Project #1

GEOG 4057

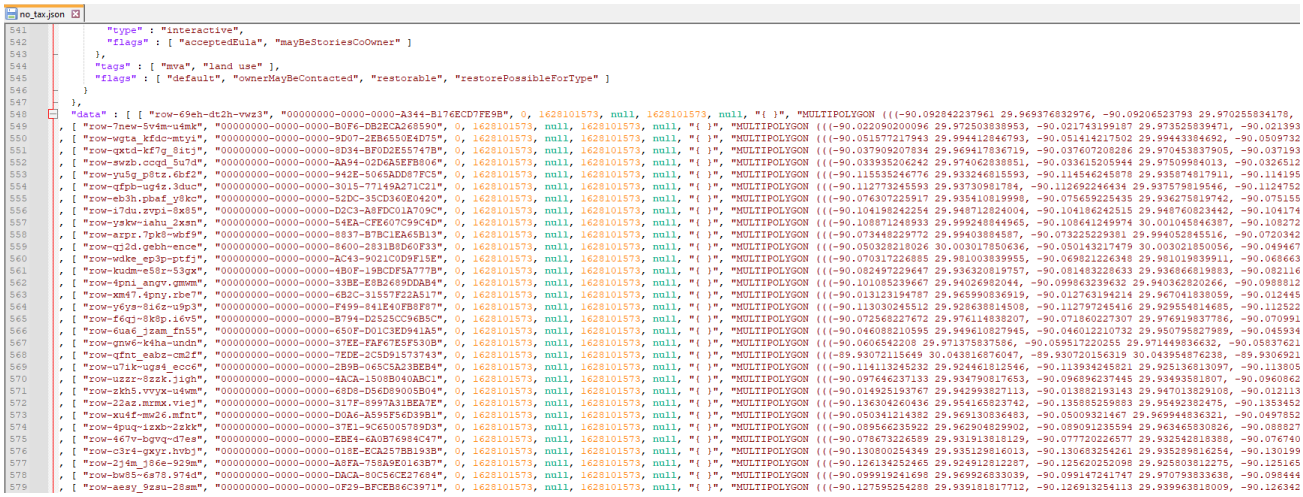
GIS programming

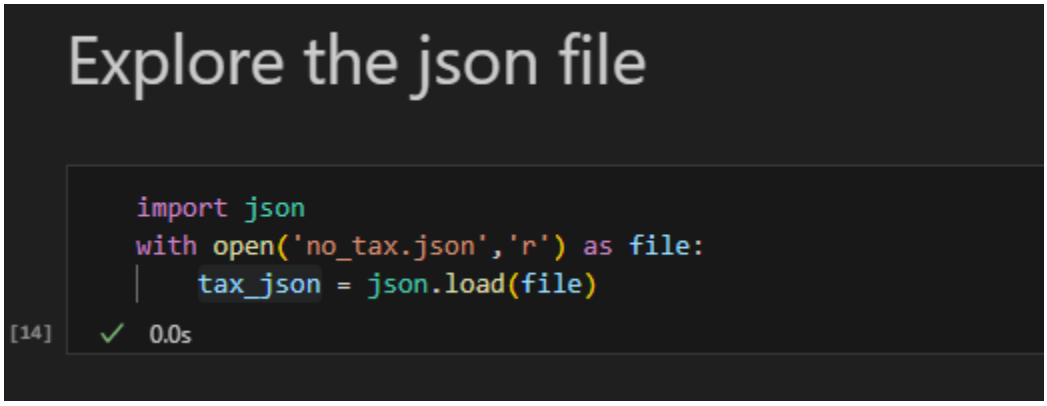
Prof. Lei Wang

Prepared by/

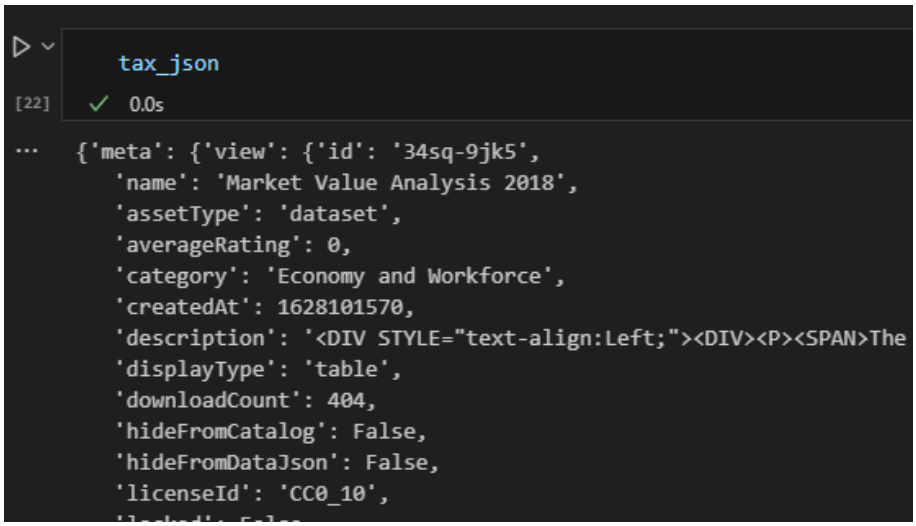
Mohammed Elkharakany

JSON file





Meta and data are explored



```
tax_json['meta']['view']['columns']
```

[23] ✓ 0.0s


```
[{'id': -1,
  'name': 'sid',
  'dataTypeName': 'meta_data',
  'fieldName': ':sid',
  'position': 0,
  'renderTypeName': 'meta_data',
  'format': {},
  'flags': ['hidden']},
 {'id': -1,
  'name': 'id',
  'dataTypeName': 'meta_data',
  'fieldName': ':id',
  'position': 0,
  'renderTypeName': 'meta_data',
  'format': {},
  'flags': ['hidden']},
```


Show field names

```
fields = tax_json['meta']['view']['columns']
for field in fields:
    print(field['name'])
```

[24] ✓ 0.0s

```
sid
id
position
created_at
created_meta
updated_at
updated_meta
meta
the_geom
OBJECTID
ID
Cluster Letter
Shape.STArea()
Shape.STLength()
```

```
import arcpy
arcpy.FromWKT(tax_json['data'][8][8])
```

[3] ✓ 4.4s

```
import arcpy
for row in tax_json['data']:
    print(row)
```

[50] ✓ 0.0s

```
['row-69eh-dt2h-vwz3', '00000000-0000-0000-A344-B176ECD7FE9B', 0, 1628101573, None, 1628101
['row-7new-5v4m~u4mk', '00000000-0000-0000-B0F6-DB2ECA268590', 0, 1628101573, None, 1628101
['row-wgta_kfdc~mtyi', '00000000-0000-0000-9D07-2EB6550E4D75', 0, 1628101573, None, 1628101
['row-qxtd-kf7g_8itj', '00000000-0000-0000-8D34-BF0D2E55747B', 0, 1628101573, None, 1628101
['row-swzb.ccqd_5u7d', '00000000-0000-0000-AA94-02D6A5EFB806', 0, 1628101573, None, 1628101
['row-yu5g_p8tz.6bf2', '00000000-0000-0000-942E-5065ADD87FC5', 0, 1628101573, None, 1628101
['row-qfpb-ug4z.3duc', '00000000-0000-0000-3015-77149A271C21', 0, 1628101573, None, 1628101
['row-eb3h.pbaf_y8kc', '00000000-0000-0000-52DC-35CD360E0420', 0, 1628101573, None, 1628101
['row-i7du.zvpi-8x85', '00000000-0000-0000-D2C3-A8FDC01A709C', 0, 1628101573, None, 1628101
['row-yskw~iahu_2xsn', '00000000-0000-0000-54EA-CFE607C99C4D', 0, 1628101573, None, 1628101
```

Create a feature class and write fields

```
import arcpy
for row in tax_json['data']:
    row[8] = arcpy.FromWKT(row[8])


import os
fcname = 'notax_fc.shp'
workspace = r'D:\Before_Hard_Drive\Study\Fifth_Semester_Fall2024\GIS_Programming_GEOG_4057\project1'
fc_fullname = os.path.join(workspace,fcname)
if arcpy.Exists(fc_fullname):
    arcpy.management.Delete(fc_fullname)

arcpy.management.CreateFeatureclass(out_path=workspace,out_name=fcname,
                                    geometry_type='POLYGON',
                                    spatial_reference=4236)

desc = arcpy.da.Describe(fc_fullname)
for field in desc['fields']:
    print(field.name)
```

[41] ✓ 3.2s

```
FID
Shape
Id
```

Add field information

+ Code + Markdown

```
## add field names
fields = tax_json['meta']['view']['columns']
for field in fields:
    print(field['name'])
field_type = ['TEXT','TEXT','LONG','LONG','TEXT','LONG','TEXT','TEXT','TEXT','TEXT','TEXT','TEXT','TEXT']
field_names = []
for ind,field in enumerate(fields):
    name = field['name']
    if name == 'the_geom':
        continue
    if name.lower() == 'id':
        name = f'id_{ind}'
    max_len = min(10,len(name))
    name = name[:max_len]
    field_names.append(name)
field_names = [field.replace(" ","_") for field in field_names]
field_names = [field.replace(".","_") for field in field_names]
field_names
```

[42] ✓ 0.0s

```
sid
id
position
created_at
created_meta
updated_at
updated_meta
meta
the_geom
OBJECTID
ID
Cluster Letter
Shape.STArea()
Shape.STLength()
```

Add field names into the feature class

```
for ind,field_name in enumerate(field_names):
    arcpy.management.AddField(fc_fullname,field_name=field_name,field_type=field_type[ind])
```

[43] ✓ 0.6s

Add data to the feature class

```
field_names.append('SHAPE@')
```

[44] ✓ 0.0s

```
field_names
```

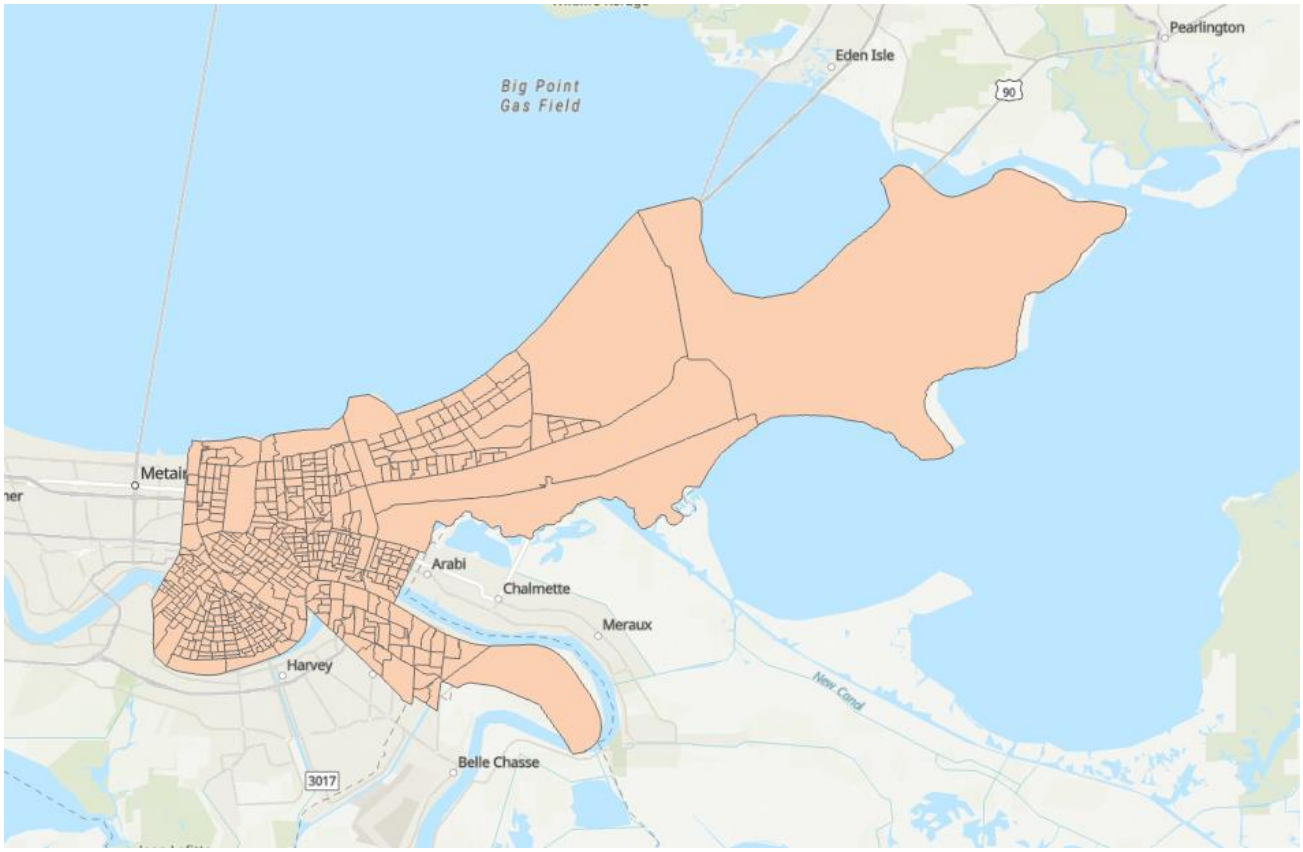
[45] ✓ 0.0s

```
['sid',
 'id_1',
 'position',
 'created_at',
 'created_me',
 'updated_at',
 'updated_me',
 'meta',
 'OBJECTID',
 'id_10',
 'Cluster_Le',
 'Shape_STAr',
 'Shape_STLe',
 'SHAPE@']
```

```
## Write data to the shapefile

with arcpy.da.InsertCursor(fc_fullname,field_names=field_names) as cursor:
    for row in tax_json['data']:
        new_row = []
        for ind, value in enumerate(row):
            if ind == 8:
                continue
            if value == None:
                value = ""
            new_row.append(value)
        new_row.append(row[8])
        cursor.insertRow(new_row)
```

[46] ✓ 0.1s


Pearlington
Eden Isle
Big Point
Gas Field
90
Arabi
Chalmette
Meraux
Harvey
New Cond
Belle Chasse
3017

Creating a python file (.py) to use it in creating an ArcGIS tool (.pyt)

D: > Before_Hard_Drive > Study > Fifth_Semester_Fall2024 > GIS_Programming_GEOG_4057 > project1 > project1.py > ...

```python
# Explore the json file
import arcpy
import json
import os


with open('no_tax.json','r') as file:
    tax_json = json.load(file)


arcpy.FromWKT(tax_json['data'][8][8])
for row in tax_json['data']:
    row[8] = arcpy.FromWKT(row[8])




## Create a feature class and write fields
fcname = 'notax_fc.shp'
workspace = r'D:\Before_Hard_Drive\Study\Fifth_Semester_Fall2024\GIS_Programming_GEOG_4057\project1'
fc_fullname = os.path.join(workspace,fcname)
if arcpy.Exists(fc_fullname):
    arcpy.management.Delete(fc_fullname)

arcpy.management.CreateFeatureclass(out_path=workspace,out_name=fcname,geometry_type='POLYGON',spatial_reference=4236)
## add field names
fields = tax_json['meta']['view']['columns']
for field in fields:
    print(field['name'])
field_type = ['TEXT','TEXT','LONG','LONG','TEXT','LONG','TEXT','TEXT','TEXT','TEXT','TEXT','TEXT','TEXT']
field_names = []
for ind,field in enumerate(fields):
    name = field['name']
    if name == 'the_geom':
        continue
    if name.lower() == 'id':
        name = f'id_{ind}'
    max_len = min(10,len(name))
    name = name[:max_len]
    field_names.append(name)
field_names = [field.replace(" ","_") for field in field_names]
field_names = [field.replace(".","_") for field in field_names]

```

```python

for ind,field_name in enumerate(field_names):
    arcpy.management.AddField(fc_fullname,field_name=field_name,field_type=field_type[ind])

field_names.append('SHAPE@')

## Write data to the shapefile
with arcpy.da.InsertCursor(fc_fullname,field_names=field_names) as cursor:
    for row in tax_json['data']:
        new_row = []
        for ind, value in enumerate(row):
            if ind == 8:
                continue
            if value == None:
                value = ""
            new_row.append(value)
        new_row.append(row[8])
        cursor.insertRow(new_row)

```

Run the python code

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  JUPYTER

amming_GE

(ArcPyClone) D:\Before_Hard_Drive\Study\Fifth_Semester_Fall2024\GIS_Programming_GEOG_4057\project1>python project1.py
sid
id
position
created_at
created_meta
updated_at
updated_meta
meta
the_geom
OBJECTID
ID
Cluster Letter
Shape.STArea()
Shape.STLength()
```

Take the inputs from the user

This part of the code is modified to take the inputs from the user:

```
6   def importNoTaxJSON(workspace = r'D:\Before_Hard_Drive\Study\Fifth_Semester_Fall2024\GIS_Programming_GEOG_4057\project1', json_file='no_tax.json', out_fc='notax_fc_1.shp'):
7
8       with open(json_file,'r') as file:
9           tax_json = json.load(file)
10
11
12      arcpy.FromWKT(tax_json['data'][8][8])
13      for row in tax_json['data']:
14          row[8] = arcpy.FromWKT(row[8])
15
16
17
18
19      ## Create a feature class and write fields
20      fcname = out_fc
21      fc_fullname = os.path.join(workspace,fcname)
22      if arcpy.Exists(fc_fullname):
23          arcpy.management.Delete(fc_fullname)
24
```

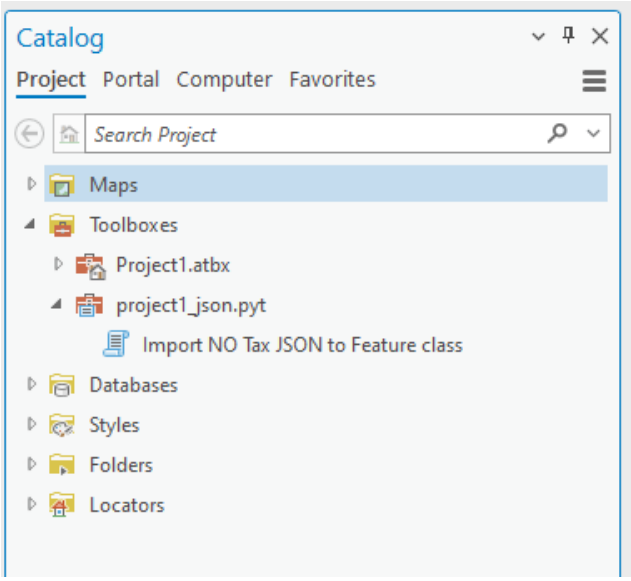
```
62  def main():
63      import sys
64      out_fc = sys.argv[1]
65      importNoTaxJSON(out_fc=out_fc)
66
67  if __name__ == '__main__':
68      main()
```

Run & output

```
(ArcPyClone) D:\Before_Hard_Drive\Study\Fifth_Semester_Fall2024\GIS_Programming_GEOG_4057\project1>python project1.py notax_fc3.shp
```

notax_fc3.cpg	12/4/2024 11:11 PM	CPG File	1 KB
notax_fc3.dbf	12/4/2024 11:11 PM	DBF File	1,249 KB
notax_fc3.prj	12/4/2024 11:11 PM	PRJ File	1 KB
notax_fc3.shp	12/4/2024 11:11 PM	SHP File	394 KB
notax_fc3.shp.xml	12/4/2024 11:11 PM	Microsoft Edge H...	5 KB
notax_fc3.shx	12/4/2024 11:11 PM	SHX File	4 KB

Creating a tool (.pyt)



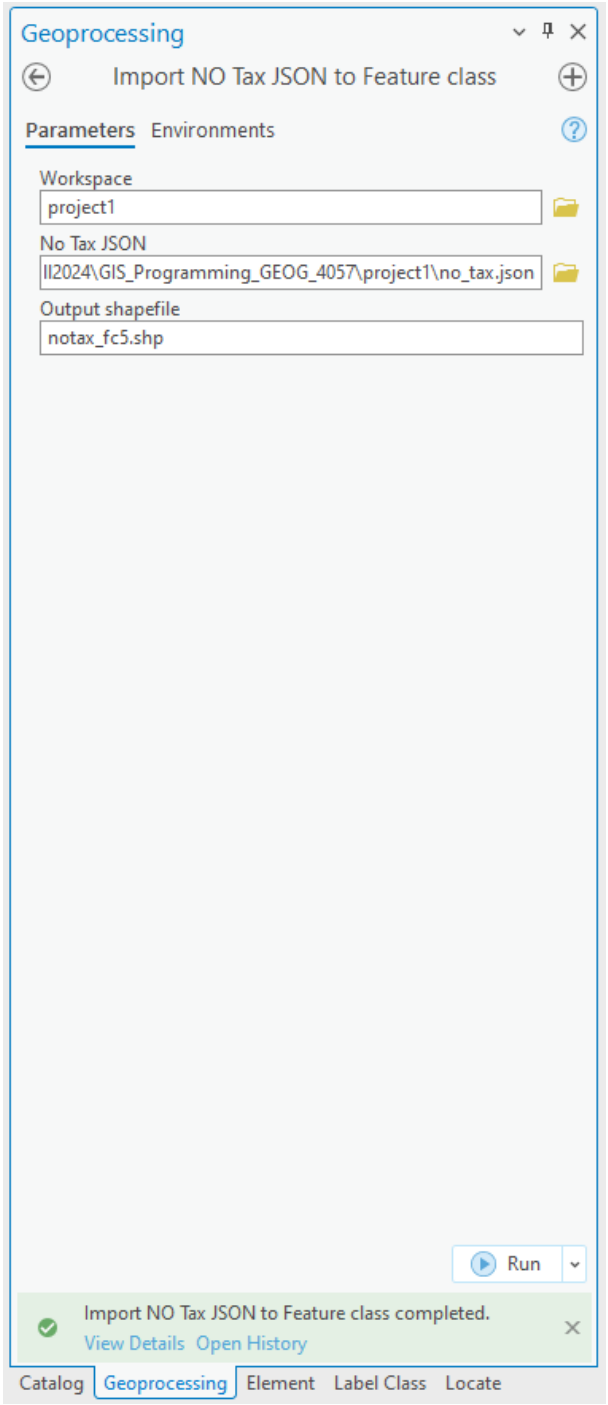
Modifications made:

```
import arcpy
from project1 import importNoTaxJSON
```

```

    def getParameterInfo(self):
        """Define the tool parameters."""
        param_ws = arcpy.Parameter(
            name='workspace',
            displayName='Workspace',
            direction='Input',
            parameterType='Required',
            datatype='DEWorkspace'
        )
        param_json = arcpy.Parameter(
            name='json',
            displayName='No Tax JSON',
            direction='Input',
            parameterType='Required',
            datatype='DEFile'
        )
        param_out = arcpy.Parameter(
            name='Output',
            displayName='Output shapefile',
            parameterType='Required',
            direction='Output',
            datatype='GPString'
        )
        params = [param_ws, param_json, param_out]
        return params
```

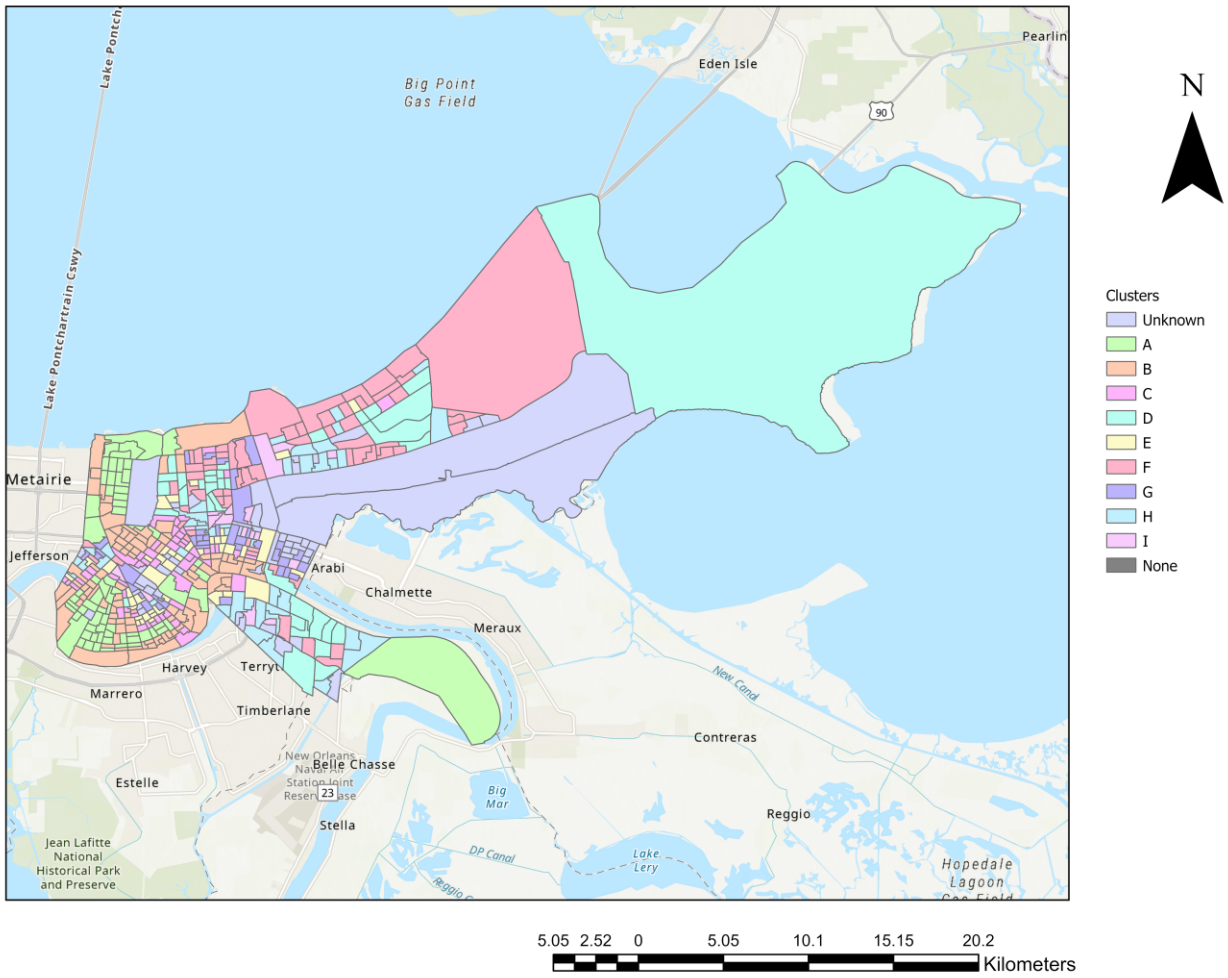
```
65      def execute(self, parameters, messages):
66
67          workspace = parameters[0].valueAsText
68          json_file = parameters[1].valueAsText
69          out_fc = parameters[2].valueAsText
70          importNoTaxJSON(workspace=workspace, json_file=json_file, out_fc=out_fc)
71          return
```



Output

notax_fc5.cpg	12/5/2024 12:09 AM	CPG File	1 KB
notax_fc5.dbf	12/5/2024 12:09 AM	DBF File	1,249 KB
notax_fc5.prj	12/5/2024 12:09 AM	PRJ File	1 KB
notax_fc5.shp	12/5/2024 12:09 AM	SHP File	394 KB
notax_fc5.shp.xml	12/5/2024 12:09 AM	Microsoft Edge H...	5 KB
notax_fc5.shx	12/5/2024 12:09 AM	SHX File	4 KB

Layout Clusters Map



GitHub Link:

https://github.com/Mohammed-Elkharakany/project1_GEOG4057.git