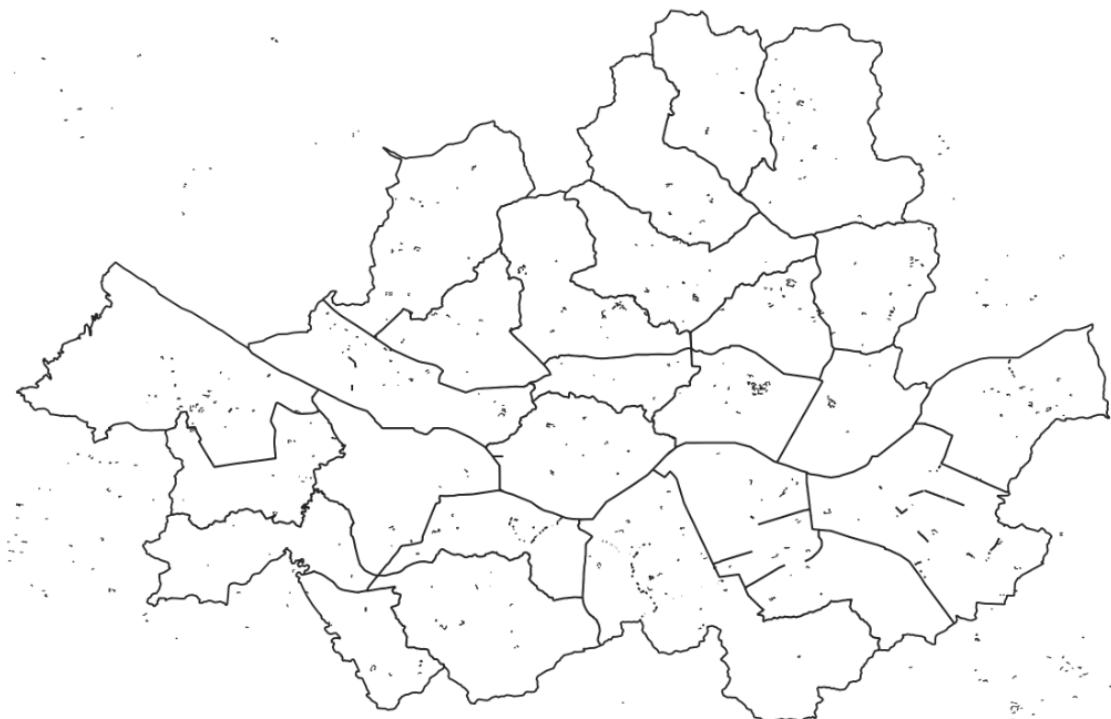
```
from korea.buildings b
```

```
where type='school';
```

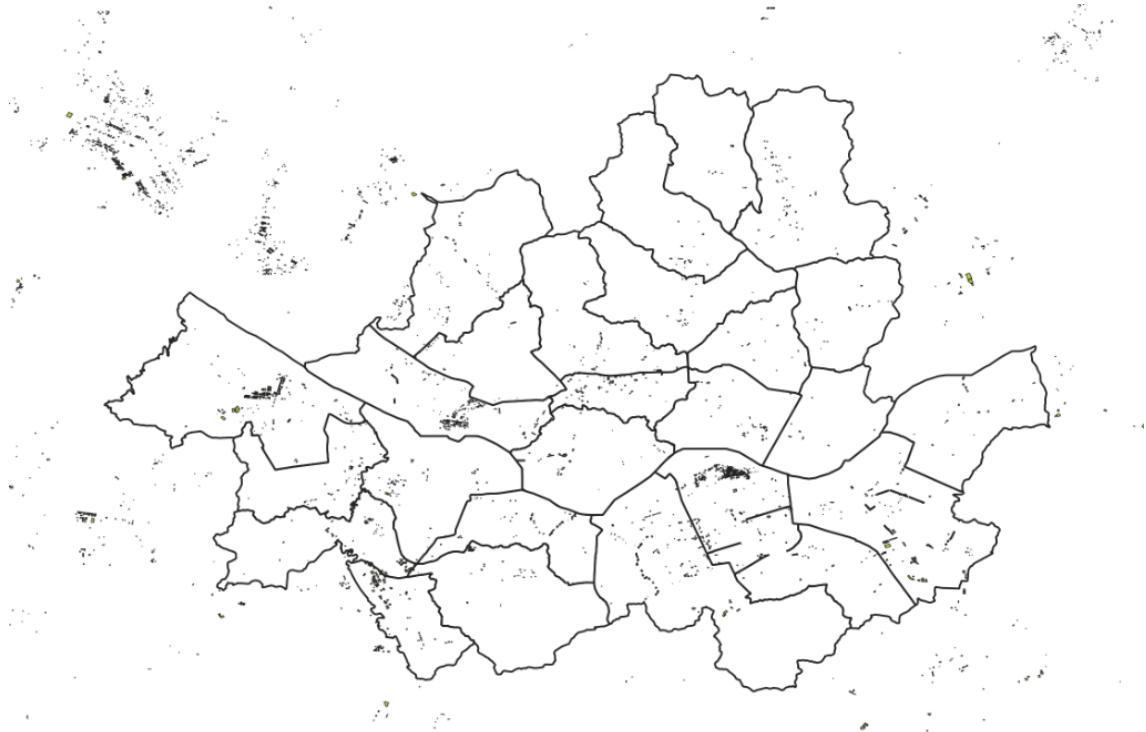
```
DELETE FROM korea.school
```

```
WHERE name IS null
```

```
or type is null;
```



```
-- method: we created table by selecting data which type is  
commercial from building table  
-- select data which is commercial  
create table korea.commercial as  
select *  
from korea.buildings b  
where b.type='commercial';  
DELETE FROM korea.commercial  
WHERE name IS null;
```



```
-- method: we created table by selecting data which fclass is park  
from building table  
-- select data which is park  
create table korea.park as  
select *  
from korea.landuse l  
where l.fclass='park';  
DELETE FROM korea.park  
WHERE name IS null;
```



```
-- method: we created table by selecting data which fclass is bussstop  
from building table  
-- select data which is bus_stop  
create table korea.busstop as  
select *  
from korea.transport t  
where t.fclass='bus_stop';  
DELETE FROM korea.busstop  
WHERE name IS null;
```



```
-- method: we found the data which contains district and number of
subway station per district and it is non-spatial data. So we will join
this table with seoul_district table later.
```

```
-- subway data
```

```
create table korea.subway
(자치구 char(4),
num float);
copy korea.subway(자치구,num)
from '/home/Korea/seoul_subway.csv'
delimiter ',';
select * from korea.subway;
```

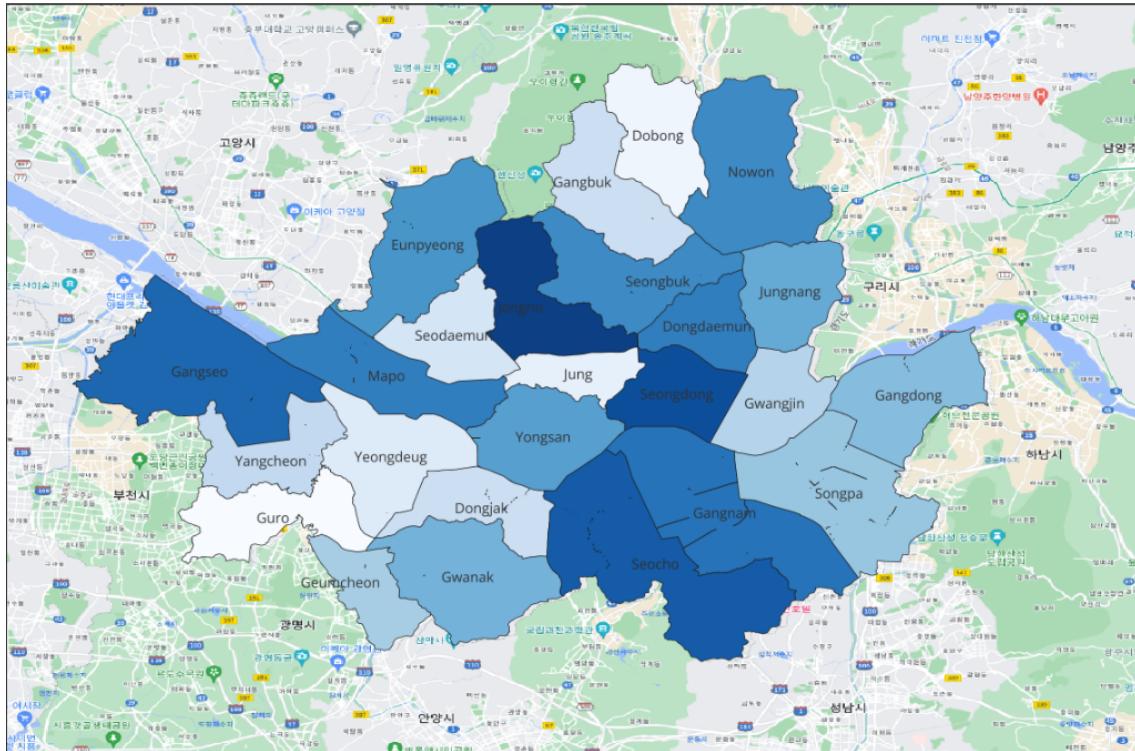
```
create table korea.seoul_subway as
select d.자치구,num, geom
from korea.seoul_district d
join korea.subway s
on s.자치구=d.자치구;
```

-- 3. COUNT NUMBERS

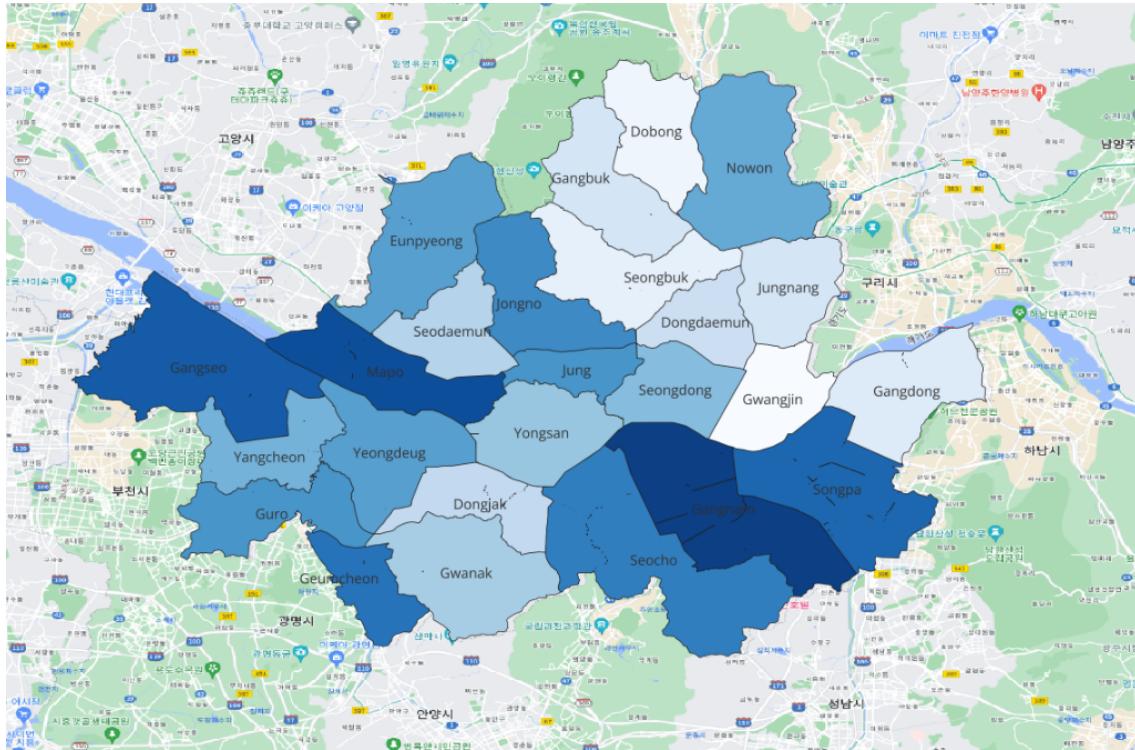
-- method: we created table by joining elements tables and seoul district table to have the geom, and count the elements by grouping with district name

-- count the numbers of school in district

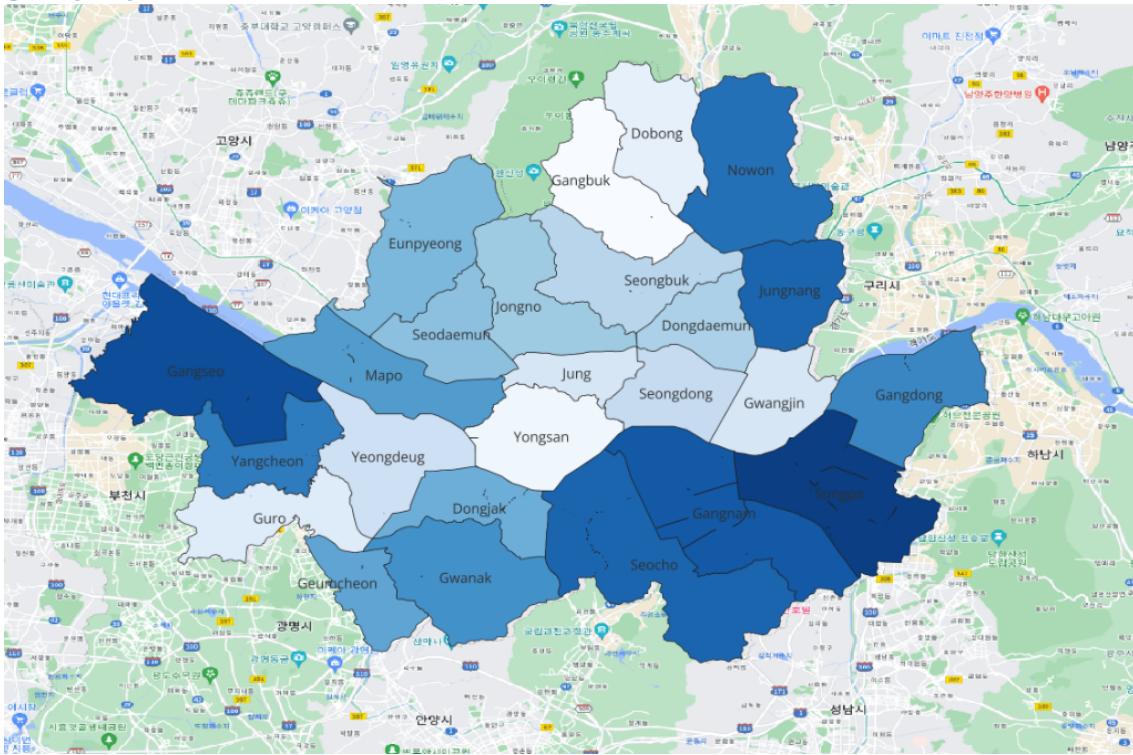
```
create table korea.seoul_school as
with seoul_school as
(select 자치구, st_setsrid(st_union(geom), 4326) geom
from korea.seoul_district sd
group by 자치구)
select sd2.자치구, count(s.ogc_fid) num, sd2.geom
from korea.seoul_district sd2
join korea.school s
on st_contains(sd2.geom, s.wkb_geometry)
group by sd2.자치구, sd2.geom;
```



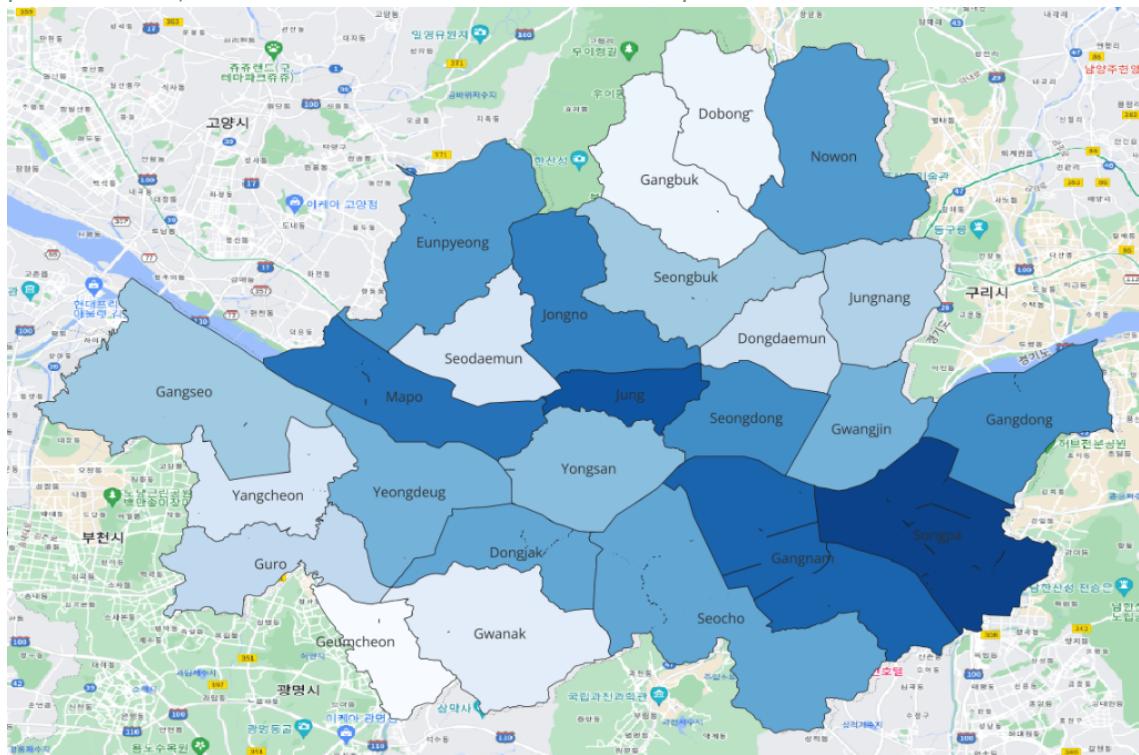
```
-- count the numbers of commercial in district
create table korea.seoul_commercial as
with seoul_commercial as
(select 자치구, st_setsrid(st_union(geom), 4326) geom
from korea.seoul_district sd
group by 자치구)
select sd2.자치구, count(c.ogc_fid) num, sd2.geom
from korea.seoul_district sd2
join korea.commercial c
on st_contains(sd2.geom, c.wkb_geometry)
group by sd2.자치구, sd2.geom;
```



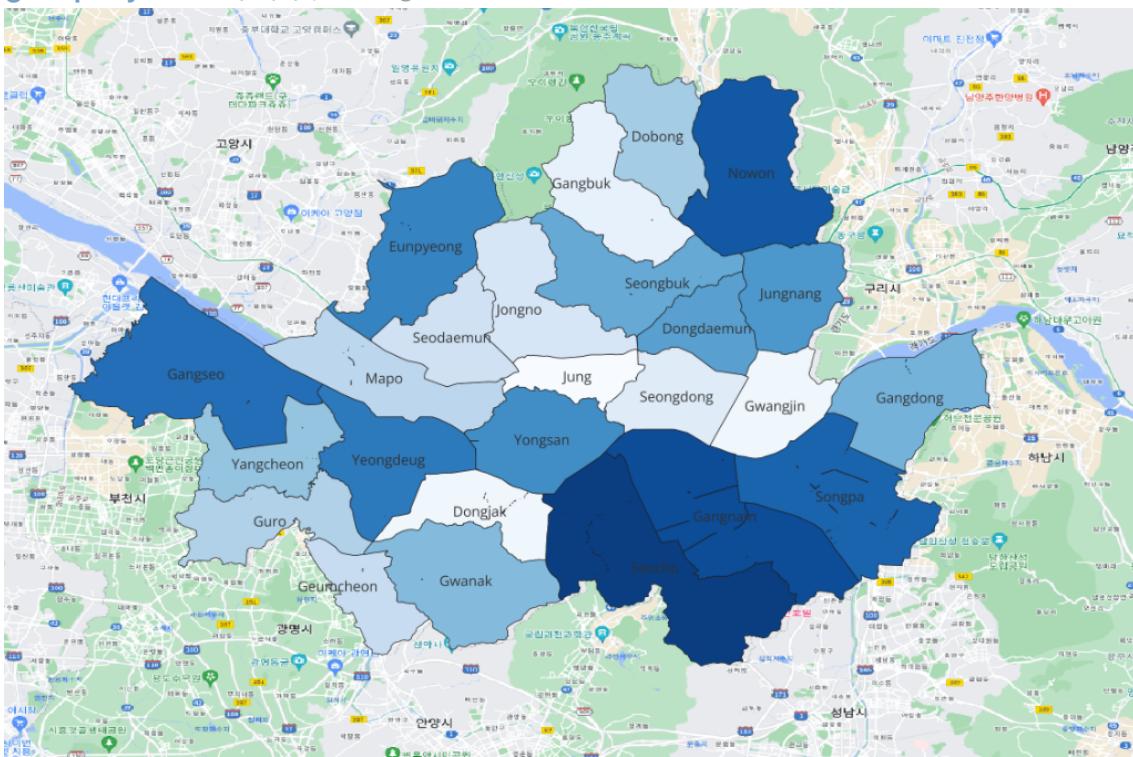
```
-- count the numbers of park in district
create table korea.seoul_park as
with seoul_park as
(select 자치구, st_setsrid(st_union(geom), 4326) geom
from korea.seoul_district sd
group by 자치구)
select sd2.자치구, count(p.ogc_fid) num, sd2.geom
from korea.seoul_district sd2
join korea.park p
on st_contains(sd2.geom, p.wkb_geometry)
group by sd2.자치구, sd2.geom;
```



-- count the numbers of subway in district
-- we have found the data which shows the number of subway station per district, so we do not have to write sql code to count the number



```
-- count the numbers of bus_stop in district
create table korea.seoul_bus_stop as
with seoul_bus_stop as
(select 자치구, st_setsrid(st_union(geom), 4326) geom
from korea.seoul_district sd
group by 자치구)
select sd2.자치구, count(bs.ogc_fid) num, sd2.geom
from korea.seoul_district sd2
join korea.busstop bs
on st_contains(sd2.geom, bs.wkb_geometry)
group by sd2.자치구, sd2.geom;
```



-- NORMALIZE

--method: we used the formula below to normalize the counted data

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

-- normalize school

```
SELECT CAST(num as float) FROM korea.seoul_school ;
```

```
select min(num)
```

```
from korea.seoul_school; --4
```

```
select max(num)
```

```
from korea.seoul_school; --60
```

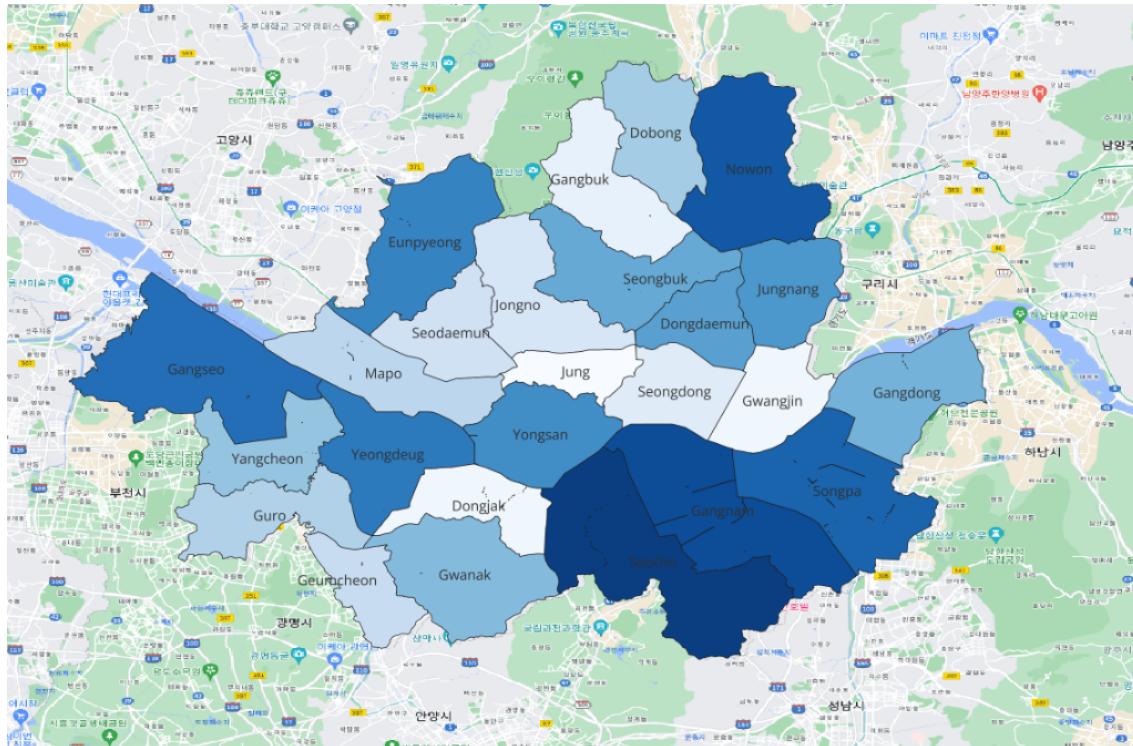
```
create table korea.normal_seoul_school as
```

```
select 자치구, (num-4.0)/56.0 normalized_num, geom
```

```
from korea.seoul_school
```

```
order by normalized_num desc;
```

```
select * from korea.normal_seoul_school;
```



```
-- normalize commercial
```

```
SELECT CAST(num as float) FROM korea.seoul_commercial ;
```

```
select min(num)
```

```
from korea.seoul_commercial; --8
```

```
select max(num)
```

```
from korea.seoul_commercial; --749
```

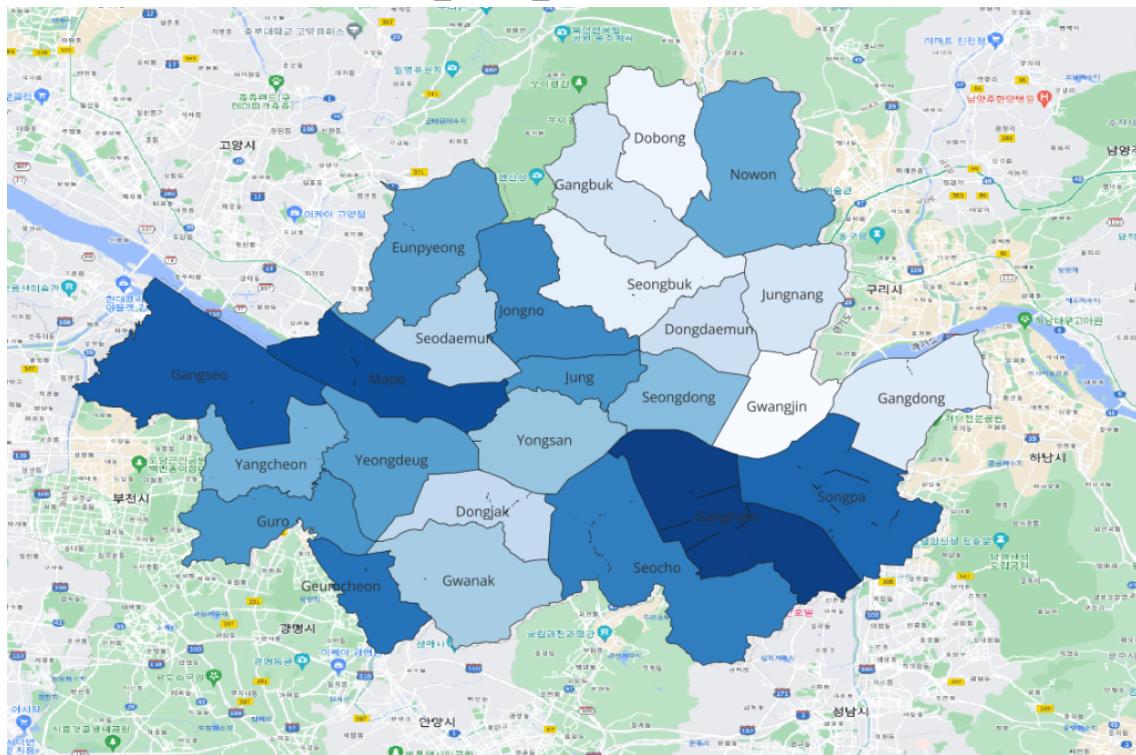
```
create table korea.normal_seoul_commercial as
```

```
select 자치구, (num-8.0)/749.0 normalized_num, geom
```

```
from korea.seoul_commercial
```

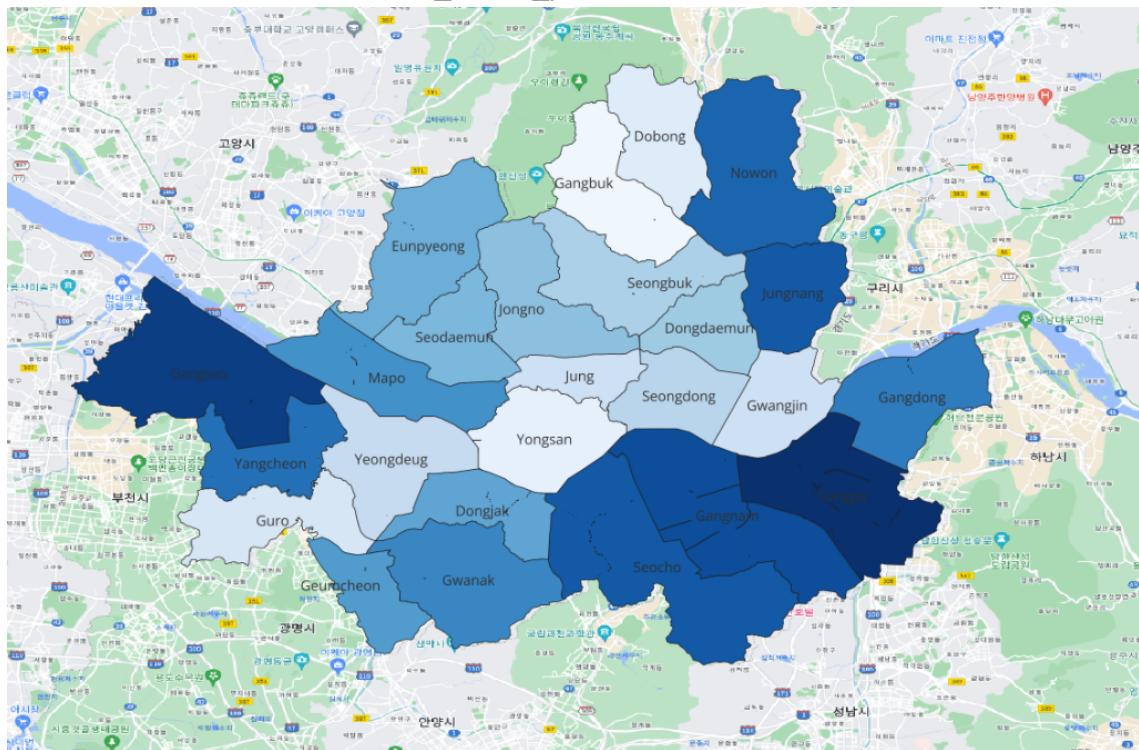
```
order by normalized_num desc;
```

```
select * from korea.normal_seoul_commercial;
```



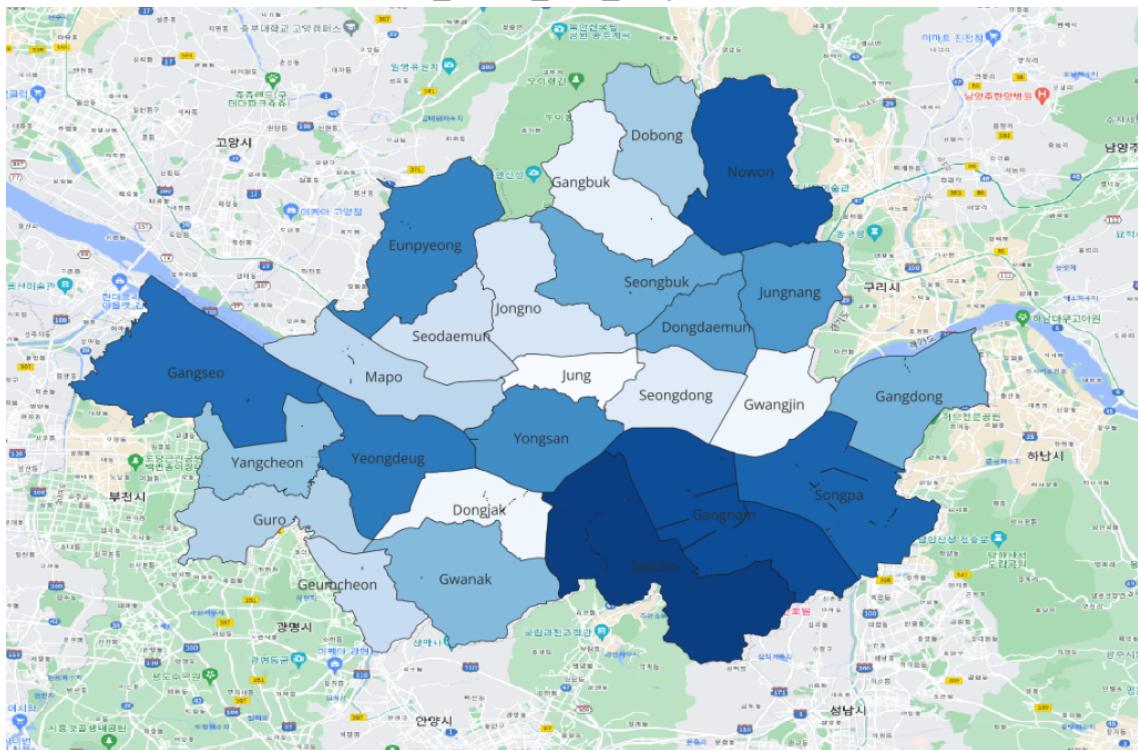
```
-- normalize park
SELECT CAST(num as float) FROM korea.seoul_park;
select min(num)
from korea.seoul_park;--6
select max(num)
from korea.seoul_park; --109
```

```
create table korea.normal_seoul_park as
select 자치구, (num-6.0)/109.0 normalized_num, geom
from korea.seoul_park
order by normalized_num desc;
select * from korea.normal_seoul_park;
```



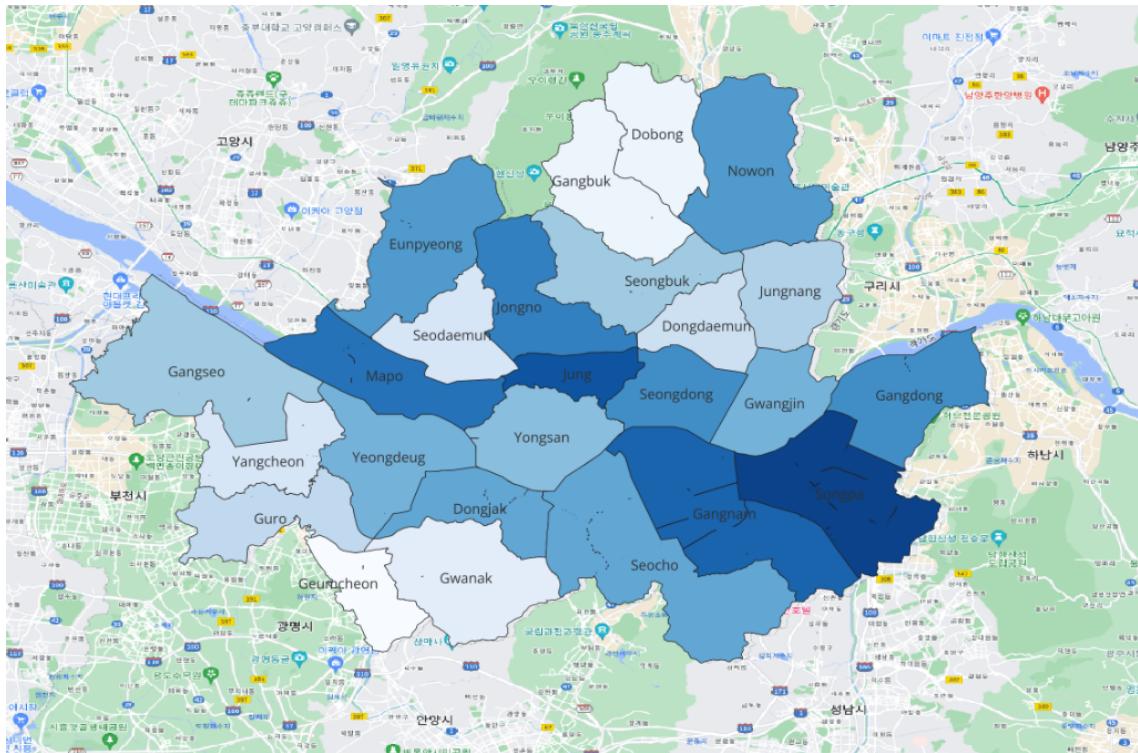
```
-- normalize bus_stop
SELECT CAST(num as float) FROM korea.seoul_bus_stop;
select min(num)
from korea.seoul_bus_stop;--164
select max(num)
from korea.seoul_bus_stop;--630
```

```
create table korea.normal_seoul_bus_stop as
select 자치구, (num-164.0)/630.0 normalized_num, geom
from korea.seoul_bus_stop
order by normalized_num desc;
select * from korea.normal_seoul_bus_stop;
```



```
-- normalize subway
SELECT CAST(num as float) FROM korea.seoul_subway;
select min(num)
from korea.seoul_subway;--1
select max(num)
from korea.seoul_subway;--28
```

```
create table korea.normal_seoul_subway as
select 자치구, (num-1.0)/28.0 normalized_num, geom
from korea.seoul_subway
order by normalized_num desc;
select * from korea.normal_seoul_subway;
```



```
-- JOIN ALL TABLES ON 자치구
create table korea.normalized_elements as
select s.자치구, s.normalized_num nor_school, s.geom
,(select      normalized_num      nor_commercial      from
korea.normal_seoul_commercial as c where c.자치구s.자치구)
,(select normalized_num nor_park from korea.normal_seoul_park as p
where p.자치구s.자치구)
,(select      normalized_num      nor_busstop      from
korea.normal_seoul_bus_stop as b where b.자치구s.자치구)
,(select normalized_num nor_subway from korea.normal_seoul_subway
as sw where sw.자치구s.자치구)
from korea.normal_seoul_school as s;
```

-- ADD ALL THE ELEMENTS

-- method: we add all the normalized element data

create table korea.salubrious_score **as**

select

nor_school+nor_commercial+nor_park+nor_busstop+nor_subway

salubrious _score, geom

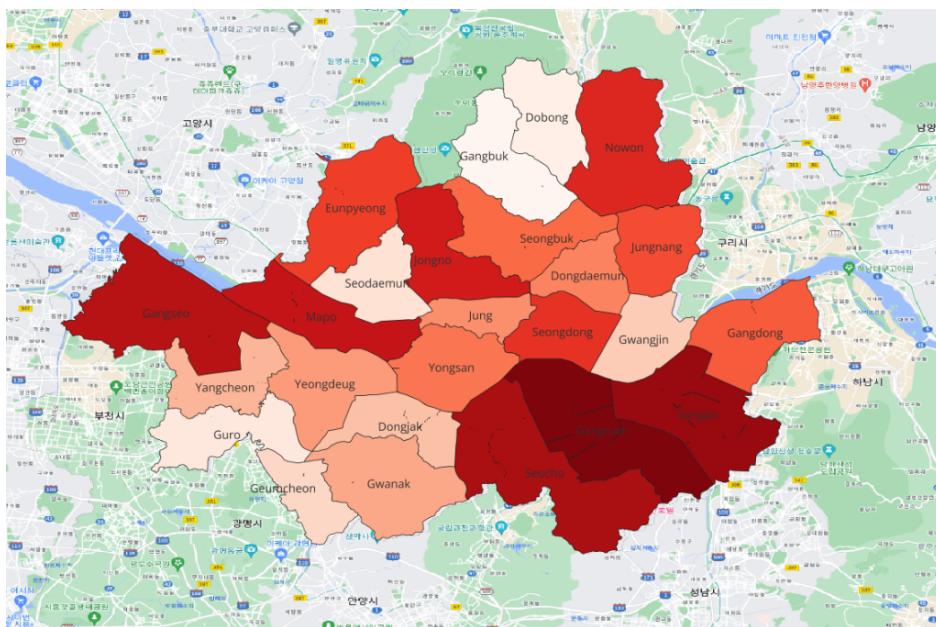
from korea.normalized_elements

order by salubrious_score **desc**

select * from korea.salubrious _score;

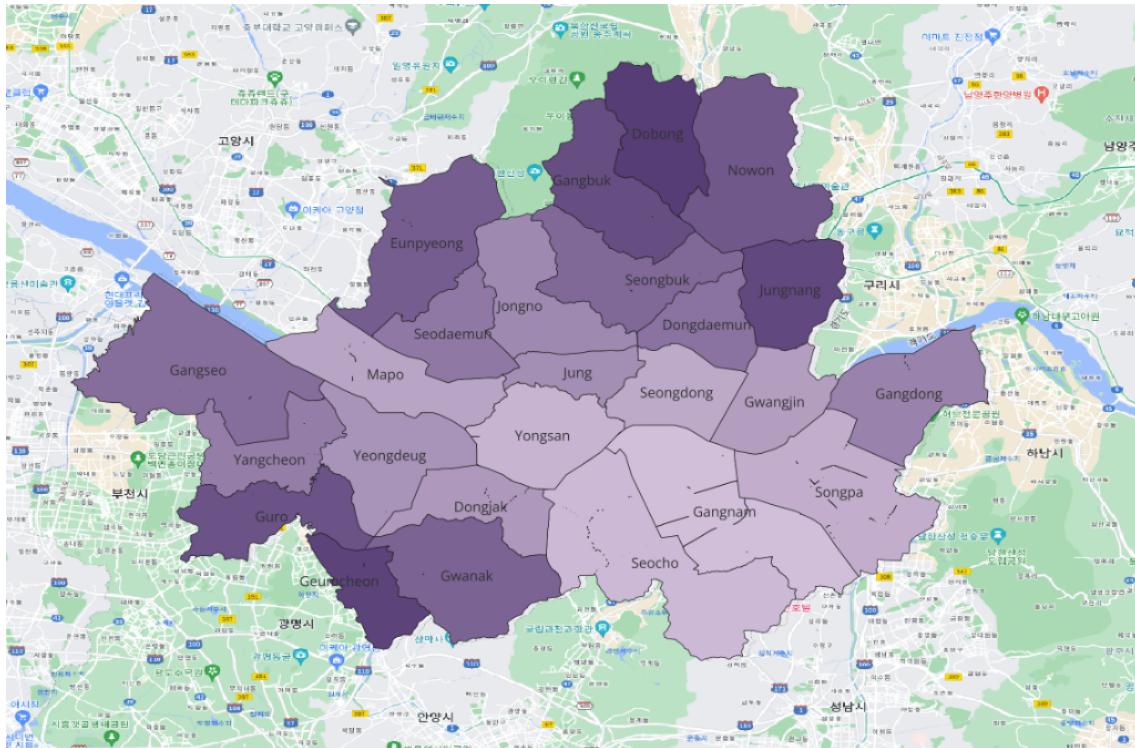
자치구,

1	abc 자치구	123 salubrious_score
1	Gangnam	3.5791581776
2	Songpa	2.6648213588
3	Seocho	2.3132270823
4	Gangseo	2.206639944
5	Mapo	1.974993059
6	Jongno	1.8920387224
7	Nowon	1.8696548847
8	Seongdong	1.6549001455
9	Eunpyeong	1.582498309
10	Jungnang	1.4398663049
11	Gangdong	1.3774485246
12	Seongbuk	1.2881019069
13	Yongsan	1.2668833334
14	Dongdaemun	1.2300347116
15	Jung	1.1232346493
16	Yeongdeug	1.0619289192
17	Gwanak	1.0314295037
18	Yangcheon	0.962282976
19	Dongjak	0.901834318
20	Gwangjin	0.8025811854
21	Geumcheon	0.7201541573
22	Seodaemun	0.6528225878
23	Guro	0.580727004
24	Dobong	0.4130932647
25	Gangbuk	0.2911659991



```
-- HOUSE PRICE
-- method: we joined seoul district data with seoul district data to
have the geom
create table korea.house_price
(자치구 char(10),
price int);
copy korea.house_price(자치구,price)
from '/home/Korea/seoul_house_price.csv'
delimiter ',';

create table korea.seoul_house_price as
select p.자치구, p.price, d.geom
from korea.seoul_district d
join korea.house_price p
on p.자치구=d.자치구
order by price desc;
```



-- SUBJECT 2: Find the element which causes the house price

normal_seoul_bus_stop 1 ×		
Ctrl+click to open SQL console		
id	district	normalized_num
1	Seocho	0.74
2	Gangnam	0.717
3	Nowon	0.451
4	Songpa	0.349
5	Gangseo	0.333
6	Yeongdeug	0.316
7	Eunpyeong	0.308
8	Yongsan	0.297
9	Jungnang	0.286
10	Dongdaemun	0.23
11	Seongbuk	0.217
12	Gangdong	0.186
13	Gwanak	0.184
14	Yangcheon	0.165
15	Dobong	0.111
16	Guro	0.11
17	Mapo	0.081
18	Geumcheon	0.078
19	Seodaemun	0.073
20	Jongno	0.052
21	Seongdong	0.046
22	Gangbuk	0.024
23	Gwangjin	0.021
24	Dongjak	0.021
25	Jung	0

normal_seoul_commercial 1 ×		
Ctrl+click to open SQL console		
id	district	normalized_num
1	Gangnam	0.989
2	Mapo	0.502
3	Gangseo	0.247
4	Songpa	0.103
5	Geumcheon	0.099
6	Seocho	0.097
7	Jongno	0.083
8	Jung	0.073
9	Guro	0.073
10	Eunpyeong	0.071
11	Yeongdeug	0.071
12	Nowon	0.06
13	Yangcheon	0.055
14	Seongdong	0.049
15	Yongsan	0.047
16	Gwanak	0.044
17	Seodaemun	0.028
18	Dongjak	0.025
19	Dongdaemun	0.02
20	Gangbuk	0.017
21	Jungnang	0.017
22	Gangdong	0.012
23	Seongbuk	0.011
24	Dobong	0.003
25	Gwangjin	0

normal_seoul_park 1 ×		
Ctrl+click to open SQL console		
id	district	normalized_num
1	Songpa	0.9449541284
2	Gangseo	0.715963303
3	Gangnam	0.5688073394
4	Seocho	0.4403669725
5	Jungnang	0.4403669725
6	Nowon	0.3944954128
7	Yangcheon	0.3853211009
8	Gangdong	0.376146789
9	Gwanak	0.3211009174
10	Mapo	0.3027522936
11	Geumcheon	0.2935779817
12	Dongjak	0.2844036697
13	Eunpyeong	0.2752293578
14	Seodaemun	0.2660550459
15	Jongno	0.2568807339
16	Dongdaemun	0.247706422
17	Seongbuk	0.2385321101
18	Seongdong	0.2201834862
19	Yeongdeug	0.2110091743
20	Jung	0.1926605505
21	Gwangjin	0.1926605505
22	Guro	0.1834862385
23	Dobong	0.1743119266
24	Yongsan	0.1376146789
25	Gangbuk	0

normal_seoul_subway 1 ×		
Ctrl+click to open SQL console		
id	district	normalized_num
1	Songpa	0.9642857143
2	Jung	0.7857142857
3	Gangnam	0.7142857143
4	Mapo	0.5357142857
5	Jongno	0.5
6	Seongdong	0.4642857143
7	Gangdong	0.4642857143
8	Eunpyeong	0.4285714286
9	Nowon	0.4285714286
10	Dongjak	0.3928571429
11	Seocho	0.3928571429
12	Yeongdeug	0.3571428571
13	Gwangjin	0.3571428571
14	Yongsan	0.3214285714
15	Seongbuk	0.2857142857
16	Gangseo	0.2857142857
17	Jungnang	0.25
18	Guro	0.2142857143
19	Dongdaemun	0.1785714286
20	Yangcheon	0.1428571429
21	Seodaemun	0.1428571429
22	Gwanak	0.1071428571
23	Dobong	0.0714285714
24	Gangbuk	0.0714285714
25	Geumcheon	0

id	district	price
1	Gangnam	5,120
2	Seocho	4,353
3	Yongsan	3,724
4	Songpa	3,470
5	Seongdong	3,441
6	Mapo	3,143
7	Gwangjin	2,922
8	Dongjak	2,858
9	Yeongdeug	2,759
10	Jung	2,644
11	Jongno	2,574
12	Gangdong	2,545
13	Yangcheon	2,471
14	Seodaemun	2,388
15	Dongdaemun	2,239
16	Gangseo	2,156
17	Seongbuk	2,118
18	Gwanak	2,010
19	Eunpyeong	1,906
20	Nowon	1,885
21	Guro	1,812
22	Gangbuk	1,770
23	Jungnang	1,759
24	Geumcheon	1,736
25	Dobong	1,583

5. Conclusion

Commercial is the elements that causes the most to the house price.
 Gangnam, Songpa, Seocho, Gangseo, Mapo is the top 5 district that has the highest livability score.