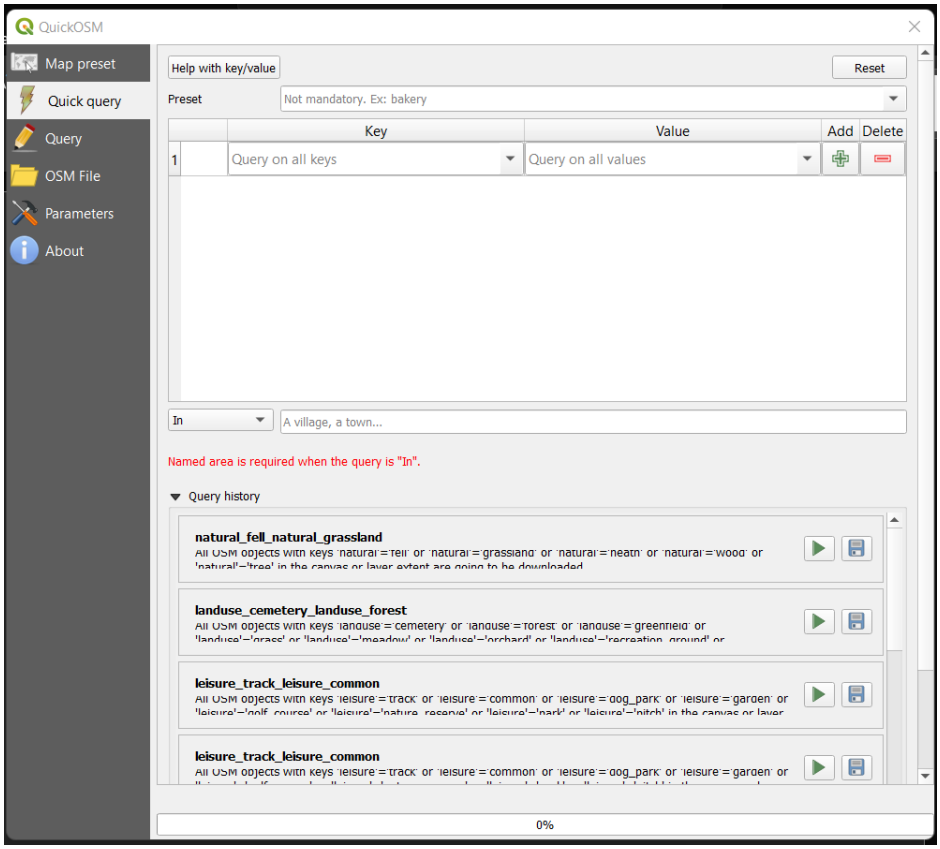
Step 1: - Installed QuickOSM plugin from plugin repository in Qgis

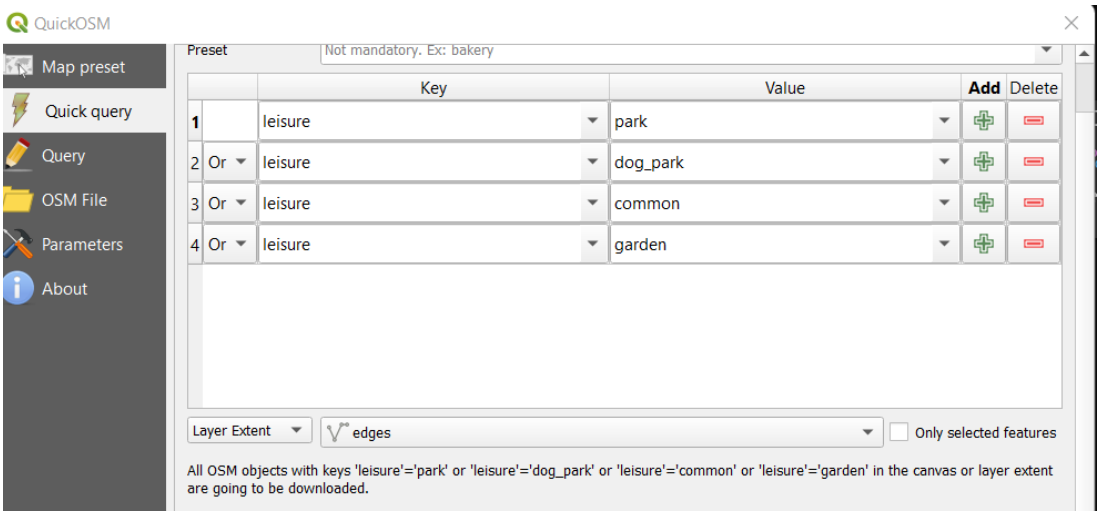
Step 2: - Went to https://wiki.openstreetmap.org/ to find key and value for parks in OSM data .

Step 3: - Created a bounding box around the Melbourne Road network shapefile file to define AOI

Step 4: - Added AOI and opened OuickOSM dialogue box to write query in order to download OSM data of the AOI



Step 5: - Wrote query with OSM key and value to search data using API



Step 6: - There are multiple polygons within a polygon at some locations. So, polygons within polygons were dissolved into larger polygon using dissolve tool in GDAL

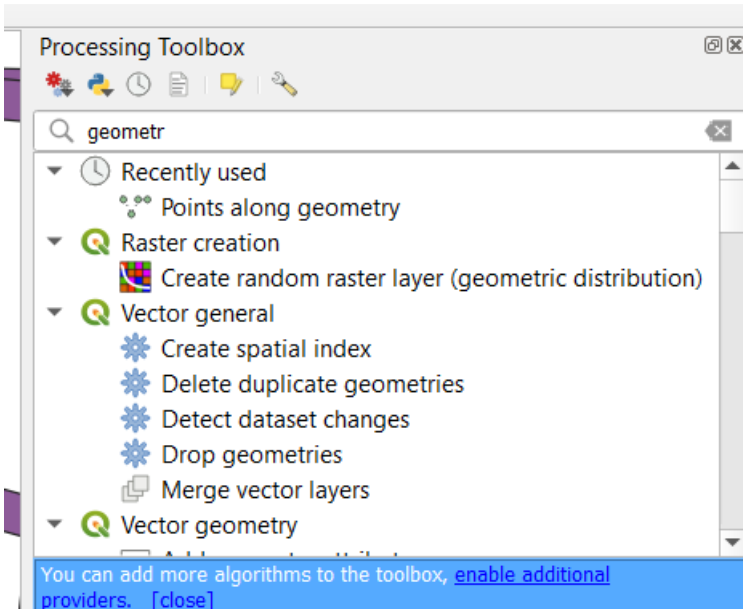
Step 7: - Search for **Points along geometry** in toolbar and create entrance point within an **interval of 50 m based on the paper:**

Public openspace,physicalactivity,urbandesignandpublichealth: Concepts,methodsandresearchagenda

Mohammad JavadKoohsari a,e,n, SuzanneMavoa a, KarenVillanueva a,b,

TakemiSugiyama c,e, HannahBadland a, AndrewT.Kaczynski d, NevilleOwen e,

Billie Giles-Corti



Step 8: - Removed points those were close to starting point from previous shapefile i.e., step 5(the points having distance less than 50 m on boundary of same polygon)

```
select fid, distance, angle, geom from
(select fid, distance, angle, geom,row_number() over (partition by fid) as r from
(select *, row_number() over (partition by fid order by distance asc) as r_asc,
row_number() over (partition by fid order by distance desc) as r_dsc
from interpolated_point_v1)t
where r_asc != 1 and r_dsc != 1)t
union
select fid, distance, angle, geom from
(select *, st_distance(a.geom, b.bgeom)*100000 as dist from
```

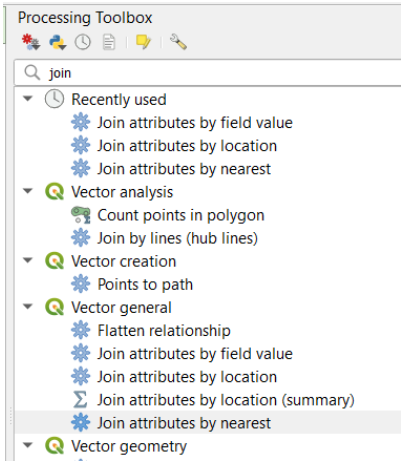
```
(select fid, distance, angle, geom from
(select fid, distance, angle, geom,row_number() over (partition by fid) as r from
(select *, row_number() over (partition by fid order by distance asc) as r_asc,
row_number() over (partition by fid order by distance desc) as r_dsc
from interpolated_point_v1)t
where r_asc = 1 or r_dsc = 1)t
where r = 1) a
join
(select fid as bfid, distance as bdistance, angle as bangle, geom as bgeom from
(select fid, distance, angle, geom,row_number() over (partition by fid) as r from
(select *, row_number() over (partition by fid order by distance asc) as r_asc,
row_number() over (partition by fid order by distance desc) as r_dsc
from interpolated_point_v1)t
where r_asc = 1 or r_dsc = 1)t
where r = 2) b
on a.fid = b.bfid)t
where dist < 50
union
select fid, distance, angle, geom from
(select *, st_distance(a.geom, b.bgeom)*100000 as dist from
(select fid, distance, angle, geom from
(select fid, distance, angle, geom,row_number() over (partition by fid) as r from
(select *, row_number() over (partition by fid order by distance asc) as r_asc,
row_number() over (partition by fid order by distance desc) as r_dsc
from interpolated_point_v1)t
where r_asc = 1 or r_dsc = 1)t
where r = 1) a
join
(select fid as bfid, distance as bdistance, angle as bangle, geom as bgeom from
(select fid, distance, angle, geom,row_number() over (partition by fid) as r from
(select *, row_number() over (partition by fid order by distance asc) as r_asc,
```

```
row_number() over (partition by fid order by distance desc) as r_dsc
from interpolated_point_v1)t
where r_asc = 1 or r_dsc = 1)t
where r = 2) b
on a.fid = b.bfid)t
where dist >= 50
union
select bfid, bdistance, bangle, bgeom from
(select *, st_distance(a.geom, b.bgeom)*100000 as dist from
(select fid, distance, angle, geom from
(select fid, distance, angle, geom,row_number() over (partition by fid) as r from
(select *, row_number() over (partition by fid order by distance asc) as r_asc,
row_number() over (partition by fid order by distance desc) as r_dsc
from interpolated_point_v1)t
where r_asc = 1 or r_dsc = 1)t
where r = 1) a
join
(select fid as bfid, distance as bdistance, angle as bangle, geom as bgeom from
(select fid, distance, angle, geom,row_number() over (partition by fid) as r from
(select *, row_number() over (partition by fid order by distance asc) as r_asc,
row_number() over (partition by fid order by distance desc) as r_dsc
from interpolated_point_v1)t
where r_asc = 1 or r_dsc = 1)t
where r = 2) b
on a.fid = b.bfid)t
where dist >= 50
```

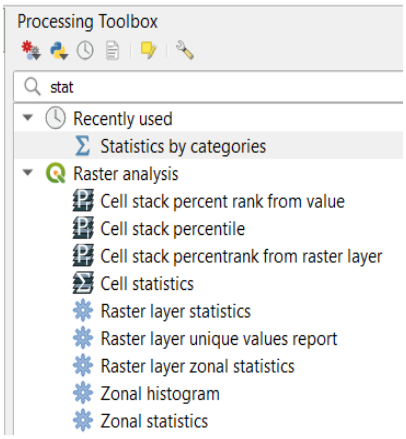
Step 9: - Created a buffer of 10 metres using the network layer to get points those are along the road segments

Step 10: - Clipped the points that lies within the buffer layer of road shapefile

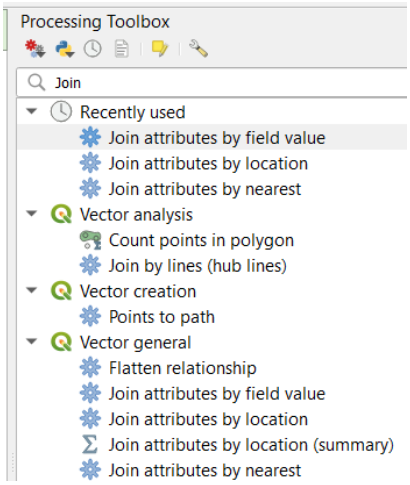
Step 11: - From processing toolbox, selected '**Join attributes by nearest'** to connect each entrance point with its respective segment



Step 12: - Then used '**Statistics by category'** on the output from previous step to count number of entrances in each road segment



Step 13: - Add original road network file and the file form step 12 i.e. file having total count of entrances in each segment. Then join the attribute table on unique ogc_fid0 with original shapefile.



File Descriptions

1) **network_edges4 –** Original Road network file
2) **checked_leisure_park_leisure_dog_park_projected –** OSM data
3) **Entrances_of_parks –** Entrances at an interval of 50 m
4) **entrances_along_each_seg –** Entrances along each segment
5) **count_of_entrances_for_each_seg –** Total count of entrances along each segment csv file
6) **green_count_final –** Total count of entrances along each segment shapefile