

Lab 1 - Minnesota Geospatial Commons

```
In [8]: import requests
import json
import pprint
import zipfile
import urllib
import arcpy
import pandas as pd
```

```
In [11]: #Setting up searching inquiry HTTP URL to extract data from website

tag = "ecological"
base_url = "https://gisdata.mn.gov/api/3/action/package_search?q=" # a search
package_information_url = base_url + tag # add them together to get the result
```

```
In [12]: # Apply Get request to the searching URL

package_information = requests.get(package_information_url, auth = ('user', 'p
print(package_information_url)
```

C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3\lib\site-packages\urllib3\connectionpool.py:1004: InsecureRequestWarning: Unverified HTTPS request is being made to host 'proxy.oit.umn.edu'. Adding certificate verification is strongly advised. See: <https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings>

InsecureRequestWarning,
https://gisdata.mn.gov/api/3/action/package_search?q=ecological

```
In [13]: #Parse json into dictionary
package_dict = json.loads(package_information.content)
#pprint.pprint(package_dict) #too long, not printed for now
```

```
In [14]: #Get a view of metatdata to find the zipfile ULR

for keys in package_dict['result']['results'][0]: # show metadata
    print(keys)
```

```
license_title
maintainer
relationships_as_object
private
maintainer_email
num_tags
id
metadata_created
metadata_modified
author
author_email
state
version
creator_user_id
type
resources
num_resources
tags
groups
license_id
relationships_as_subject
organization
name
isopen
url
notes
owner_org
extras
title
revision_id
```

```
In [82]: #Apply GET request to the zipfile URL

r = requests.get(package_dict['result']['results'][0]['resources'][3]['url'])
open('shp_geos_ecological_class_system.zip','wb').write(r.content)
print("extracting the content...")
```

extracting the content...

```
In [84]: #Download shapefile and unzip it

with zipfile.ZipFile('shp_geos_ecological_class_system.zip','r') as zip_ref:
    zip_ref.extractall('Data')
```

```
In [ ]:
```

```
In [3]: #Setting up a second searching inquiry HTTP URL to extract the second dataset

tag2 = "Minnesota Climate Divisions"
base_url2 = "https://gisdata.mn.gov/api/3/action/package_search?q=" # a search
package_information_url2 = base_url2 + tag2 # add them together to get the res
```

```
In [ ]: # Apply Get request to the searching URL and parse json into dictionary

package_information2 = requests.get(package_information_url2, auth = ('user',
print(package_information_url2)

package_dict2 = json.loads(package_information2.content) # parse json into di
#pprint.pprint(package_dict) #too long, not printed for now
```

```
In [14]: #Get a view of metatdata to find the zipfile ULR

for keys in package_dict2['result']['results'][0]: # show metadata
    print(keys)
```

```
license_title
maintainer
relationships_as_object
private
maintainer_email
num_tags
id
metadata_created
metadata_modified
author
author_email
state
version
creator_user_id
type
resources
num_resources
tags
groups
license_id
relationships_as_subject
organization
name
isopen
url
notes
owner_org
extras
title
revision_id
```

```
In [21]: #Apply GET request to the second zipfile URL

r = requests.get(package_dict2['result']['results'][0]['resources'][2]['url'])
open('shp_bdry_climate_divisions.zip', 'wb').write(r.content)
print("extracting the content...")
```

extracting the content...

```
In [22]: #Download the second shapefile and unzip it
with zipfile.ZipFile('shp_bdry_climate_divisions.zip', 'r') as zip_ref:
    zip_ref.extractall('Data')
```

In []:

```
In [1]: #Read the coordinate system information

fc1 = "Data/ecs_provinces_of_mn_v99a.shp"
desc = arcpy.Describe(fc1)
sr1 = desc.spatialReference
sr1
```

```
Out[1]:          type          Projected
        name  NAD_1983_UTM_Zone_15N
        factoryCode          26915
        linearUnitName          Meter
        GCS.name  GCS_North_American_1983
```

```
In [2]: #Read the coordinate system information

fc2 = "Data/climate_divisions.shp"
desc2 = arcpy.Describe(fc2)
sr2 = desc2.spatialReference
sr2
```

```
Out[2]:          type          Projected
        name  NAD_1983_UTM_Zone_15N
        factoryCode          26915
        linearUnitName          Meter
        GCS.name  GCS_North_American_1983
```

```
In [3]: # Changing the coordinate system of the Ecological Section of Minnesota shape  
# And save it as a new projected shapefile  
  
out1 = "Data/ecs_provinces_of_mn_v99a_projected.shp"  
out_coordinate_system = arcpy.SpatialReference('NAD 1983 UTM Zone 11N')  
  
arcpy.Project_management(fc1,out1,out_coordinate_system)
```

Out[3]: **Output**

Data\ecs_provinces_of_mn_v99a_projected.shp

Messages

Start Time: Tuesday, October 5, 2021 1:17:54 PM

Succeeded at Tuesday, October 5, 2021 1:17:54 PM (Elapsed Time: 0.50 seconds)

```
In [4]: # Perform spatial join on the climate divisions  
# and ecological section of minnesota shaepfiles  
# save it as new shapefile spatialjoined into geodatabase (transform the data  
  
arcpy.analysis.SpatialJoin("ecs_provinces_of_mn_v99a", "climate_divisions", r
```

Out[4]: **Output**

C:\Users\zhux0474\Desktop\Lab1_1\MyProject\MyProject.gdb\spatialjoined

Messages

Start Time: Tuesday, October 5, 2021 1:18:17 PM

Succeeded at Tuesday, October 5, 2021 1:18:18 PM (Elapsed Time: 1.20 seconds)

```
In [5]: # Convert the spatially joined shapefile to table csv and save it to Data fol  
  
arcpy.TableToTable_conversion("spatialjoined","Data","spatialjoined_tbl.csv")  
#
```

Out[5]:

Output

Data\spatialjoined_tbl.csv

Messages

Start Time: Tuesday, October 5, 2021 1:18:55 PM
Succeeded at Tuesday, October 5, 2021 1:18:55 PM (Elapsed Time: 0.24 seconds)

In [9]:

```
# Read the table csv as panda dataframe

tbl = "Data/spatialjoined_tbl.csv"
df = pd.read_csv(tbl)
```

In [10]:

```
#Print out the head of the table (load the data)

df.head()
```

Out[10]:

	OBJECTID	Join_Count	TARGET_FID	JOIN_FID	AREA	PERIMETER	ECSLTPY2_	EC
0	1	1	0	0	9.375321e+10	2.816036e+06		2.0
1	2	1	0	1	9.375321e+10	2.816036e+06		2.0
2	3	1	0	2	9.375321e+10	2.816036e+06		2.0
3	4	1	0	3	9.375321e+10	2.816036e+06		2.0
4	5	1	0	4	9.375321e+10	2.816036e+06		2.0

In []:

