

**Project #1**  
**GEOG 4057**  
**GIS programming**  
**Prof. Lei Wang**  
**Prepared by/**  
**Mohammed Elkharakany**

## JSON file

```

541:         "type": "interactive",
542:         "flags": [ "acceptedRule", "mayBeOwnerOwner" ]
543:     },
544:     {
545:         "type": [ "mva", "land use" ],
546:         "flags": [ "default", "ownerMayBeContacted", "restorable", "restorePossibleForType" ]
547:     },
548:     {
549:         "data": [ [ "row-69eh-dtzh-vvaz3", "0000000000-0000-0000-A344-1B76CD2FEB0", 0, 1628101573, null, 1628101573, null, "1", "MULTIPOLYGON (((-90.092462237961 29.969376832976, -90.092062523793 29.970255834178,
550:         [ "row-k7ew-5vnm-utkm", "0000000000-0000-0000-B0FE-62D8A2C68590", 0, 1628101573, null, 1628101573, null, "1", "MULTIPOLYGON (((-90.092000200096 29.972503389593, -90.074131594187 29.974538239471, -90.021393
551:         [ "row-4kfc-3v8d-3v8d", "0000000000-0000-0000-CEB2-4F747D7347", 0, 1628101573, null, 1628101573, null, "1", "MULTIPOLYGON (((-90.053172115649 29.984412646793, -90.03014217002 29.9844384662, -90.050212
552:         [ "row-gxtd-kftg-1bt", "0000000000-0000-0000-84D4-BF0D2E557474", 0, 1628101573, null, 1628101573, null, "1", "MULTIPOLYGON (((-90.092090080024 29.969431783619, -90.037607202826 29.970453837905, -90.037193
553:         [ "row-svzb-qcoz-5utd", "0000000000-0000-0000-AA94-02D6A5E7B806", 0, 1628101573, null, 1628101573, null, "1", "MULTIPOLYGON (((-90.033932402242 29.974062388851, -90.033615205944 29.97059984013, -90.0326512
554:         [ "row-yusg-pbtz-6b2z", "0000000000-0000-0000-942E-50A4D5D87FC6", 0, 1628101573, null, 1628101573, null, "1", "MULTIPOLYGON (((-90.115552642676 29.933224615593, -90.114346245978 29.935874817911, -90.114195
555:         [ "row-qfqb-qvys-3dzc", "0000000000-0000-0000-3015-771494271C21", 0, 1628101573, null, 1628101573, null, "1", "MULTIPOLYGON (((-90.112692244634 29.937578195446, -90.1124752
556:         [ "row-ebhh-pbaf-5v8d", "0000000000-0000-0000-52DC-AC3D46A46430", 0, 1628101573, null, 1628101573, null, "1", "MULTIPOLYGON (((-90.076700228197 29.98401819599, -90.06846225433 29.983768157447, -90.07612
557:         [ "row-17du-zvpi-8x5d", "0000000000-0000-0000-02C3-AD3F0A15705C", 0, 1628101573, null, 1628101573, null, "1", "MULTIPOLYGON (((-90.104194222554 29.948071282404, -90.10418624255 29.948780823242, -90.104174
558:         [ "row-yvse-iabw-3u8d", "0000000000-0000-0000-54DE-CE607C95C4D", 0, 1628101573, null, 1628101573, null, "1", "MULTIPOLYGON (((-90.108871248993 29.959224848495, -90.108621249974 29.960104554637, -90.108272
559:         [ "row-azpr-1p8k-wb5d", "0000000000-0000-0000-8837-B761EA651361", 0, 1628101573, null, 1628101573, null, "1", "MULTIPOLYGON (((-90.073474229732 29.96403848587, -90.0734225931 29.96405848587, -90.0720342
560:         [ "row-12p2-8v8d", "0000000000-0000-0000-942E-50A4D5D87FC6", 0, 1628101573, null, 1628101573, null, "1", "MULTIPOLYGON (((-90.053172115649 29.984412646793, -90.03014217002 29.9844384662, -90.050212
561:         [ "row-wdte-ep3p-ctfc", "0000000000-0000-0000-AC43-901CD2F5DAB1", 0, 1628101573, null, 1628101573, null, "1", "MULTIPOLYGON (((-90.070317222685 29.981003839595, -90.069832122638 29.981019359111, -90.068663
562:         [ "row-kudm-85E-533x", "0000000000-0000-0000-4B0F-198C0DFA577B", 0, 1628101573, null, 1628101573, null, "1", "MULTIPOLYGON (((-90.082492726547 29.9363203819757, -90.08481228633 29.9368661983, -90.082216
563:         [ "row-4pni-aznq-8v8d", "0000000000-0000-0000-33BE-EBE29ED9D5A1", 0, 1628101573, null, 1628101573, null, "1", "MULTIPOLYGON (((-90.101085296736 29.9402692004, -90.09963233632 29.94036202066, -90.0989812
564:         [ "row-xmt7-tpny-rbzt", "0000000000-0000-0000-6B3C-3157F72D8177", 0, 1628101573, null, 1628101573, null, "1", "MULTIPOLYGON (((-90.013231947787 29.96839036915, -90.012961394214 29.967041830959, -90.012445
565:         [ "row-yv8e-61dc-9v8d", "0000000000-0000-0000-7439-941E0FB0F81F", 0, 1628101573, null, 1628101573, null, "1", "MULTIPOLYGON (((-90.113020245512 29.92638314508, -90.11279745416 29.92854814685, -90.112522
566:         [ "row-f6qj-8ktp-1v8d", "0000000000-0000-0000-74B9-D2525C9F845C", 0, 1628101573, null, 1628101573, null, "1", "MULTIPOLYGON (((-90.072568227674 29.976114832807, -90.07186022379 29.976198337789, -90.070991
567:         [ "row-6u4d-3jag-5v8d", "0000000000-0000-0000-650F-0D13CED9145A", 0, 1628101573, null, 1628101573, null, "1", "MULTIPOLYGON (((-90.046020812055 29.949610287945, -90.046012210732 29.95798327989, -90.045934
568:         [ "row-gm6f-6ka-uhnd", "0000000000-0000-0000-370E-7AF67E5F93D0", 0, 1628101573, null, 1628101573, null, "1", "MULTIPOLYGON (((-90.0665420204 29.971375837586, -90.059517220255 29.97144936432, -90.05837621
569:         [ "row-qfnt-ebdz-ctdc", "0000000000-0000-0000-7EE2-2C3D9147D7", 0, 1628101573, null, 1628101573, null, "1", "MULTIPOLYGON (((-89.93071156319 29.94494676235, -89.9306921
570:         [ "row-11dc-ep3p-ctfc", "0000000000-0000-0000-2B9B-65A5C3A2B81E", 0, 1628101573, null, 1628101573, null, "1", "MULTIPOLYGON (((-90.11413342523 29.924461812546, -90.11394245821 29.92513613097, -90.113805
571:         [ "row-uizz-82zk-3jtg", "0000000000-0000-0000-4ACA-1508B0A0ABC1", 0, 1628101573, null, 1628101573, null, "1", "MULTIPOLYGON (((-90.097466237133 29.934790781763, -90.09686237445 29.9349581807, -90.0960682
572:         [ "row-zkhs-vvpx-utkm", "0000000000-0000-0000-60D8-25E08A9
```

## Explore the json file

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

✓ 0.0s

## Meta and data are explored

tax json

✓ 0.05

```
... {'meta': {'view': {'id': '34sq-9jk5',
    'name': 'Market Value Analysis 2018',
    'assetType': 'dataset',
    'averageRating': 0,
    'category': 'Economy and Workforce',
    'createdAt': 1628101570,
    'description': '<DIV STYLE="text-align:left;"><DIV><P><SPAN>The',
    'displayType': 'table',
    'downloadCount': 404,
    'hideFromCatalog': False,
    'hideFromDataJson': False,
    'licenseId': 'CC0_10',
    'locked': False,
```

```
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, 1628101573, ...  
['row-7new-5v4m~u4mk', '00000000-0000-0000-B0F6-DB2ECA268590', 0, 1628101573, None, 1628101573, ...  
['row-wgta\_kfdc~mtyi', '00000000-0000-0000-9D07-2EB6550E4D75', 0, 1628101573, None, 1628101573, ...  
['row-qxtd-kf7g\_8itj', '00000000-0000-0000-8D34-BF0D2E55747B', 0, 1628101573, None, 1628101573, ...  
['row-swzb.ccqd\_5u7d', '00000000-0000-0000-AA94-02D6A5EFB806', 0, 1628101573, None, 1628101573, ...  
['row-yu5g\_p8tz.6bf2', '00000000-0000-0000-942E-5065ADD87FC5', 0, 1628101573, None, 1628101573, ...  
['row-qfpb-ug4z.3duc', '00000000-0000-0000-3015-77149A271C21', 0, 1628101573, None, 1628101573, ...  
['row-eb3h.pbaf\_y8kc', '00000000-0000-0000-52DC-35CD360E0420', 0, 1628101573, None, 1628101573, ...  
['row-i7du.zvpi-8x85', '00000000-0000-0000-D2C3-A8FDC01A709C', 0, 1628101573, None, 1628101573, ...  
['row-yskw~iahu\_2xsn', '00000000-0000-0000-54EA-CFE607C99C4D', 0, 1628101573, None, 1628101573, ...  
['row-8ppp-7ek8-uhf0', '00000000-0000-0000-8837-B70C1FA65F13', 0, 1628101573, None, 1628101573, ...

## 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
```

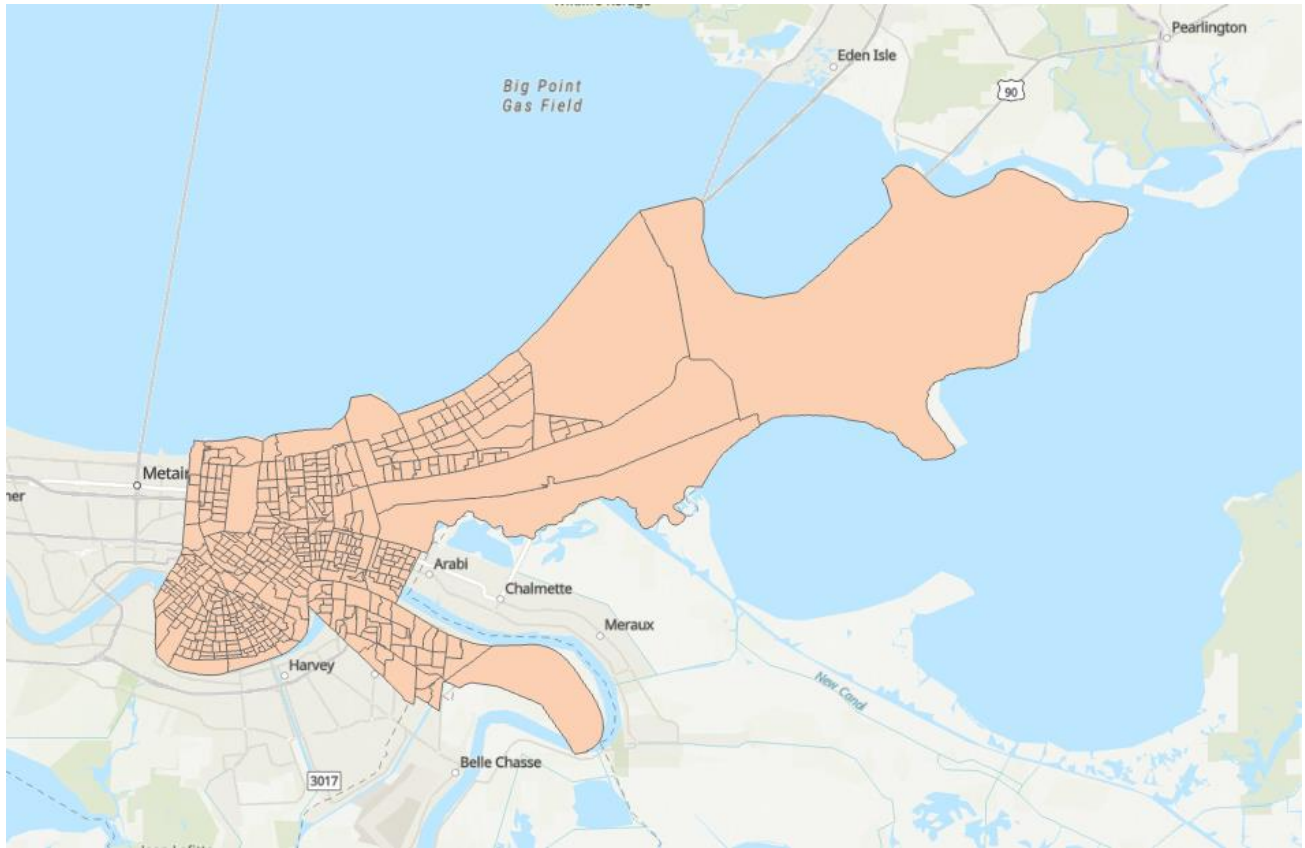
[45] ✓ 0.0s

```
... ['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



## 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 > ...
1  # Explore the json file
2  import arcpy
3  import json
4  import os
5
6
7  with open('no_tax.json','r') as file:
8      tax_json = json.load(file)
9
10
11  arcpy.FromMKT(tax_json['data']][8][8])
12  for row in tax_json['data']:
13      row[8] = arcpy.FromMKT(row[8])
14
15
16
17
18  ## Create a feature class and write fields
19  fcname = 'notax_fc.shp'
20  workspace = r'D:\Before_Hard_Drive\Study\Fifth_Semester_Fall2024\GIS_Programming_GEOG_4057\project1'
21  fc_fullname = os.path.join(workspace,fcname)
22  if arcpy.Exists(fc_fullname):
23      arcpy.management.Delete(fc_fullname)
24
25  arcpy.management.CreateFeatureclass(out_path=workspace,out_name=fcname,geometry_type='POLYGON',spatial_reference=4236)
26  ## add field names
27  fields = tax_json['meta']][['view']][['columns']]
28  for field in fields:
29      print(field['name'])
30      field_type = ['TEXT','TEXT','LONG','LONG','TEXT','LONG','TEXT','TEXT','TEXT','TEXT','TEXT','TEXT','TEXT']
31      field_names = []
32  for ind,field in enumerate(fields):
33      name = field['name']
34      if name == 'the_geom':
35          continue
36      if name.lower() == 'id':
37          name = f'id_{ind}'
38          max_len = min(10,len(name))
39          name = name[:max_len]
40          field_names.append(name)
41  field_names = [field.replace(" ","_") for field in field_names]
42  field_names = [field.replace(".", "_") for field in field_names]
43
44
45  for ind,field_name in enumerate(field_names):
46      arcpy.management.AddField(fc_fullname,field_name=field_name,field_type=field_type[ind])
47
48  field_names.append('SHAPE@')
49
50  ## Write data to the shapefile
51  with arcpy.da.InsertCursor(fc_fullname,field_names=field_names) as cursor:
52      for row in tax_json['data']:
53          new_row = []
54          for ind, value in enumerate(row):
55              if ind == 8:
56                  continue
57              if value == None:
58                  value = ""
59              new_row.append(value)
60          new_row.append(row[8])
61          cursor.insertRow(new_row)
```



## Run the python code

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER

aming_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.FromMKT(tax_json['data'][8][8])
13     for row in tax_json['data']:
14         row[8] = arcpy.FromMKT(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
```

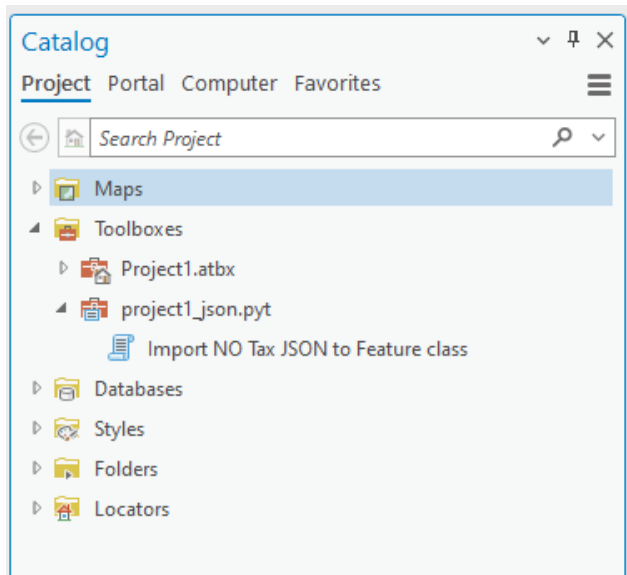
```
62 def main():
63     import sys
64     out_fc = sys.argv[1]
65     importNoTaxJSON(out_fc=out_fc)
66
67     if __name__ == '__main__':
68         main()
```

## 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:

```
3 import arcpy
4 from project1 import importNoTaxJSON
```

```
23
24     def getParameterInfo(self):
25         """Define the tool parameters."""
26         param_ws = arcpy.Parameter(
27             name='workspace',
28             displayName='Workspace',
29             direction='Input',
30             parameterType='Required',
31             datatype='DEWorkspace'
32         )
33         param_json = arcpy.Parameter(
34             name='json',
35             displayName='No Tax JSON',
36             direction='Input',
37             parameterType='Required',
38             datatype='DEFile'
39         )
40         param_out = arcpy.Parameter(
41             name='Output',
42             displayName='Output shapefile',
43             parameterType='Required',
44             direction='Output',
45             datatype='GPString'
46         )
47         params = [param_ws, param_json, param_out]
48         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
```

Geoprocessing

← Import NO Tax JSON to Feature class →

Parameters Environments ?

Workspace  
project1

No Tax JSON  
I:\2024\GIS\_Programming\_GEOG\_4057\project1\notax.json







Output shapefile  
notax\_fc5.shp

Run

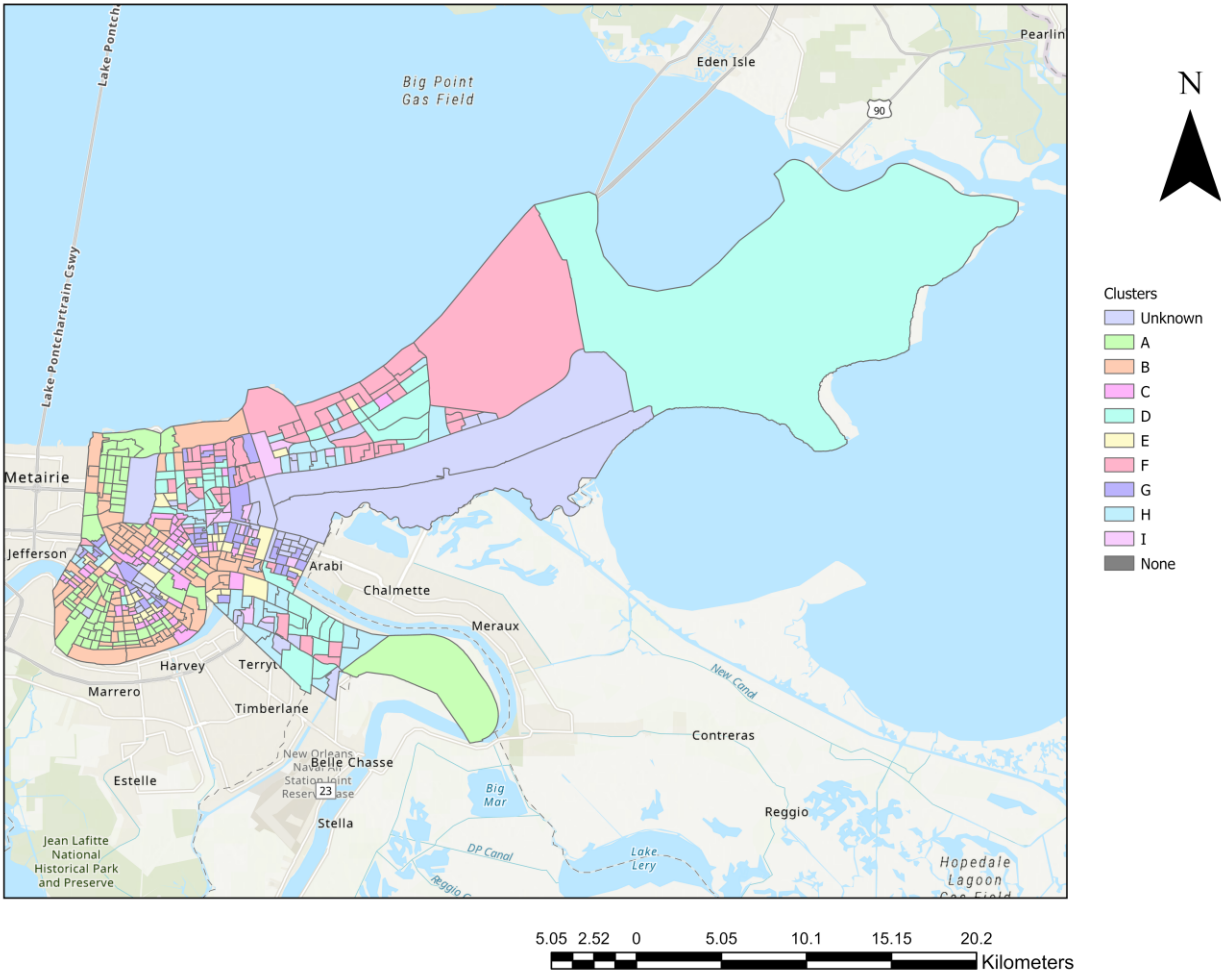
Import NO Tax JSON to Feature class completed.  
[View Details](#) [Open History](#)

Catalog Geoprocessing Element Label Class Locate

# 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](https://github.com/Mohammed-Elkharakany/project1_GEOG4057.git)