Melbourne Transportation Project

KANG SEONGYEON

Contents

1. Introduction

We are living in various inconveniences. In this situation, the ability to analyze maps and understand which areas are in what environment is very important. This is because we can find improvements based on it.
In this project, we want to develop the ability to analyze maps through the process of identifying how many bus routes run in each mesh block in Melbourne.

2. Data

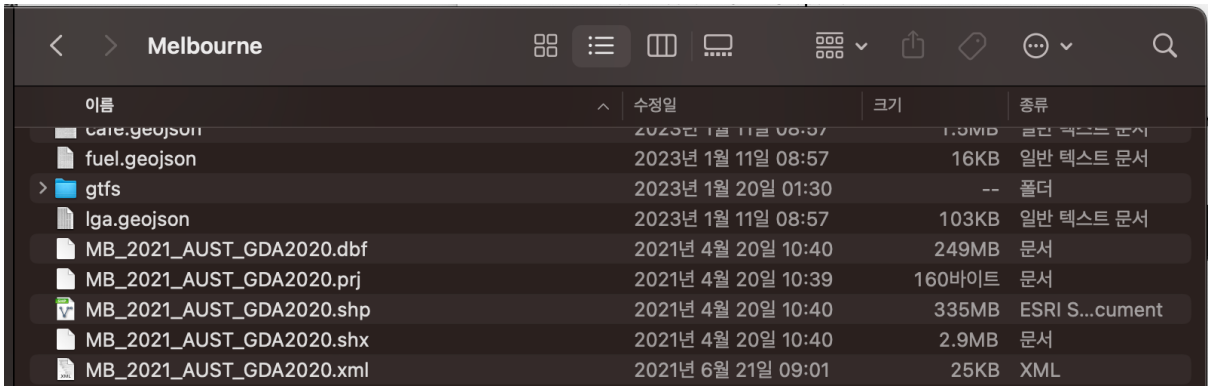
What we need is mesh block data in Melbourne and public transportation data in Melbourne.

2.1. Data source & Insert Data – Mesh block

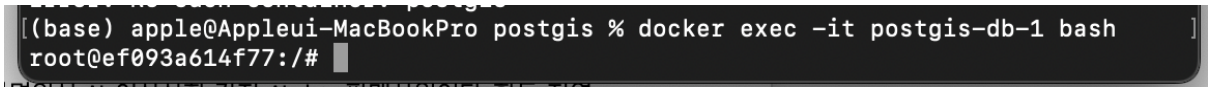
We will get the mesh block data in Australia and process it into the mesh block data in Melbourne in the processing step
Get the Australia meshblock data from this reference
https://www.abs.gov.au/statistics/standards/australian-statistical-geography-standard-asgs-edition-3/jul2021-jun2026/access-and-downloads/digital-boundary-files

Mesh Blocks - 2021 - Shapefile	Download ZIP [217.44 MB]

download this file.
Then Insert the data into the database.
first, put the data(shp file) into the vector folder(which is the home for the docker configuration)
in my case this Melbourne folder is in the vector folder.



make new terminal and execute the docker by
docker exec -it <container_name> bash



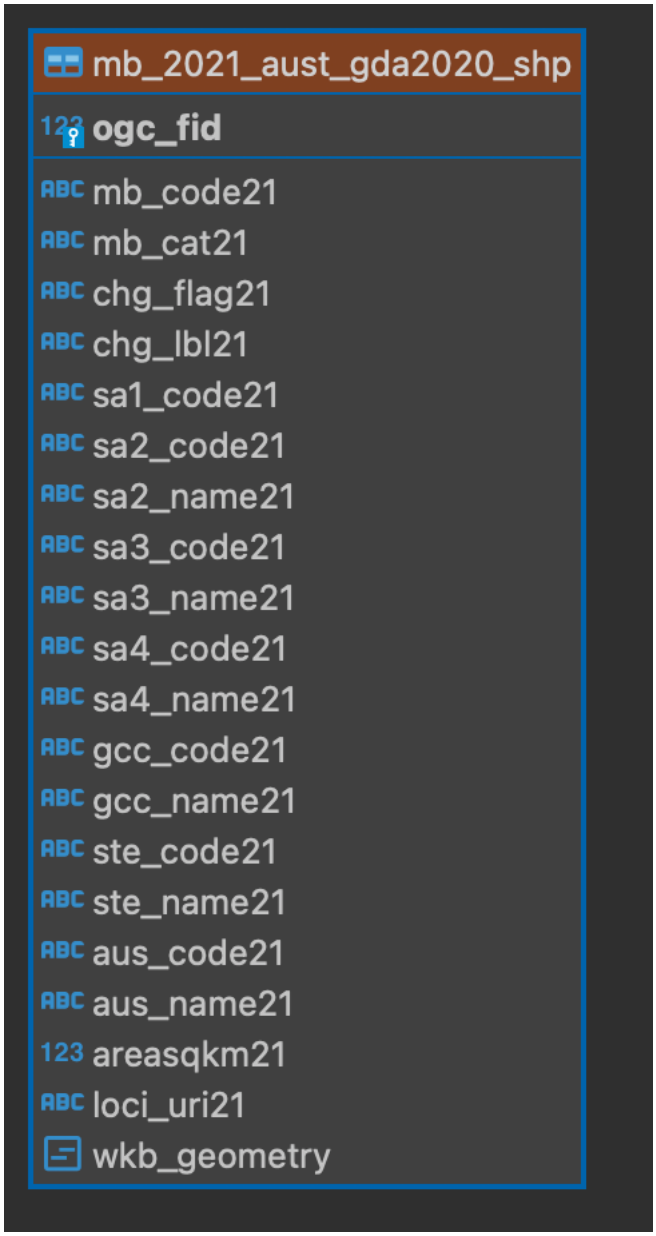
then insert the shp file into the database by ogr2ogr

```
$ ogr2ogr [-f "<Driver>"] PG:"dbname=<db_name> user=<user_name> \
  "<path>/<file_name>" -nln <schema>.<table_name> -overwrite|-append
```

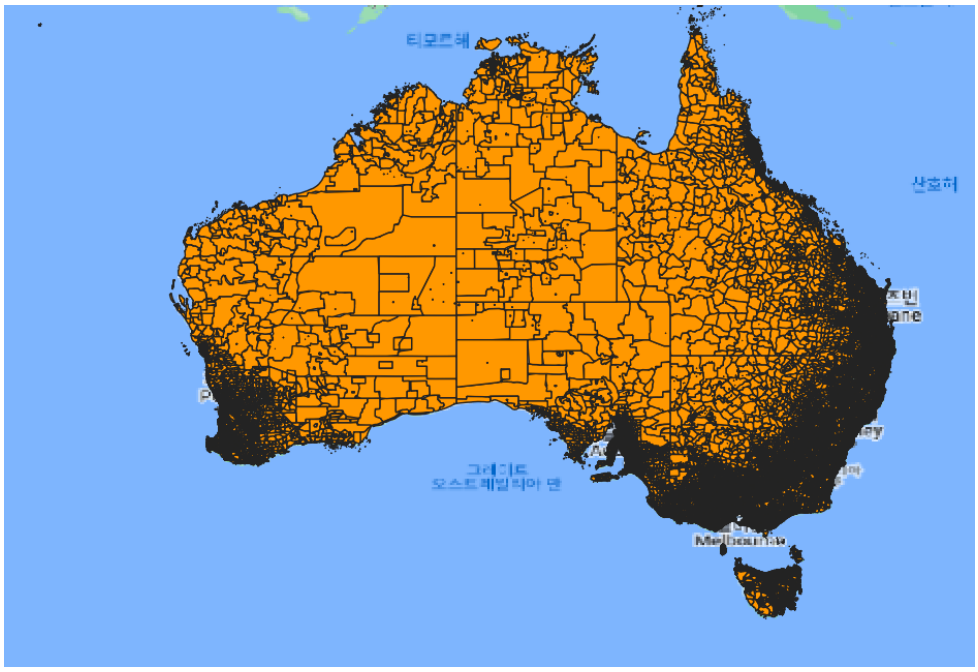
in my case

```
root@ef093a614f77:/# ogr2ogr PG:"dbname=gisdb user=postgres" "/home/Melbourne/MB
_2021_AUST_GDA2020.shp" -nln melbourne.MB_2021_AUST_GDA2020_shp -overwrite
root@ef093a614f77:/#
```

Then we can find this Australia meshblock data table in the dbeaver



368,286 rows in this table.



this is the result that visualize the spatial data in that table by QGIS

2.2. Data source & Insert Data - Transportation

Now we need the data about Melbourne transportation.

Get the Melbourne transportation data from this reference

https://discover.data.vic.gov.au/dataset/ptv-timetable-and-geographic-information-2015-gtfs

Data and Resources

PTV Timetable and Geographic Information - GTFS Explore

Download above.

you can read about the data in PTV GTFS Release Notes

http://data.ptv.vic.gov.au/downloads/PTVGTFSReleaseNotes.pdf

This is the detail of the data in GTFS release.

GTFS Release

The PTV GTFS data has been exported by operational branches listed in the folder numbers below:

1 - Regional Train
2 - Metropolitan Train
3 - Metropolitan Tram
4 - Metropolitan Bus

PUBLIC TRANSPORT VICTORIA PT

5 - Regional Coach
6 - Regional Bus
7 - TeleBus
8 – Night Bus
10 - Interstate
11 - SkyBus

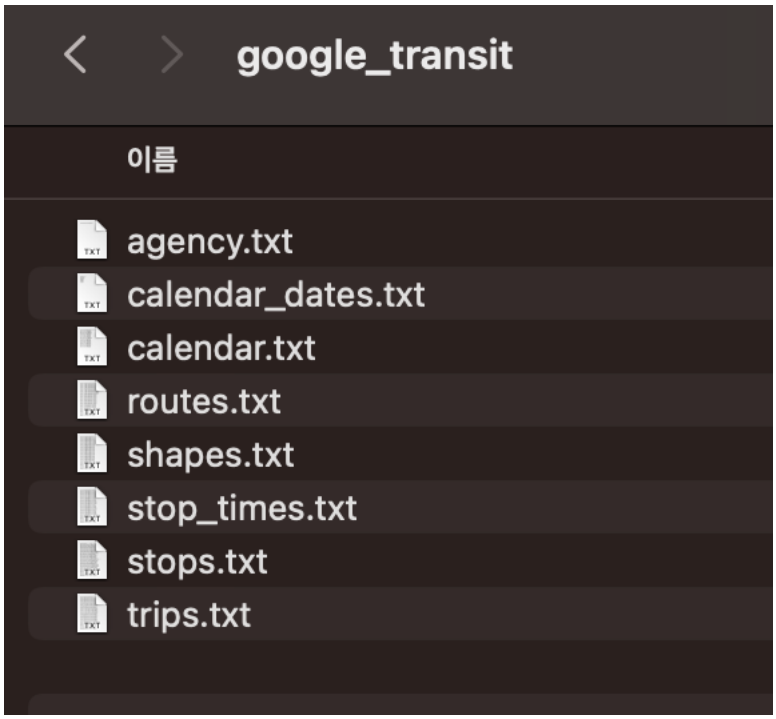
The GTFS data provided for each of the 10 Operational branches is in the form of 8 files and is described in the following table:

GTFS File	**Fields**
agency.txt	agency_id, agency_name, agency_url, agency_timezone, agency_lang
calendar.txt	service_id, monday, tuesday, wednesday, thursday, friday, saturday, sunday, start_date, end_date
calendar_dates.txt	service_id ,date, exception_type
routes.txt	route_id, agency_id, route_short_name, route_long_name, route_type
trips.txt	route_id, service_id, trip_id, shape_id, trip_headsign, direction_id
stops.txt	stop_id, stop_name, stop_lat, stop_lon
stop_times.txt	trip_id, arrival_time, departure_time, stop_id, stop_sequence, stop_headsign, pickup_type, drop_off_type, shape_dist_traveled
shapes.txt	shape_id, shape_pt_lat, shape_pt_lon, shape_pt_sequence, shape_dist_traveled

Then we will get this file

1~11 are different condition of transportation.

each folder contains a file in this format. At this time, 7 to 9 have no content.



we should insert the data into the database but in this case we cannot use ogr2ogr because the file format is .txt.

so, we will make a table in the DB and copy the data into the table

It is the same for all folders, so only folder 1 will be described

first, make the whole table.

```
--1
create table melbourne.calendar_dates_1
(service_id varchar(20),
date date,
exection_type char(1));

create table melbourne.calendar_1
(service_id varchar(20),
monday varchar(1),
tuesday varchar(1),
wednesday varchar(1),
thursday varchar(1),
friday varchar(1),
saturday varchar(1),
sunday varchar(1),
start_date date,
end_date date);

create table melbourne.routes_1
(route_id varchar(20),
agency_id varchar(5),
route_short_name varchar(20),
route_long_name varchar(100),
route_type varchar(5),
route_color varchar(10),
route_text_color varchar(10));

create table melbourne.shapes_1
(shape_id varchar(30),
shape_pt_lat float,
shape_pt_lon float,
shape_pt_sequence int,
shape_dist_traveled float);

create table melbourne.stop_times_1
(trip_id varchar(50),
arrival_time char(8),
departure_time char(8),
stop_id varchar(10),
stop_sequence int,
stop_headsign varchar(5),
pickup_type int,
drop_off_type int,
shape_dist_traveled varchar(20));

create table melbourne.stops_1
(stop_id char(5),
stop_name varchar(100),
stop_lat float,
stop_lon float);

create table melbourne.trips_1
(route_id varchar(20),
service_id varchar(20),
```

second, copy the data from the txt file into the table.

```
59
60  copy melbourne.calendar_dates_1
61  from '/home/Melbourne/gtfs/1/google_transit/calendar_dates.txt'
62  delimiter ','
63  csv header;
64
65  copy melbourne.calendar_1
66  from '/home/Melbourne/gtfs/1/google_transit/calendar.txt'
67  delimiter ','
68  csv header;
69
70  copy melbourne.routes_1
71  from '/home/Melbourne/gtfs/1/google_transit/routes.txt'
72  delimiter ','
73  csv header;
74
75  copy melbourne.shapes_1
76  from '/home/Melbourne/gtfs/1/google_transit/shapes.txt'
77  delimiter ','
78  csv header;
79
80  copy melbourne.stop_times_1
81  from '/home/Melbourne/gtfs/1/google_transit/stop_times.txt'
82  delimiter ','
83  csv header;
84
85  copy melbourne.stops_1
86  from '/home/Melbourne/gtfs/1/google_transit/stops.txt'
87  delimiter ','
88  csv header;
89
90  copy melbourne.trips_1
91  from '/home/Melbourne/gtfs/1/google_transit/trips.txt'
92  delimiter ','
93  csv header;
94
```

after insert 1~11 folder into the table union it so that we can get the total table.

```
/*
 * union the tables so that we could get the total tables of
 * calendar_date
 * calendar
 * routes
 * shapes
 * stop_times
 * stops
 * trips
 */
--calendar_date_total
create table melbourne.calendar_dates_total as
select *
from melbourne.calendar_dates_1
union
select *
from melbourne.calendar_dates_2
union
select *
from melbourne.calendar_dates_3
union
select *
from melbourne.calendar_dates_4
union
select *
from melbourne.calendar_dates_5
union
select *
from melbourne.calendar_dates_6
union
select *
from melbourne.calendar_dates_10
union
select *
from melbourne.calendar_dates_11
```

then if there is a column about latitude and longitude make it into the geometry

```
--make a lon and lat geomtry by update
update melbourne.stops_total
set geom = st_setsrid(st_makepoint(stop_lon, stop_lat),4326);
```

then we have the total table from the GTFS data

3. Processing the data

3.1. Meshblocks

We check that Australia meshblocks are 368,286 rows with the 21 columns.

It's too big and we don't need to use this whole data for our project.

So, get the mesh block data in Melbourne.

In this case I changed the name of the table

MB_2021_AUST_GDA2020_shp -> aus_meshblocks_shp

```
-- STEP 1 : get the mesh block data in melbourne form the mesh block data in australia
-- 59,483
create table melbourne_homework.melb_mb as
select *
from melbourne.aus_meshblocks_shp ams
where gcc_name21 like '%Melb%';
```

then it becomes to 59,483 rows.

3.2. Transportation

Likewise, we don't need to use whole transportation data for our project.

so, filtering the stops in the Melbourne Meshblock

for this we need 2 steps

```
-- STEP 2A : union all geometries in melborune meshblocks : 2m
 select st_union(wkb_geometry) geom
 from melbourne_homework.melb_mb;

-- STEP 2B : meke 500m buffer for union meshblock : 2m 5s
 select st_buffer(st_union(wkb_geometry)::geography,500) geom_buff
 from melbourne_homework.melb_mb;
```

union the mesh blocks in the mebloume and make a 500m buffer for union meshblock so that we can get the stops in the melbourne meshblocks

then make that 2 step as a table.

```
create table melbourne_homework.melb_buff as
select st_buffer(st_union(wkb_geometry)::geography,500)::geometry geom_buff,
       st_union(wkb_geometry) geom
from melbourne_homework.melb_mb;
```

Make the table that filtering the stops as said above

```
-- Step 3: Filter the stops in Melbourne
-- 4s
-- 20,658
create table melbourne_homework.stops_melb as
select distinct s.stop_id , s.geom
from melbourne.stops_total s
join melbourne_homework.melb_buff m
on st_contains(m.geom_buff,s.geom);
```

Our project objective is identifying how many bus routes run in each mesh block in Melbourne.

but, there is no information about the different bus numbers in stops_total table.

컬럼명	#	Data type	Identity	Collation	Not Null
stop_id	1	bpchar(5)		default	[ ]
stop_name	2	varchar(100)		default	[ ]
stop_lat	3	float8			[ ]
stop_lon	4	float8			[ ]
geom	5	geometry(point,			[ ]

different bus numbers in the routes_total table.(routes_short_name is it)

컬럼명	#	Data type	Identity	Collation	Not Null	De
route_id	1	varchar(20)		default	[ ]	
agency_id	2	varchar(5)		default	[ ]	
route_short_	3	varchar(20)		default	[ ]	
route_long_n	4	varchar(100)		default	[ ]	
route_type	5	varchar(5)		default	[ ]	
route_color	6	varchar(10)		default	[ ]	
route_text_c	7	varchar(10)		default	[ ]	

so we should join stops_total table with the routes_total table but there is no common column.

we can solve this problem by joining with other table.

As you can see first, join the routes_total with the trips_total(common column = route_id)

second, join that with the stop_times_total(common column = trip_id)

third, join that with the stops_melb(table that filtering the stops in Melbourne)

Then we can get the table that stop_id with the route_short_name(different bus numbers).

```
-- Step 4 : all routes-stops in area from step 3
-- 10,848,013 (for mine 10,853,806)
create table melbourne_homework.routes_stops as
select r.route_short_name, r.route_long_name,
       r.route_type, t.trip_id, s.stop_id
from melbourne.routes_total r
join melbourne.trips_total t
on (r.route_id=t.route_id)
join melbourne.stop_times_total st
on (t.trip_id = st.trip_id)
join melbourne_homework.stops_melb s
on (s.stop_id=st.stop_id);
```

The combination data of the stop_id and the bus_number may overlap. Our purpose is to count the number of different routes, so we eliminate duplication.

in addition to that, make the table that drop the useless columns.

```
--step 5: routes, stops(eliminate duplication) and drop the useless columns
-- 31,134 recs(31,151) -> 30s
create table melbourne_homework.stops_routes as
select distinct rs.route_short_name, rs.route_long_name, rs.stop_id
from melbourne_homework.routes_stops rs;
```

Then we can count how many routes for each stops

```
-- step 6: count how many routes for each stop : 386ms
-- 20,640
create table melbourne_homework.stops_num as
select stop_id, count(*) num
from melbourne_homework.stops_routes
group by stop_id;
```

This is the table.

	stop_id	num
1	7266	2
2	37722	1
3	18782	1
4	38682	2
5	12471	1
6	22565	1
7	19895	2
8	7457	1
9	11270	2
10	12671	2
11	12533	2
12	16692	2
13	48181	1
14	20691	2
15	45445	1
16	50057	1
17	247	2
18	12377	1
19	2275	1
20	15967	1
21	21930	1
22	46227	1

Create the buffer for residential meshblocks

```
-- step 7: create buffer for residential meshblocks
-- 59,483 -> 46,608 in 49.524s
create table melbourne_homework.melb_mb_buff as
select *, st_buffer(mm.wkb_geometry::geography,500)::geometry geom_buff
from melbourne_homework.melb_mb mm
where mb_cat21 ='Residential';
```

Now, we know how many routes each stop has. We want to connect the information with which mesh block the stop belongs -> that way, we can find the number of routes for each meshblock

so, make a table with stop and mesh block connected

```
-- Step 8: get the stops in the meshblocks
-- 612,673 48 secs without index(-> 15m 1s)
-- 612,673 2.2 secs with index(-> 16s)
create table melbourne_homework.mb_stops as
select m.mb_code21 , s.stop_id
from melbourne_homework.melb_mb_buff m
join melbourne_homework.stops_melb s
on st_contains(m.geom_buff,s.geom);
```

It takes pretty long time with my mac(spec : M1, 16GB Memory) without index(15m 1s)

with the indexing it takes short time.(16s)

```
create index on melbourne_homework.stops_melb using gist(geom);
```

Now we can join two tables. One contain the data of stop_id and routes, one contain the data of meshblock_id and stop_id.

```
-- step 9 : get routes in each mb.
-- 925,861 in 15.67s
create table melbourne_homework.mb_routes as
select m.mb_code21, s.route_short_name
from melbourne_homework.mb_stops m
join melbourne_homework.stops_routes s
on (m.stop_id=s.stop_id);
```

Next step is just counting it

You may wonder why use the join when only need to count.

The geometry data of the mesh block is needed for later visualizations.

that's why join the step 9 table with the melb_mb_buff table because it contained that data.

```
-- step 10: aggregate and join with the geometry
-- 44,183 in 17.594s
create table melbourne_homework.mb_routes_num as
select mb.mb_code21 , mb.wkb_geometry , count(mr.route_short_name) num
from melbourne_homework.mb_routes mr
join melbourne_homework.melb_mb_buff mb
on (mr.mb_code21=mb.mb_code21)
group by mb.mb_code21 , mb.wkb_geometry ;
```

so below table is the result of the step 1 to step 10

It became possible to know the number of routes per mesh block

컬럼명	#	Data type	Identity	Collation	Not Null	Default
ABC mb_code21	1	varchar(11)		default	[ ]	
wkb_geomet	2	geometry(polyg			[ ]	
123 num	3	int8			[ ]	

4. Visualization

4.1. QGIS

First, connect the qgis with the dbeaver table.

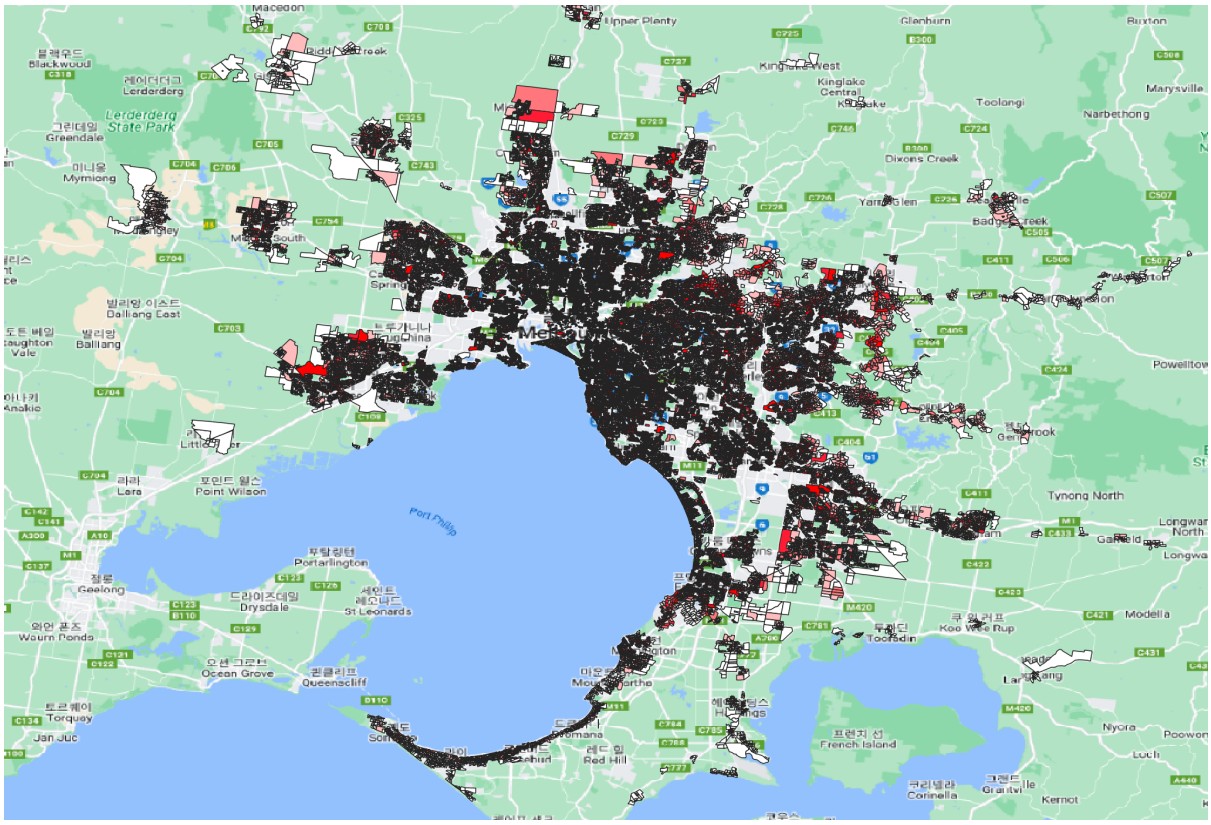
you can use the

https://sites.google.com/view/kikimaulana/home/my-notes/system-installation/qgis-postgis-and-dbeaver?pli=1

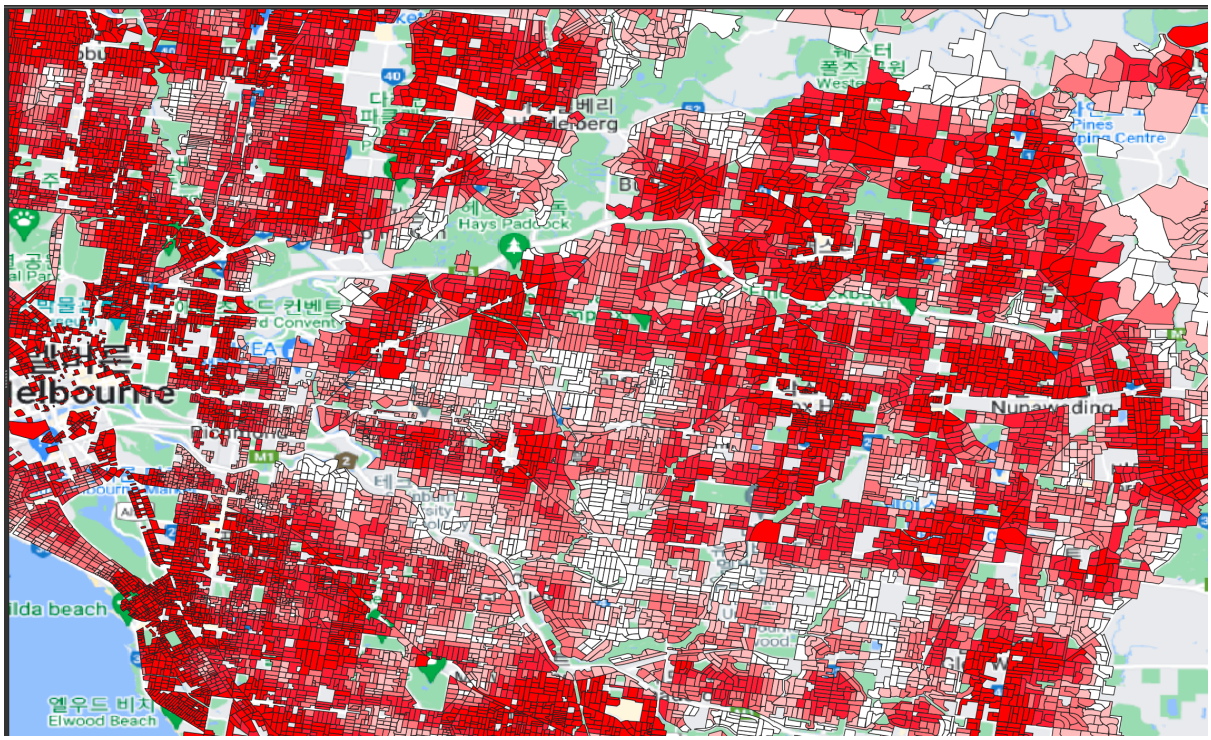
above website's method.

We already ready to make a visualization in the QGIS by the step 10 table.

Near the city there are too much mesh blocks.



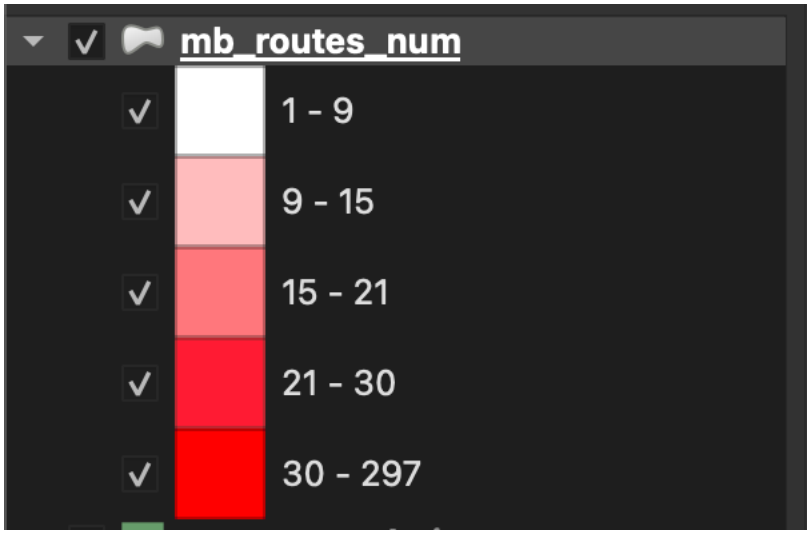
If zoom in,



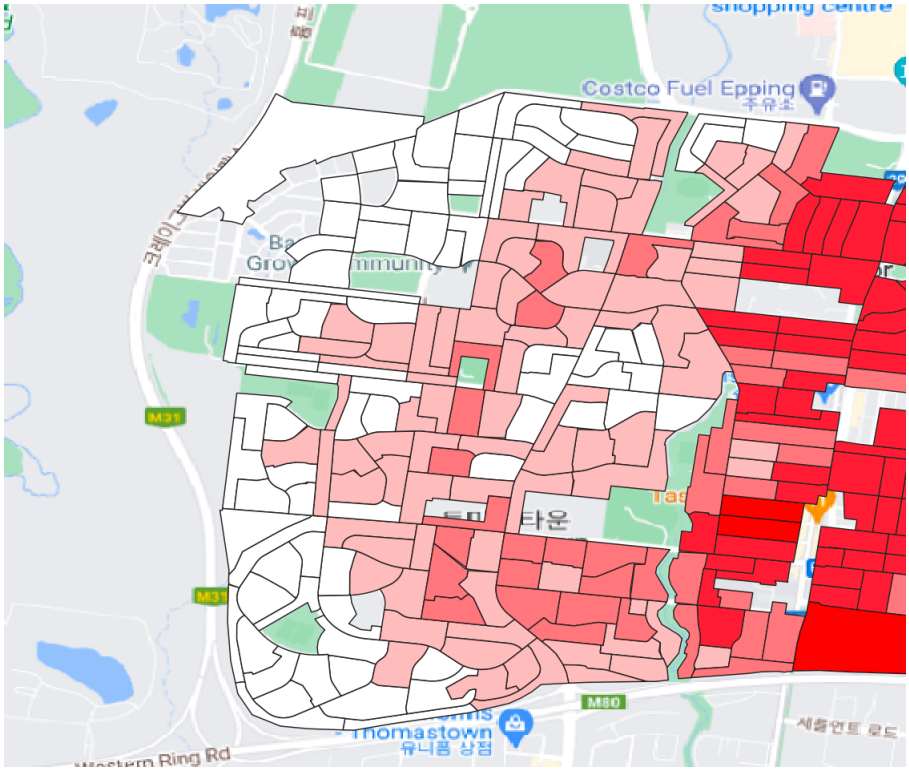
You can check the mesh blocks with more bus routes and mesh blocks with less bus routes.

this is a table of colors expressed by the number of routes.

so, the closer it is to the prefect red, the more routes it means



You can check that mesh blocks which are far from the city have a small number of routes.



5. Conclusion

As I mentioned in the introduction, it is really important ability that analysis map,
by this project I can learn how to analyze the map and visualize the result so that we can check the whole situation easily and find what places have problems.
also if you look at the process, you would check that continue to work by creating tables, which is due to efficiency. By dealing with real big data, I was able to learn that computers cannot handle it unless query is done efficiently.
I hope that this ability will grow further and have an opportunity to work in the actual field.