In [96]:
```
#Sets environment
from arcgis import GIS
from arcgis.geocoding import batch_geocode
gis = GIS(username = 'sunderwood_intern', password = 'Yellowbird123$')
import pandas as pd
import arcpy
import os
import shutil
arcpy.env.workspace = r"C:\Users\sunderwood\PythonPrograms\PythonScriptingforArcGISPro"
```

In [97]:
```
#Gets content from GIS
NPULayer = gis.content.get('cfa8f05f467141bd9c55cb8411b08fbe')
NPULayer
```

Out[97]:

NPU 2021 Permits (https://coaplangis.maps.arcgis.com/home/item.html?id=cfa8f05f467141bd9c55cb8411b08fbe)
Auto update this layer of NPU 2021 Permints

Feature Collection by sunderwood_intern
Last Modified: October 31, 2020
0 comments, 7 views
(https://coaplangis.maps.arcgis.com/home/item.html?id=cfa8f05f467141bd9c55cb8411b08fbe)

In [98]:
```
#Imports toolbox to download layer from online
arcpy.ImportToolbox(r"C:\Users\sunderwood\PythonPrograms\PythonScriptingforArcGISPro\DownloadService")
```

Out[98]:
```
<module ''>
```

In [99]:
```
#Checks if shapefile folder exists. if not makes folder and downloads layer to shapefile folder. If so deletes fol
der, makes it and downloads shapefile
if not os.path.exists(r"C:\Users\sunderwood\PythonPrograms\PythonScriptingforArcGISPro\Shapefile"):
    os.makedirs(r"C:\Users\sunderwood\PythonPrograms\PythonScriptingforArcGISPro\Shapefile")
    arcpy.FeatureClassToFeatureClass_conversion(r"https://services5.arcgis.com/5RxyIIJ9boPdptdo/arcgis/rest/servic
es/NPU_2021_Permits/FeatureServer/0",r"C:\Users\sunderwood\PythonPrograms\PythonScriptingforArcGISPro\Shapefile",
r"2021Permits")
else:
    shutil.rmtree(r"C:\Users\sunderwood\PythonPrograms\PythonScriptingforArcGISPro\Shapefile")
    os.makedirs(r"C:\Users\sunderwood\PythonPrograms\PythonScriptingforArcGISPro\Shapefile")
    arcpy.FeatureClassToFeatureClass_conversion(r"https://services5.arcgis.com/5RxyIIJ9boPdptdo/arcgis/rest/servic
es/NPU_2021_Permits/FeatureServer/0",r"C:\Users\sunderwood\PythonPrograms\PythonScriptingforArcGISPro\Shapefile",
r"2021Permits")
```

In [104]:
```
#Checks if excel file exists, if not does table to excel conversion if so deletes old excel file then does new con
version
if not os.path.exists(r"C:\Users\sunderwood\PythonPrograms\PythonScriptingforArcGISPro\Permits2021Excel.xlsx"):
    arcpy.TableToExcel_conversion(r"C:\Users\sunderwood\PythonPrograms\PythonScriptingforArcGISPro\Shapefile\2021P
ermits.shp","Permits2021Excel.xlsx")

else:
    os.remove(r"C:\Users\sunderwood\PythonPrograms\PythonScriptingforArcGISPro\Permits2021Excel.xlsx")
    arcpy.TableToExcel_conversion(r"C:\Users\sunderwood\PythonPrograms\PythonScriptingforArcGISPro\Shapefile\2021P
ermits.shp","Permits2021Excel.xlsx")
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
c:\program files\arcgis\pro\Resources\ArcToolbox\toolboxes\Conversion Tools.tbx#TableToExcel_conversion.Initialize
Parameters.py in <module>

AttributeError: 'ToolValidator' object has no attribute 'isLicensed'

---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
c:\program files\arcgis\pro\Resources\ArcToolbox\toolboxes\Conversion Tools.tbx#TableToExcel_conversion.Initialize
Parameters.py in <module>

AttributeError: 'ToolValidator' object has no attribute 'isLicensed'
```

In [105]:
```
#Reads excel sheets as dataframes
NPUExcel = pd.read_excel(r"C:\Users\sunderwood\PythonPrograms\PythonScriptingforArcGISPro\Permits2021Excel.xlsx")
NPUNew = pd.read_excel(r"C:\Users\sunderwood\PythonPrograms\PythonScriptingforArcGISPro\ExcelAddresses1.xlsx")
NPUExcel
```

Out[105]:

	FID	Month1	AppType	NPU	Name	Address	Latitude	Longitude
0	0	02 - Feb	Mayor's Office of Special Events	A	Chastain Spring Arts Festival	215 W Wieuca Rd NW, Atlanta, Georgia, 30342	33.875199	-84.393661
1	1	04 - Apr	Board of Zoning Adjustments	A	V-19-49	3995 Randall Mill Rd NW, Atlanta, Georgia, 30327	33.863162	-84.428632
2	2	02 - Feb	Board of Zoning Adjustments	A	V-18-394	4323 Mount Paran Pkwy NW, Atlanta, Georgia, 30327	33.873279	-84.417369

In [106]:
```
NPUNew
```

Out[106]:

	Month1	AppType	NPU	Name	Address	Latitude	Longitude
0	abc	abc	abc	Chastain Spring Arts Festival	215 W Wieuca Rd NW, Atlanta, Georgia, 30342	NaN	NaN
1	abc	abc	abc	V-19-49	3995 Randall Mill Rd NW, Atlanta, Georgia, 30327	NaN	NaN
2	abc	abc	abc	V-18-394	4323 Mount Paran Pkwy NW, Atlanta, Georgia, 30327	NaN	NaN
3	abc	abc	abc	SD-18-53	4400 Northside Dr NW, Atlanta, Georgia, 30327	NaN	NaN
4	abc	abc	abc	sd-19-4	1347 W Wesley Rd NW, Atlanta, Georgia, 30327	NaN	NaN
5	abc	abc	abc	V-92-026	55 Park Place, Atlanta, GA 30303	NaN	NaN
6	abc	abc	abc	SD-190-73	55 Trinity Ave	NaN	NaN
7	abc	abc	abc	PR-190-79	33 Gilmer St SE Atlanta, GA 30303	NaN	NaN

In [107]:
```
NPUNew.fillna(0)
```

Out[107]:

	Month1	AppType	NPU	Name	Address	Latitude	Longitude
0	abc	abc	abc	Chastain Spring Arts Festival	215 W Wieuca Rd NW, Atlanta, Georgia, 30342	0.0	0.0
1	abc	abc	abc	V-19-49	3995 Randall Mill Rd NW, Atlanta, Georgia, 30327	0.0	0.0
2	abc	abc	abc	V-18-394	4323 Mount Paran Pkwy NW, Atlanta, Georgia, 30327	0.0	0.0
3	abc	abc	abc	SD-18-53	4400 Northside Dr NW, Atlanta, Georgia, 30327	0.0	0.0
4	abc	abc	abc	sd-19-4	1347 W Wesley Rd NW, Atlanta, Georgia, 30327	0.0	0.0
5	abc	abc	abc	V-92-026	55 Park Place, Atlanta, GA 30303	0.0	0.0
6	abc	abc	abc	SD-190-73	55 Trinity Ave	0.0	0.0
7	abc	abc	abc	PR-190-79	33 Gilmer St SE Atlanta, GA 30303	0.0	0.0

In [108]:
```
NPUExcel = pd.merge(NPUExcel, NPUNew, how='right', left_on=['Month1','AppType','Name','Address','Latitude','Longitude'], right_on = ['Month1','AppType','Name','Address','Latitude','Longitude'])
NPUExcel
```

Out[108]:

	FID	Month1	AppType	NPU_x	Name	Address	Latitude	Longitude	NPU_y
0	NaN	abc	abc	NaN	Chastain Spring Arts Festival	215 W Wieuca Rd NW, Atlanta, Georgia, 30342	NaN	NaN	abc
1	NaN	abc	abc	NaN	V-19-49	3995 Randall Mill Rd NW, Atlanta, Georgia, 30327	NaN	NaN	abc
2	NaN	abc	abc	NaN	V-18-394	4323 Mount Paran Pkwy NW, Atlanta, Georgia, 30327	NaN	NaN	abc
3	NaN	abc	abc	NaN	SD-18-53	4400 Northside Dr NW, Atlanta, Georgia, 30327	NaN	NaN	abc
4	NaN	abc	abc	NaN	sd-19-4	1347 W Wesley Rd NW, Atlanta, Georgia, 30327	NaN	NaN	abc
5	NaN	abc	abc	NaN	V-92-026	55 Park Place, Atlanta, GA 30303	NaN	NaN	abc
6	NaN	abc	abc	NaN	SD-190-73	55 Trinity Ave	NaN	NaN	abc
7	NaN	abc	abc	NaN	PR-190-79	33 Gilmer St SE Atlanta, GA 30303	NaN	NaN	abc

In [109]:
```
NPUExcel = NPUExcel.fillna(0)
NPUExcel
```

Out[109]:

	FID	Month1	AppType	NPU_x	Name	Address	Latitude	Longitude	NPU_y
0	0.0	abc	abc	0	Chastain Spring Arts Festival	215 W Wieuca Rd NW, Atlanta, Georgia, 30342	0.0	0.0	abc
1	0.0	abc	abc	0	V-19-49	3995 Randall Mill Rd NW, Atlanta, Georgia, 30327	0.0	0.0	abc
2	0.0	abc	abc	0	V-18-394	4323 Mount Paran Pkwy NW, Atlanta, Georgia, 30327	0.0	0.0	abc
3	0.0	abc	abc	0	SD-18-53	4400 Northside Dr NW, Atlanta, Georgia, 30327	0.0	0.0	abc
4	0.0	abc	abc	0	sd-19-4	1347 W Wesley Rd NW, Atlanta, Georgia, 30327	0.0	0.0	abc
5	0.0	abc	abc	0	V-92-026	55 Park Place, Atlanta, GA 30303	0.0	0.0	abc
6	0.0	abc	abc	0	SD-190-73	55 Trinity Ave	0.0	0.0	abc
7	0.0	abc	abc	0	PR-190-79	33 Gilmer St SE Atlanta, GA 30303	0.0	0.0	abc

In [113]:
```
os.remove(r"C:\Users\sunderwood\PythonPrograms\PythonScriptingforArcGISPro\Permits2021Excel.xlsx")
```

In [114]:
```
NPUExcel.to_excel(r"C:\Users\sunderwood\PythonPrograms\PythonScriptingforArcGISPro\Permits2021Excel.xlsx")
```

In [132]:
```
input_table = r"C:\Users\sunderwood\PythonPrograms\PythonScriptingforArcGISPro\Permits2021Excel.xlsx"
locator = "https://utility.arcgis.com/usrsvcs/servers/b4652533cc834fdf9d56af21a4c03b67/rest/services/World/Geocode
Server"
address_fields = "'Address'"
output_type = "Feature_CLASS"
output_folder = r"C:\Users\sunderwood\PythonPrograms\PythonScriptingforArcGISPro"
output_name = "NPU2021_geocoded"
```

In [133]:
```
for i, row in NPUExcel.iterrows():
    if row['Latitude'] == 0.0:
        arcpy.geocoding.GeocodeFile(input_table, locator, address_fields, output_type, output_folder, output_name)
```

```
---------------------------------------------------------------------------
ExecuteError                              Traceback (most recent call last)
<ipython-input-133-e1139de84116> in <module>
      1 for i, row in NPUExcel.iterrows():
      2     if row['Latitude'] == 0.0:
----> 3         arcpy.geocoding.GeocodeFile(input_table, locator, address_fields, output_type, output_folder, outp
ut_name)

C:\Program Files\ArcGIS\Pro\Resources\ArcPy\arcpy\geocoding.py in GeocodeFile(in_table, locator, address_fields, o
utput_type, output_location, output_name, country, location_type, category)
    995         return retval
    996     except Exception as e:
--> 997         raise e
    998
    999 @gptooldoc('GeocodeLocationsFromTable_geocoding', None)

C:\Program Files\ArcGIS\Pro\Resources\ArcPy\arcpy\geocoding.py in GeocodeFile(in_table, locator, address_fields, o
utput_type, output_location, output_name, country, location_type, category)
    992     from arcpy.arcobjects.arcobjectconversion import convertArcObjectToPythonObject
    993     try:
--> 994         retval = convertArcObjectToPythonObject(gp.GeocodeFile_geocoding(*gp_fixargs((in_table, locator, a
ddress_fields, output_type, output_location, output_name, country, location_type, category), True)))
    995         return retval
    996     except Exception as e:

C:\Program Files\ArcGIS\Pro\Resources\ArcPy\arcpy\geoprocessing\_base.py in <lambda>(*args)
    509         val = getattr(self._gp, attr)
    510         if callable(val):
--> 511             return lambda *args: val(*gp_fixargs(args, True))
    512         else:
    513             return convertArcObjectToPythonObject(val)

ExecuteError: Failed to execute. Parameters are not valid.
ERROR 000735: Input Table: Value is required
WARNING 002897: Performance of this locator can be improved.
Failed to execute (GeocodeFile).
```

In [ ]:
```
#if not os.path.exists(r"C:\Users\sunderwood\PythonPrograms\PythonScriptingforArcGISPro\NPUGeodatabasefile.gdb"):
   # arcpy.ExcelToTable_conversion("Permits2021Excel.xlsx", "NPUGeodatabasefile.gdb")
#else:
   # os.remove(r"C:\Users\sunderwood\PythonPrograms\PythonScriptingforArcGISPro\NPUGeodatabasefile.gdb")
   # arcpy.ExcelToTable_conversion("Permits2021Excel.xlsx", "NPUGeodatabasefile.gdb")


#Address2 = arcpy.ExcelToTable_conversion("ExcelAddresses2.xlsx","AddressV2.gdb")
```

In [ ]:
```
#for i, row in NPUExcel.iterrows():
   # if row['Latitude'] == 0.0:
       # arcpy.GeocodeAddresses_geocoding()
```

In [ ]:
```
#for i in NPUExcel.iterrows():
   # if ['Latitude'] == '0':
     #   results = batch_geocode(Address)
```

In [ ]:
```
#results
```

In [ ]:
```
#Converted 1st Excel Sheet to geodatabase
#Address1 = arcpy.ExcelToTable_conversion("ExcelAddresses1.xlsx","AddressV1.gdb")
```

In [ ]:
```
#Not working for some reason
#with arcpy.da.SearchCursor('AddressesV1.dbf','*') as cursor:
   # print(cursor.fields)
  #  for row in cursor:
   #     print (row)
```

In [ ]:
```
#FeatureClass_Item
```

In [ ]:
```
#Converts 2nd Excel Sheet to geodatabase
#Address2 = arcpy.ExcelToTable_conversion("ExcelAddresses2.xlsx","AddressV2.gdb")
```

In [ ]:
```
#Uses Pandas to read xlsx
#addressesdf1 = pd.read_excel(r'ExcelAddressesV1.xlsx')
#addressesdf1
```

In [ ]:
```
#Uses Pandas to read 2nd CSV
#addressesdf2 = pd.read_csv(r'ExcelAddressesV2.csv')
#addressesdf2
```

In [ ]:
```
#Does a join on the right data frame, df2, and keeps everything in both data frames and those exclusive to data fr
ame 2
#addressesdf1 = pd.merge(addressesdf1, addressesdf2, how='right', left_on=['Name','Address'], right_on = ['Nam
e','Address'])
#addressesdf1
```

In [ ]:
```
#Writes new dataframe 1 to excel sheet
#addressesdf1.to_excel("ExcelAddressesV1.xlsx")
```

In [ ]:
```
#Checking to see if dataframe was actually written to excel
#addressesdf1 = pd.read_excel(r'ExcelAddressesV1.xlsx')
#addressesdf1
```

In [ ]:
```
#Brings excel sheet into GIS to be compared to already existing layer
#NPUAddresses = arcpy.ExcelToTable_conversion = ("ExcelAddressesV1.xlsx", "NPUAddresses.gdb")
```

In [ ]:
```
#with arcpy.da.SearchCursor(r'NPUAddresses','*') as cursor:
   # print(cursor.fields)
  #  for row in cursor:
      # print (row)
```

In [ ]:
```
```

In [ ]:
```
```