2. Essential Facilities for Spatial Analysis

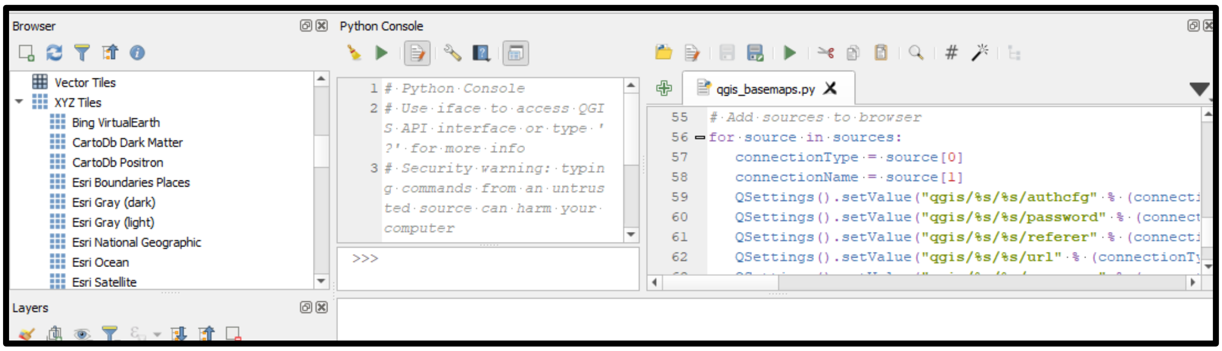
Adding Basemap using PyQGIS:

Download the basemap.py using URL: **https://github.com/giswqs/qgis-earthengine-examples/blob/master/Basemaps/qgis_basemaps.py**

Or

https://github.com/Haseeb-oss-eng/Geospatial_Python/blob/main/2.Essential%20Facilities%20for%20Spatial%20Analysis/qgis_basemaps.py

- Open this above qgis_basemaps.py in QGIS Python Console and running this code which results in adding basemap in Data Source XYZ Tiles.

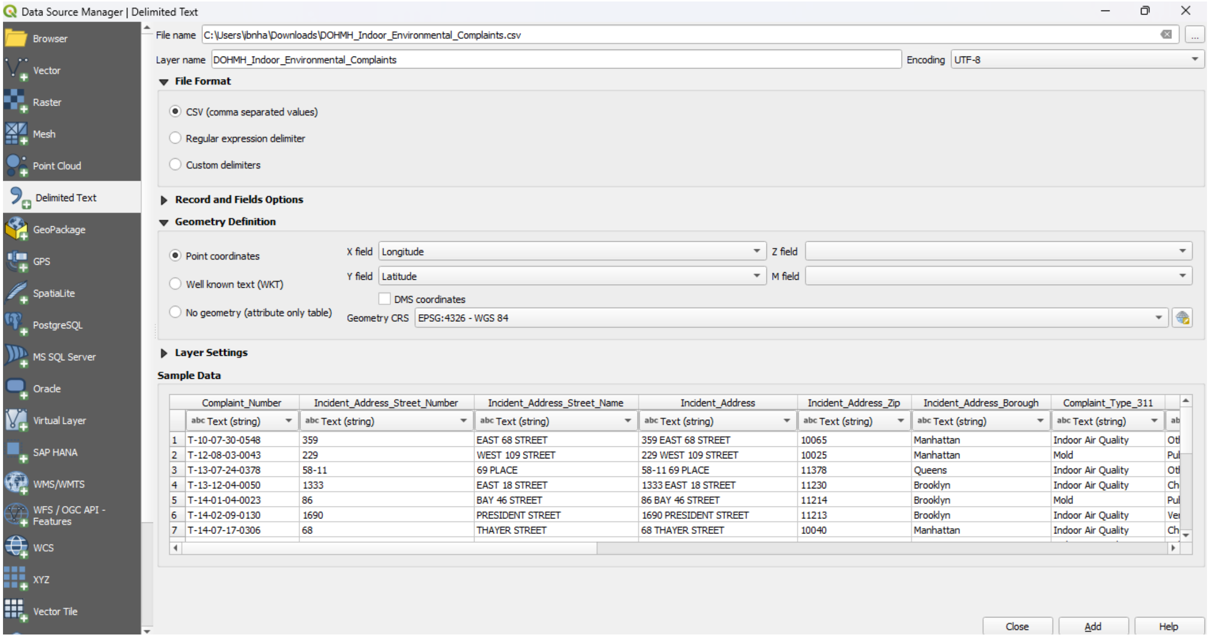


```
# Python Console
# Use iface to access QGIS API interface or type '?' for more info
# Security warning: typing commands from an untrusted source can harm your computer
>>>
```

```
# Add sources to browser
for source in sources:
    connectionType = source[0]
    connectionName = source[1]
    QSettings().setValue("qgis/%s/%s/authcfg" % (connecti
    QSettings().setValue("qgis/%s/%s/password" % (connect
    QSettings().setValue("qgis/%s/%s/referer" % (connecti
    QSettings().setValue("qgis/%s/%s/url" % (connectionTy
```

2.1 Visualizing Environmental Complaints in New York City

- Download Environmental Complaint data
  - https://github.com/Haseeb-oss-eng/Geospatial_Python/blob/main/2.Essential%20Facilities%20for%20Spatial%20Analysis/DOHMH_Indoor_Environmental_Complaints.csv
- Go to Data Source -> Add Delimited Text -> Add path of DOHMH_Indoor_Environmental_Complaints.csv



File name C:\Users\bnha\Downloads\DOHMH_Indoor_Environmental_Complaints.csv

Layer name DOHMH_Indoor_Environmental_Complaints Encoding UTF-8

File Format

● CSV (comma separated values)

○ Regular expression delimiter

○ Custom delimiters

Geometry Definition

● Point coordinates X field Longitude Z field

○ Well known text (WKT) Y field Latitude M field

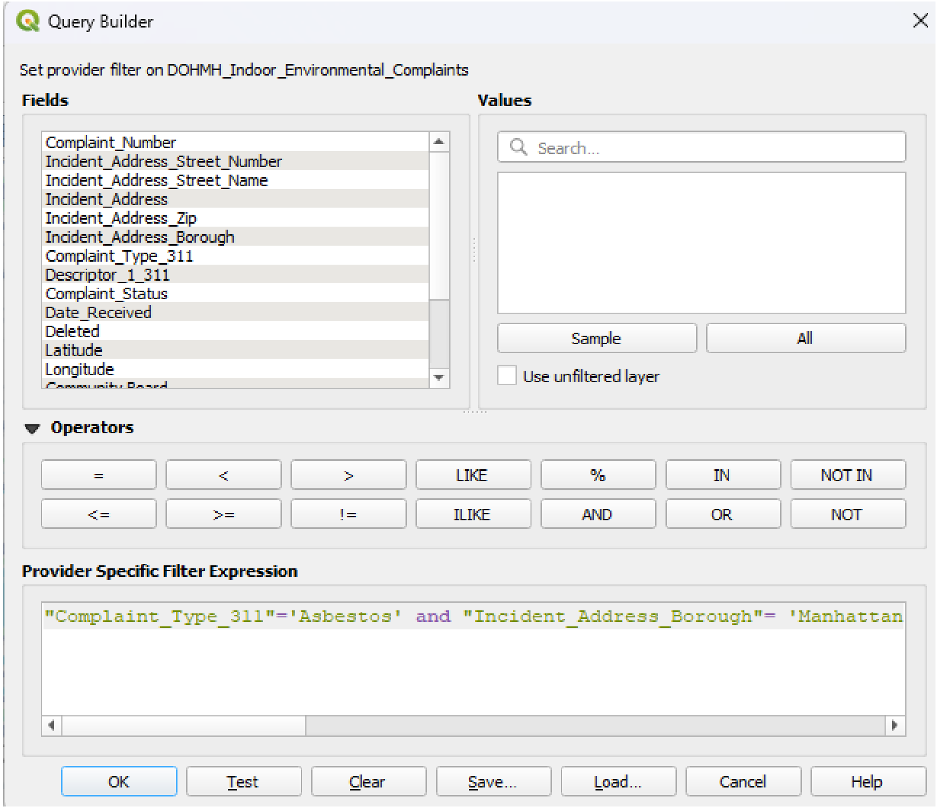
□ DMS coordinates

○ No geometry (attribute only table) Geometry CRS EPSG:4326 - WGS 84

Sample Data

	Complaint_Number	Incident_Address_Street_Number	Incident_Address_Street_Name	Incident_Address	Incident_Address_Zip	Incident_Address_Borough	Complaint_Type_311	
	abc Text (string)	abc Text (string)	abc Text (string)	abc Text (string)	abc Text (string)	abc Text (string)	abc Text (string)	ab
1	T-10-07-30-0548	359	EAST 68 STREET	359 EAST 68 STREET	10065	Manhattan	Indoor Air Quality	Ot
2	T-12-08-03-0043	229	WEST 109 STREET	229 WEST 109 STREET	10025	Manhattan	Mold	Pu
3	T-13-07-24-0378	58-11	69 PLACE	58-11 69 PLACE	11378	Queens	Indoor Air Quality	Ot
4	T-13-12-04-0050	1333	EAST 18 STREET	1333 EAST 18 STREET	11230	Brooklyn	Indoor Air Quality	Ch
5	T-14-01-04-0023	86	BAY 46 STREET	86 BAY 46 STREET	11214	Brooklyn	Mold	Pu
6	T-14-02-09-0130	1690	PRESIDENT STREET	1690 PRESIDENT STREET	11213	Brooklyn	Indoor Air Quality	Ve
7	T-14-07-17-0306	68	THAYER STREET	68 THAYER STREET	10040	Manhattan	Indoor Air Quality	Ch

- Right Click the data in Layer Panel -> Select Filter option -> Query Builder Pop Up -> Give the Condition that
  - **“Complaint_Type_311” =’Asbestos’ and “Incident_Address_Borough” = ’Manhattan’**



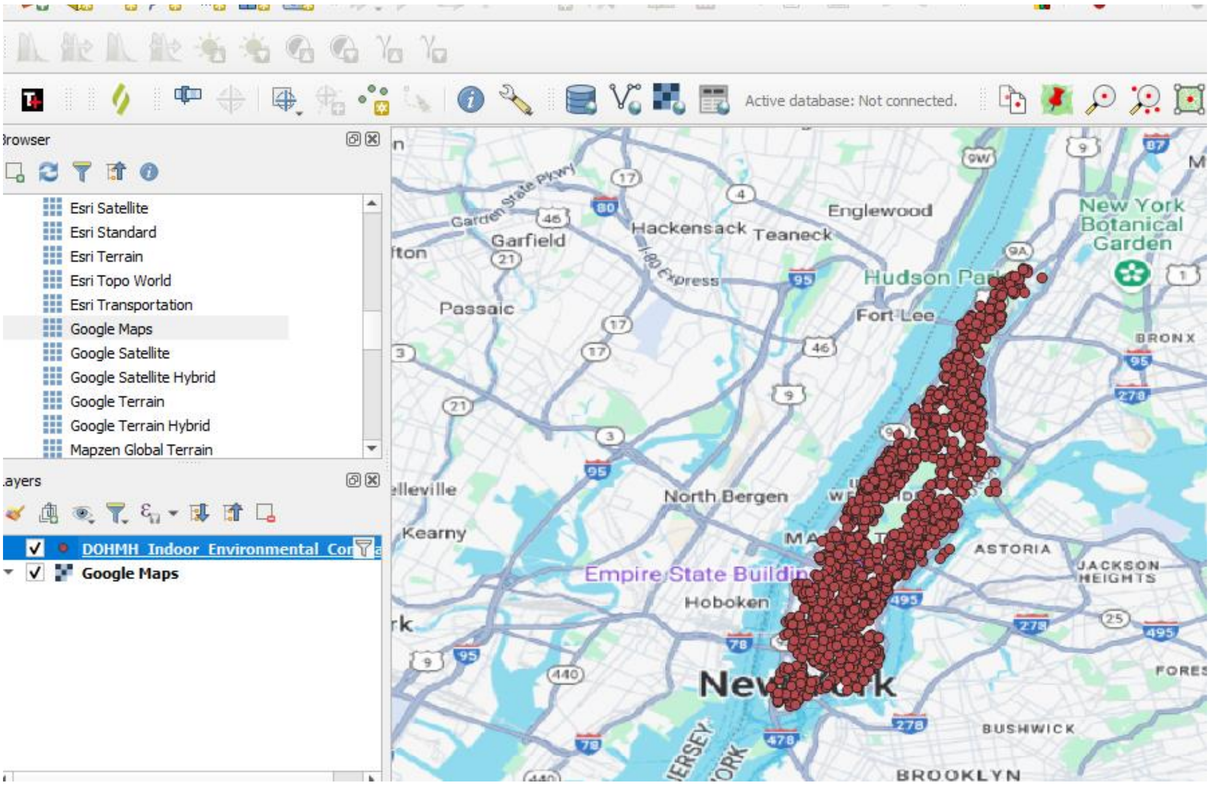
Query Builder

Set provider filter on DOHMH_Indoor_Environmental_Complaints

Provider Specific Filter Expression

```
"Complaint_Type_311"='Asbestos' and "Incident_Address_Borough"= 'Manhattan
```

Result:



Filtered Data Based on Filter Conditions

PyQGIS

1. PyQGIS :- **Vector Analysis**

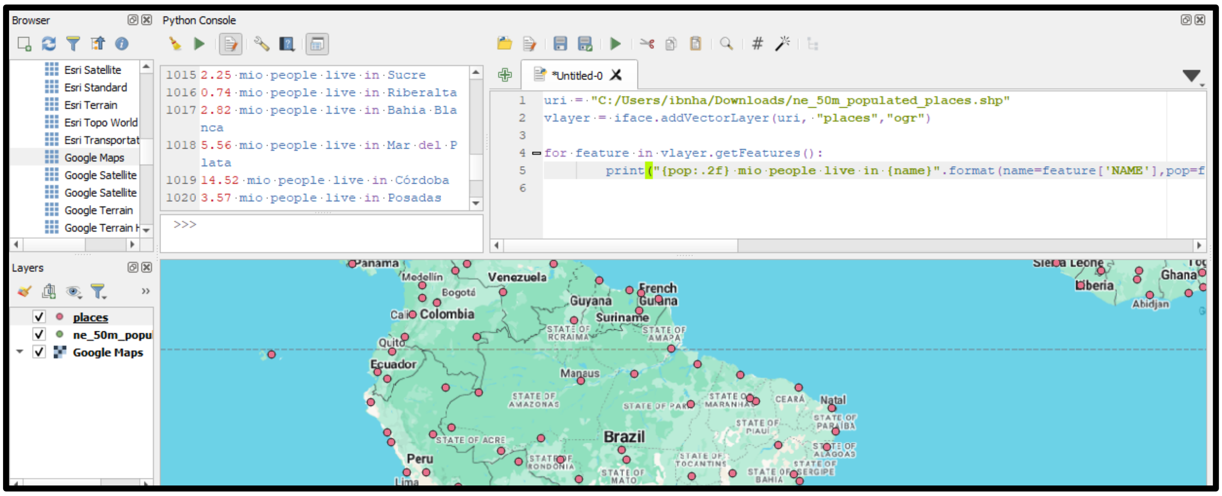
- Go to Plugins in QGIS -> Select Python Console -> Enter the code that add the vector data in QGIS
- **Vector_PyQGIS.py:** https://github.com/Haseeb-oss-eng/Geospatial_Python/blob/main/2.Essential%20Facilities%20for%20Spatial%20Analysis/VectorQgis.py
- **Shape File:** https://github.com/Haseeb-oss-eng/Geospatial_Python/blob/main/2.Essential%20Facilities%20for%20Spatial%20Analysis/ne_50m_populated_places.shp

- To Add Vector Data in QGIS:
  - Use the Syntax

    ```
    iface.addVectorLayer(data_path:String,Name_Layer:String,library:String)

    uri = "C:/Users/ibnha/Downloads/ne_50m_populated_places.shp"

    vlayer = iface.addVectorLayer(uri, "places","ogr")
    ```

  - function to display features from shapefile

    ```
    for feature in vlayer.getFeatures():

        print("{pop:.2f} mio people live in {name}".format(name=feature['NAME'],pop=feature['POP_MAX']/100000))
    ```



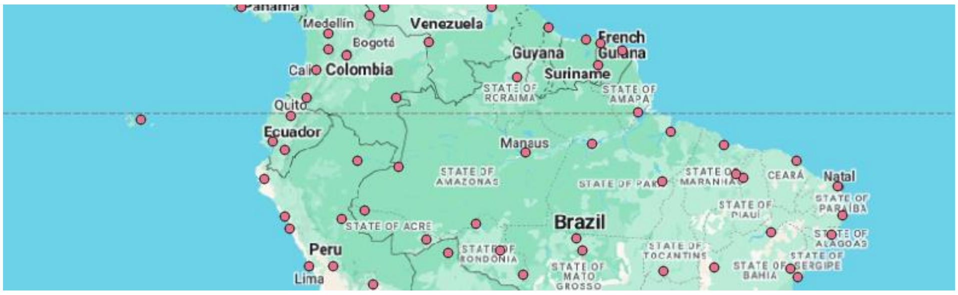
```
2.25 mio people live in Sucre
0.74 mio people live in Riberalta
2.82 mio people live in Bahia Bla
nca
5.56 mio people live in Mar del P
lata
14.52 mio people live in Córdoba
3.57 mio people live in Posadas
>>>
```

```
uri = "C:/Users/ibnha/Downloads/ne_50m_populated_places.shp"
vlayer = iface.addVectorLayer(uri, "places","ogr")

for feature in vlayer.getFeatures():
    print("{pop:.2f} mio people live in {name}".format(name=feature['NAME'],pop=f
```

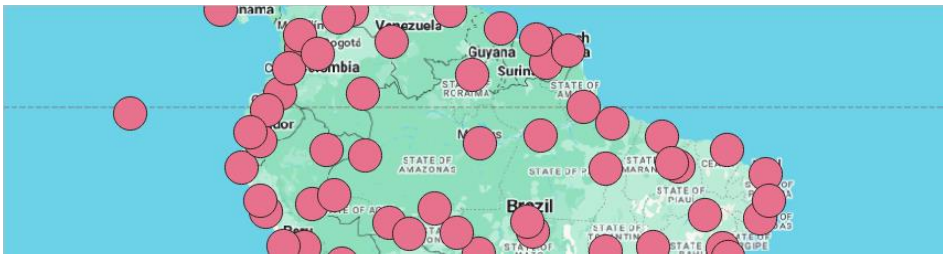
- Change the size of Points

vlayer.renderer().symbol().setSize(8)

vlayer.triggerRepaint()



Before Set Size

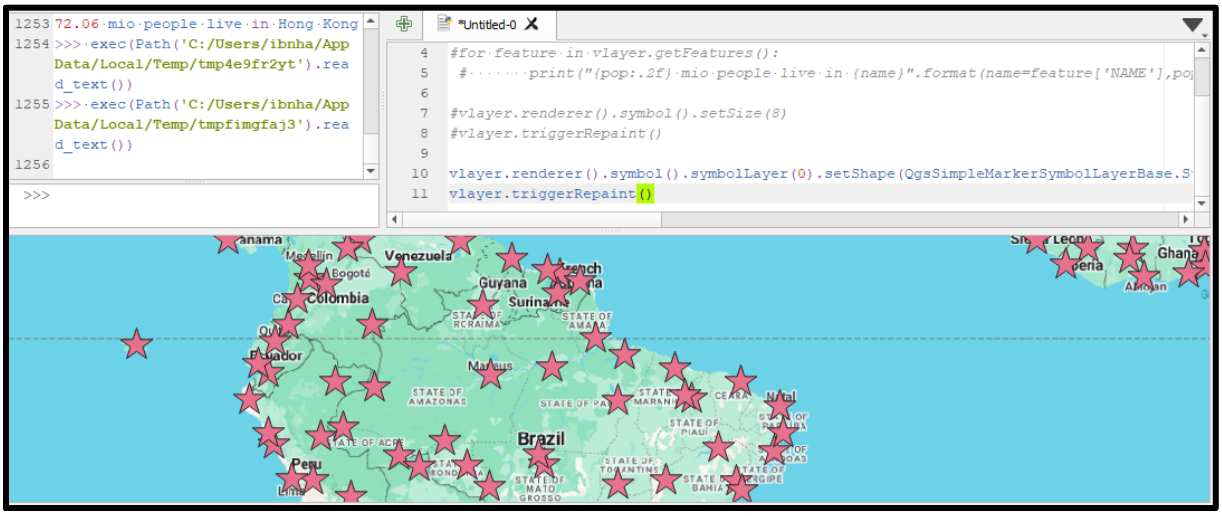


After Set Size – 8

Change the Marker Shape

vlayer.renderer().symbol().symbolLayer(0).setShape(QgsSimpleMarkerSymbolLayerBase.Star)

vlayer.triggerRepaint()



```
72.06 mio people live in Hong Kong
>>> exec(Path('C:/Users/ibnha/AppData/Local/Temp/tmp4e9fr2yt').read_text())
>>> exec(Path('C:/Users/ibnha/AppData/Local/Temp/tmpfimgfaj3').read_text())

>>>
```

*Untitled-0

```
#for feature in vlayer.getFeatures():
 #       print("{pop:.2f} mio people live in {name}".format(name=feature['NAME'],po

#vlayer.renderer().symbol().setSize(8)
#vlayer.triggerRepaint()

vlayer.renderer().symbol().symbolLayer(0).setShape(QgsSimpleMarkerSymbolLayerBase.S
vlayer.triggerRepaint()
```

Changed the Marker Shape using QgisSimpleMarkerSymbolLayerBase

2. PyQGIS: **Raster Analysis**

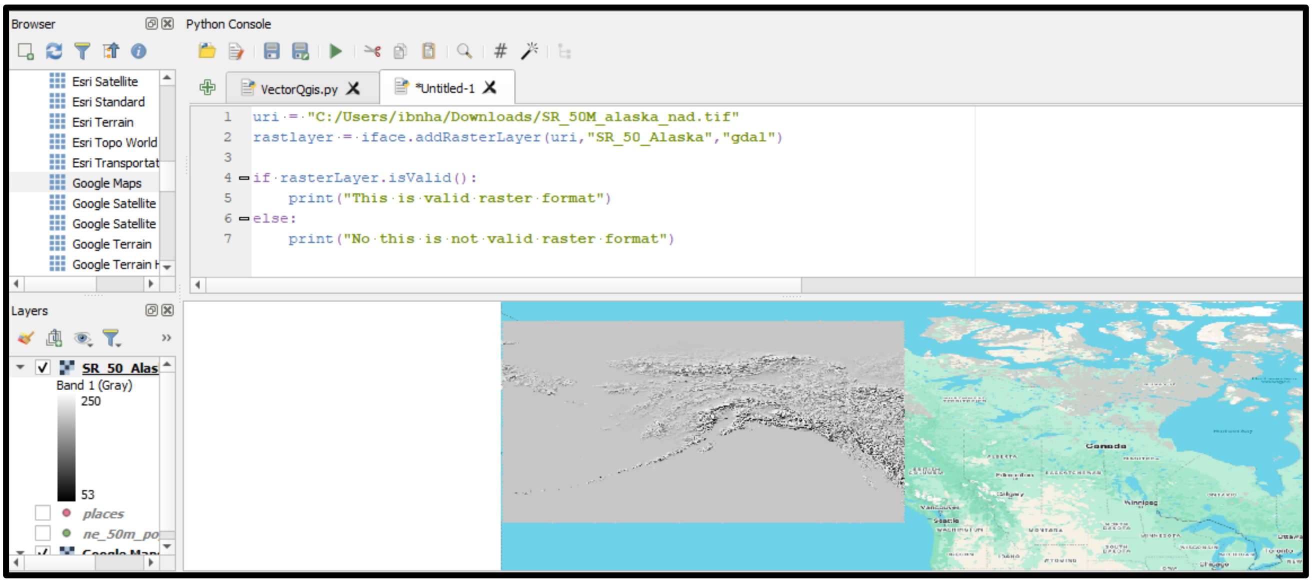
- **Raster File: https://github.com/Haseeb-oss-eng/Geospatial_Python/blob/main/2.Essential%20Facilities%20for%20Spatial%20Analysis/SR_50M_alaska_nad.tif**

- Add Raster in Layer Panel Using Python Console

uri = "C:/Users/ibnha/Downloads/SR_50M_alaska_nad.tif"

rastlayer = iface.addRasterLayer(uri,"SR_50_Alaska","gdal")

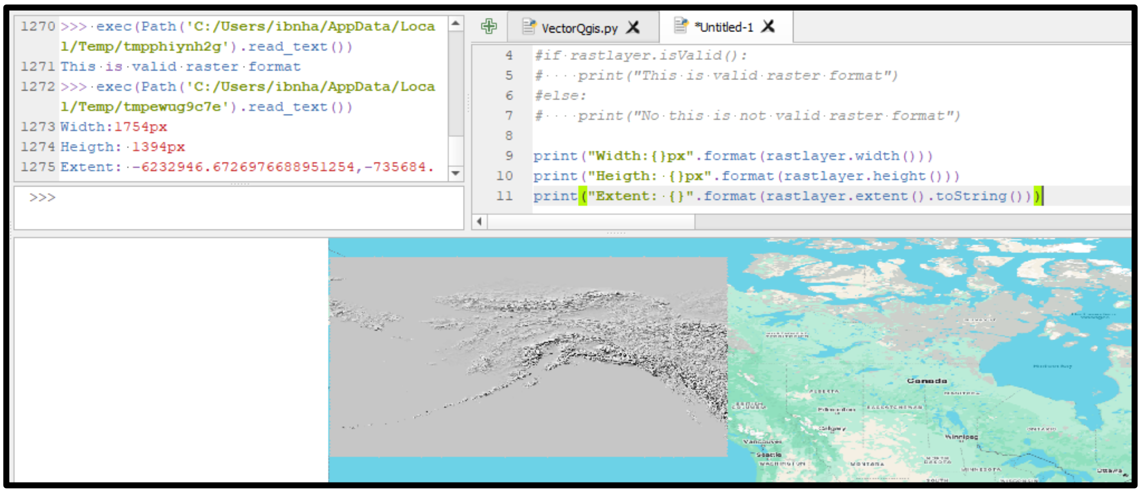
```
if rastlayer.isValid():
    print("This is valid raster format")
else:
    print("No this is not valid raster format")
```



```
uri = "C:/Users/ibnha/Downloads/SR_50M_alaska_nad.tif"
rastlayer = iface.addRasterLayer(uri,"SR_50_Alaska","gdal")

if rasterLayer.isValid():
    print("This is valid raster format")
else:
    print("No this is not valid raster format")
```

- Print the Metadata like Height and Width of a Raster in Pixel

```
print("Width:{}px".format(rastlayer.width()))
print("Heigth: {}px".format(rastlayer.height()))
print("Extent: {}".format(rastlayer.extent().toString()))
```



```
>>> exec(Path('C:/Users/ibnha/AppData/Local/Temp/tmpphiynh2g').read_text())
This is valid raster format
>>> exec(Path('C:/Users/ibnha/AppData/Local/Temp/tmpewug9c7e').read_text())
Width:1754px
Heigth: 1394px
Extent:  -6232946.6726976688951254,-735684.
>>>
```

```
#if rastlayer.isValid():
#    print("This is valid raster format")
#else:
#    print("No this is not valid raster format")

print("Width:{}px".format(rastlayer.width()))
print("Heigth: {}px".format(rastlayer.height()))
print("Extent: {}".format(rastlayer.extent().toString()))
```

In Left Panel the Raster Data Height, Width and Extent is Printed

Styling Shape File Using Attribute Data

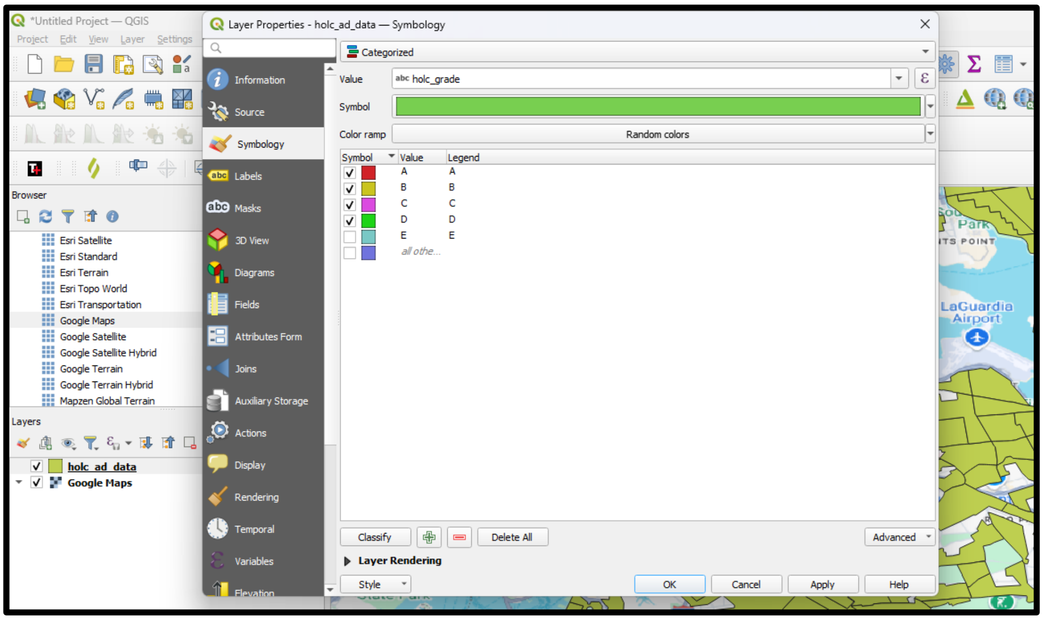
Redlining Maps from the Home Owners Loan Corporation, 1937

"HOLC staff members, using data and evaluations organized by local real estate professionals--lenders, developers, and real estate appraisers--in each city, assigned grades to residential neighborhoods that reflected their "mortgage security" that would then be visualized on color-coded maps. Neighborhoods receiving the highest grade of "A"--colored green on the maps--were deemed minimal risks for banks and other mortgage lenders when they were determining who should receive loans and which areas in the city were safe investments. Those receiving the lowest grade of "D," colored red, were considered "hazardous."

Conservative, responsible lenders, in HOLC judgment, would "refuse to make loans in these areas [or] only on a conservative basis." HOLC created area descriptions to help to organize the data they used to assign the grades. Among that information was the neighborhood's quality of housing, the recent history of sale and rent values, and, crucially, the racial and ethnic identity and class of residents that served as the basis of the neighborhood's grade. These maps and their accompanying documentation helped set the rules for nearly a century of real estate practice. "[1]

Shape File: https://github.com/Haseeb-oss-eng/Geospatial_Python/blob/main/2.Essential%20Facilities%20for%20Spatial%20Analysis/holc_ad_data.shp

- Add Shape File -> Right Click the Shp file -> Go to Properties -> Go to Symbology -> Choose Categorized -> Select the column **“holc_grade”**



Layer Properties - holc_ad_data — Symbology

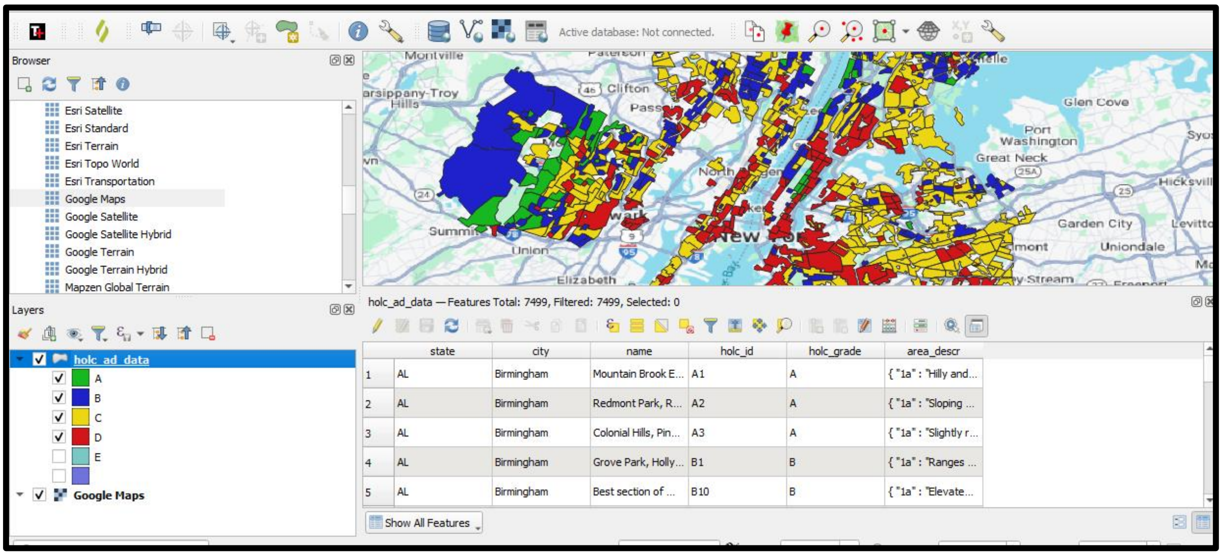
Categorized

Value: holc_grade

Color ramp: Random colors

Symbol	Value	Legend
☑ 🟥	A	A
☑ 🟨	B	B
☑ 🟪	C	C
☑ 🟩	D	D
☐ 🟦	E	E
☐ 🟦	*all othe...*	

Styling Shape File



holc_ad_data — Features Total: 7499, Filtered: 7499, Selected: 0

	state	city	name	holc_id	holc_grade	area_descr
1	AL	Birmingham	Mountain Brook E...	A1	A	{ "1a" : "Hilly and...
2	AL	Birmingham	Redmont Park, R...	A2	A	{ "1a" : "Sloping ...
3	AL	Birmingham	Colonial Hills, Pin...	A3	A	{ "1a" : "Slightly r...
4	AL	Birmingham	Grove Park, Holly...	B1	B	{ "1a" : "Ranges ...
5	AL	Birmingham	Best section of ...	B10	B	{ "1a" : "Elevate...

Red Lined Map

References

1. Redlining Maps from the Home Owners Loan Corporation, 1937 - https://catalog.data.gov/dataset/redlining-maps-from-the-home-owners-loan-corporation-1937
2. Data and Tutorials - https://github.com/Haseeb-oss-eng/Geospatial_Python.git
3. All Practical's is based on Book – Python for Geospatial Data Analysis (Author: Bonny P. McClain, Released October 2022, Publisher(s): O'Reilly Media, Inc. ISBN: 9781098104795)