Joseph Zaborek

GIS-5653

07/28/2021

An interactive Script to Generate Shapefiles Near Designated Wetland Areas in Georgia by Census Tract

- **Abstract:**

This project created a python script that serves as a tool to aide in creating shapefiles for further data exploitation and analysis. The script takes in census tract shapefiles from the U.S. Census Bureau, wetlands designation from the U.S. Fish and Wildlife Service, and a user input to specify a buffer distance in feet. Once all the inputs are secured as variables, the script creates a geo-database with all of the input files, a file created as a result of buffering the wetlands shapefile with the user's distance, and a version of the census tract shapefile clipped to the new buffer shapefile. The script returns to the user the area of the new buffer shapefile in square miles, and the file path to the newly created geo-database.

The tool is designed to save time for users who are trying to make judgements on properties or points of interest that need information on proximity to protected wetlands. This information is pertinent to investors speculating on the useability of properties based on what type of environmental mitigations may be legally required before construction can begin in the area.

- **Introduction**:

In the recent decades there has been a significant increase in visibility in the conservation and preservation of naturally occurring water-based resources in the United States. “The Federal Water Pollution Control Act of 1948 was the first major U.S. law to address water pollution. Growing public awareness and concern for controlling water pollution led to sweeping amendments in 1972. As amended in 1972, the law became commonly known as the Clean Water Act (CWA).” (U.S. EPA) Most states, including Georgia, use this updated act as the basis for state and municipal wetland regulation. Since 1972 there was a simultaneous increase in the remote sensing sciences, and the insight garnered through GIS has dramatically increased the U.S.’s ability to classify wetlands.

““Wetlands” mean those areas that are inundated or saturated by surface or groundwater at a frequency and duration sufficient to support, and that under normal circumstances do support, a prevalence of vegetation typically adapted for life in saturated soil conditions.” (Georgia DNR) The wetlands designations have increased so significantly, it becomes difficult for a non-expert to determine a property or location’s proximity to designated wetlands. This interactive script was designed to create custom dissolved shapefiles in order to provide users with the data they need to make determinations using the resultant data around whether or not they intersect, or are adjacent to, designated wetlands. It will also provide related census tract data so the user can be equipped with as much data as possible for decision making.

- **Materials and Methods**:

The script has a few configurable inputs. These various configurations allow the script to be updated in order to change the areas under study. As presented for the purposes of this project, the originating datasets are Georgia us census tract data and Georgia costal wetland shapefiles. These inputs can me modified to be any two census tract shapefiles and wetlands data from the U.S. Fish and Wildlife as long as the two inputs spatially overlap.

The reason for the test data being based in Georgia is due to the fact that the researcher recently began spending time in the central part of the state, and made an observation that compared to other states, there was a significant increase in posted designated wetlands. The researcher also was volunteering to assist in property investigation for purchasing purposes, and wetland and flood plane considerations were paramount to assist in that decision making. An easy way to determine those proximities or intersections was not readily available, and the researcher sought to remedy that situation.

A user is then prompted to provide a distance, including negative distances, to the script, and the script will reference designated Georgia wetlands and re-draw the wetland shapes with the provided distance buffer. The next step is that the script clips the census tract shapefile and saves the reduced census tract data as a new shapefile. The script then takes the new area under study and calculates the area of each involved census tract and presents the summed total are in square miles to the user.

The last portion of the script creates a new folder and places a new empty geo database in that folder, them migrates all of the files, source and resultant, into the new geodatabase. The new folder location is then returned to the user for archival or data exploitation purposes.

One final note about the script, in order to support readability, the various steps of the process were broken into functions and stored in a module.py file. The main.py file then calls the various functions in order to accomplish the goals of each run.

The overall workflow is described in Figure 1.1. The table containing the arcpy methods used can be located in Figure 1.2.

Figure 1.1

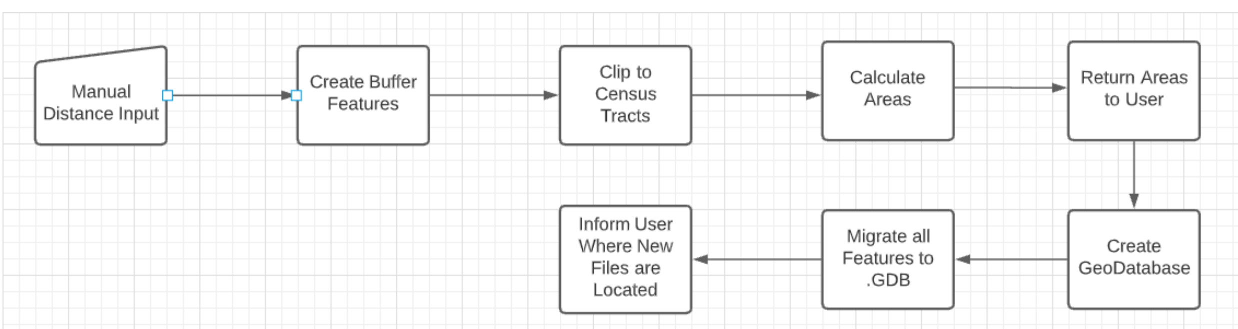

Figure 1.2

Methods Used	
ArcPy Methods	**OS Methods**
arcpy.Buffer_analysis	os.mkdir
arcpy.Clip_analysis	os.path.join
arcpy.AddField_management	
arcpy.CalculateField_management	
arcpy.da.SearchCursor	
arcpy.CreateFileGDB_management	
arcpy.ListFeatureClasses	
arcpy.CopyFeatures_management	

- **Results:**

The results can be summarized as a stepwise experiment that ultimately worked, but the development of the script and its associated process are where most of the lessons learned were gleaned. The resultant tool is able to fulfill the requirements laid out ahead of time and produces the output data as expected. Figure 2.1 shows the console output of a successful run of the tool. Figure 2.2 shows the main.py file. Figure 2.3 shows the module.py file.

Figure 2.1

```
Please enter a distance in feet to create a new shapefile around Georgia wetlands:100
Thank you! Please wait while we create your dataset.
Your new shape has an area of 6397.544999899167 square miles
Your files can be located at: C:\Users\Owner\Desktop\Spatial Programming\FinalProject\output\wetland_oututput.gdb
```

Figure 2.2

main.py - C:\Users\Owner\Desktop\Spatial Programming\FinalProject\main.py (2.7.18)

File Edit Format Run Options Window Help

```python
# main.oy
## Runs the GA wetland and buffering shapefile tool


##### Import system modules
import arcpy
import os
import module

# Set workspace
arcpy.env.workspace = "C:\\Users\\Owner\\Desktop\\Spatial Programming\\FinalProject"


# Start of module calls
user_in = module.user_input()

module.buff(user_in)

module.clip()

module.area()

module.area_return()

# Make Output Directory
my_dir = "C:\\Users\\Owner\\Desktop\\Spatial Programming\\FinalProject\\output"
os.mkdir(my_dir)
if not os.path.exists(my_dir):
    os.mkdir(my_dir)

module.file_mgmt()

module.file_mover()
```

Ln: 28 Col: 30

Figure 2.3

module.py - C:\Users\Owner\Desktop\Spatial Programming\FinalProject\module.py (2.7.18)

File Edit Format Run Options Window Help

```
import os

## Make Output Directory
def outDir():
    my_dir = "C:\\Users\\Owner\\Desktop\\Spatial Programming\\FinalProject\\output"
    os.mkdir(my_dir)
    if not os.path.exists(my_dir):
        os.mkdir(my_dir)


## Solicit user for distance input and ensure user enters an input less than one mile
def user_input():
    user_in = input("Please enter a distance in feet to create a new shapefile around Georgia wetlands:")
    try:
        if int(user_in) > 5280:
            print("Please enter a number less than 5280")
        else:
            print("Thank you! Please wait while we create your dataset.")

    except ValueError:
        print("Please Enter an Integer")
    return user_in

#Perform the buffering of the wetlands shapefile
def buff(x):
    ga_wetlands = "CoastalZoneManagementArea.shp"
    user_output_buffer = "buffer_output"
    arcpy.Buffer_analysis(ga_wetlands, user_output_buffer, str(x) + " Feet")


#Clip the census tract data, to the new wetland buffer shapefile
def clip():
    tracts = "tl_2019_13_tract.shp"
    arcpy.Clip_analysis("tl_2019_13_tract.shp", "buffer_output.shp", "out_clipped")


#calculate the total area of the new shapefile
def area():
    arcpy.AddField_management("out_clipped.shp", "Shape_area", "DOUBLE")
    exp = "!SHAPE.AREA@SQUAREMILES!"
    arcpy.CalculateField_management("out_clipped.shp", "Shape_area", exp, "PYTHON_9.3")

#Calculate the area in square miles and return to the user
def area_return():
    summed_total = 0
    with arcpy.da.SearchCursor("out_clipped.shp", "Shape_area") as cursor:
        for row in cursor:
            summed_total = summed_total + row[0]
    print("Your new shape has an area of " + str(summed_total) + " square miles")

#create an output geodatabase
def file_mgmt():
    arcpy.CreateFileGDB_management("C:\\Users\\Owner\\Desktop\\Spatial Programming\\FinalProject\\output", "wetland_output.gdb")

#list then Convert all shapefiles in a folder to geodatabase feature classes
def file_mover():
    fc_list = arcpy.ListFeatureClasses()
    out_workspace = "C:\\Users\\Owner\\Desktop\\Spatial Programming\\FinalProject\\output\\wetland_oututput.gdb"

    for shapefile in fc_list:
        # Determine the new output feature class path and name
        out_featureclass = os.path.join(out_workspace, os.path.splitext(shapefile)[0])
        arcpy.CopyFeatures_management(shapefile, out_featureclass)

    print("Your files can be located at: " + out_workspace)
```

Ln: 62 Col: 0

- **Discussion and Conclusions**:

The manual testing of the script was limited to “happy path” testing and really does not address any off-nominal uses of the tool. Once all of the happy-path defects were addressed, the script itself is quite useful and creates the shapefiles as intended.

Error handling and proper guidance regarding re-directing users when the tool is provided with incorrect values would be the most likely place to begin enhancements to the tool. The next largest focus area for further developing the tool would be in cleaning up the file handling. The script cannot run twice in a row without some manual folder and file cleanup. Adding a method to create unique output names would be very useful.

- **References:**

Rules of Georgia Department of natural resources ... (n.d.). http://www.georgiaplanning.com/documents/EnvironmentalPlanning/EnviroPlanningRulesEPD_DNR.pdf.

Environmental Protection Agency. (n.d.). *History of the Clean Water Act*. EPA. https://www.epa.gov/laws-regulations/history-clean-water-act.