**Lab Report**

Title: Lab 0

Notice: Dr. Bryan Runck

Author: Eric Gibson

Date: September 10, 2023

**Project Repository:**https://github.com/gibso632/GIS-5571.git

**Time Spent:** ~10 hours

**Abstract**

There are many ways to perform functions in ESRI products, but this lab will focus on three: ArcGIS Pro ModelBuilder, ArcGIS Pro Notebooks, and ArcGIS Online Notebooks. Using three different network datasets with varying levels of data density, a 100-meter buffer was added utilizing all three methods of performing the function mentioned above. This was done, not only for practice, but also for testing how the number of features in a feature class may affect some of these methods. It was found both the ModelBuilder and ArcGIS Pro Notebooks were fairly simple for me to use, with more issues arising with ArcGIS Online Notebooks. In ArcGIS Pro, there were no issues with the density of the dataset, however, ArcGIS Online could not execute a buffer on a feature class with more than 9999 features and, thus, the densest data could not be buffered. This could possibly be fixed by performing a dissolve on streets with the same name in the attribute table.

**Problem Statement**

There are many different ways to use ESRI systems for GIS and various other functions. Some of these include ArcGIS Pro, Jupyter Notebooks in ArcGIS Pro, and Jupyter Notebooks in ArcGIS Online. Within this Lab, I will compare and contrast the functionality between these three different ways of performing a simple functionality: creating a buffer for a feature dataset.

Table 1. Dataset metadata and preparation for use within this lab

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Requirement** | **Defined As** | **(Spatial) Data** | **Attribute Data** | **Dataset** | **Preparation** |
| 1 | High-density road network | Dakota County road network centerline data | Centerlines depicting all roads in Dakota County, Minnesota | Street Name, Street Type, ZipCode, City, Surface Type, StreetID | [Minnesota Geospatial Commons](https://gisdata.mn.gov/) Dataset | Use Project tool to change projection to UTM Zone 15 |
| 2 | Mid-density road network | Rice County road network centerline data | Centerlines depicting all roads in Rice County, Minnesota | Street Name, Street Type, ZipCode, City, Surface Type, StreetID | [Minnesota Geospatial Commons](https://gisdata.mn.gov/) Dataset | Use Project tool to change projection to UTM Zone 15 |
| 3 | Low-density road network | State Forest road system networks depicting roads accessible to the general public with any vehicle and roads which are only accessible to some of the population | Road networks depicting more accessible roads in the Minnesota State Forest Roads system in Aitkin County | Maintenance Level, Use Description, Maintenance Type, Private (Y or N), Distance (Miles) | [Minnesota Geospatial Commons](https://gisdata.mn.gov/) Dataset | Use Project tool to change projection to UTM Zone 15 |

**Input Data**

I chose three different datasets, all of which are road networks originally from the Minnesota Geospatial Commons. I wanted to originally get some data with different shapes (point, polyline, and polygon) to test how to buffer each of them, though I am sure the workflow is nearly the same. Due to each of the datasets existing of road networks, there are all in polyline format, however. Still, I tried to choose three different polyline datasets in different areas. For example, one of the datasets I chose was the road network for Dakota County, which is included in the Minneapolis-Saint Paul metropolitan area, meaning the number of roads are much greater and more dense than the dataset I chose showing State Forest roads. This should hopefully provide different examples of the workflow for creating a buffer in these three different ESRI products. Once again, I’m sure performing a buffer will be roughly the same workflow even with these three different datasets, but unfortunately I could not find road datasets with points or polygons. The only other dataset I found within the Minnesota Geospatial Commons with a different data type was a raster dataset.

Table 2. Simple description of datasets used within this lab

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Title** | **Purpose in Analysis** | **Link to Source** |
| 1 | Roads, Rice County, Minnesota | Dataset showing buffer application for a mid-density roads network | https://gisdata.mn.gov/dataset/us-mn-co-rice-trans-roads |
| 2 | Road Centerlines, Dakota County, Minnesota | Dataset showing buffer application for a high-density road network | https://gisdata.mn.gov/dataset/us-mn-co-dakota-trans-trans-streets |
| 3 | MNDNR Administered State Forest Roads | Dataset showing buffer application for a low-density road network | https://gisdata.mn.gov/dataset/trans-state-forest-roads-wheels |

**Methods**

ModelBuilder:

To begin, using ArcGIS Pro to perform a buffer is fairly simple. An underrated addition to ArcGIS Pro from ArcMap is the search bar. Instead of sifting through hundreds of different sections of tools, it is easy to search the buffer tool and perform a buffer on a feature class or shapefile. After finding the buffer, the user can simply input the feature class along with the radius of the buffer the user wants and a new layer will be created with the polygon buffer layer. Within the ModelBuilder, you can drag all the feature classes you need into the model, then search for the “Buffer” tool in the tools search bar, select and drag the buffer tool into the model, connect the feature classes to the buffer tool by adding an arrow from feature class to buffer, and finally open the buffer tool in the Model and select the radius of the buffer you want and make sure the output feature classes add to the map by selecting “Add to Display”. Click “Run” and it should create a buffer for all feature classes.

The screenshot of the model in ModelBuilder is presented below:

Figure 1. Screenshot of ModelBuilder in ArcGIS Pro for creating a 100-meter buffer for three different features

A screenshot of a computer

Description automatically generated

Notebook in ArcGIS Pro

While slightly more difficult since it utilizes Python, creating a buffer using a Jupyter Notebook in ArcGIS Pro is still simple. Instead of visually dragging and dropping the various components of the process into a model, code is used to essentially depict and perform the same process. First, a workspace is set up a place to add the output shapefiles, then a list is created with each feature class which needs to be buffered, finally a “for” loop is created to loop through each feature and create a buffer of 100 meters around each feature.

A screenshot of the code in a Jupyter Notebook in ArcGIS Pro is shown on the next page:

Figure 2. Screenshot of a notebook in ArcGIS Pro for creating a 100-meter buffer for three different features

A screenshot of a computer

Description automatically generated

Raw code is depicted below:

import arcpy

arcpy.env.workspace = (r'C:\Users\15612\Documents\GIS-5571\Lab0')

feat\_class\_list = ['trans\_roads','forest\_roads','trans\_trans\_streets']

for feat\_class in feat\_class\_list:

arcpy.analysis.Buffer(feat\_class,feat\_class + '\_buf','100 Meters')

Notebook for ArcGIS Online

After using the three different ways of performing a buffer, it is clear using notebooks in ArcGIS Online was the most difficult for me. I struggled with various warnings for not having authorization for buffers and not being able to display the buffers on a map. As far as I could tell, first the layers need to be added into ArcGIS Online and a notebook needs to be opened, then the area within ArcGIS Online where the layers are needs to be input to find and perform functions on them, next a buffer needs to be created using the use\_proximity.create\_buffers() function, and finally this should allow the user to visualize the buffers they created on a map created using the .map() function in the notebook. Unfortunately, I got an error displaying “RuntimeError: Cannot infer layer: will not be added to map”. From what I could tell, this is because Python API cannot understand the feature class. Another issue was the Dakota County Roads

feature due to the number of road segments, which reached over 9999 individual features. The

notebook could not handle that many features, so it seems the number of roads does affect if a

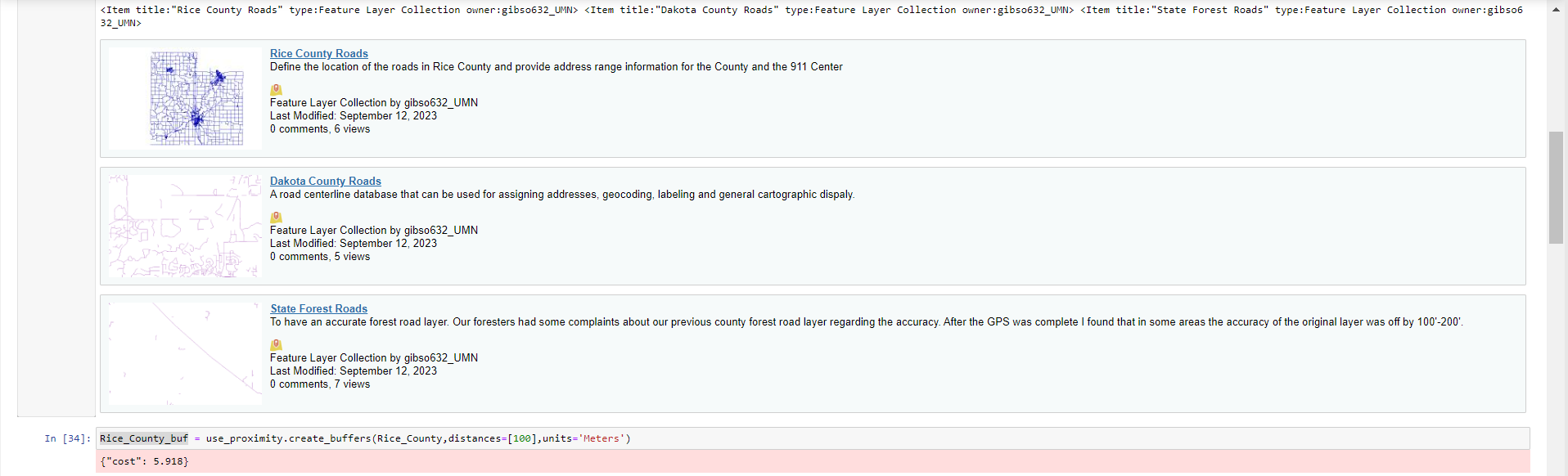
buffer can be created.

Below are some screenshots of my notebook in ArcGIS Online:

Figure 3. Screenshots depicting the ArcGIS Online notebook to for creating buffers on three feature classes

A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated

A white background with text

Description automatically generated

A map of the united states

Description automatically generated

A white background with blue text

Description automatically generated

Below is the raw code:

from arcgis.gis import GIS

gis = GIS("home")

from IPython.display import display

from arcgis.features import use\_proximity

me = gis.users.me

me

my\_content = me.items()

for item in my\_content:

if item.title == "Rice County Roads":

Rice\_County = item

if item.title == "Dakota County Roads":

Dakota\_County = item

if item.title == "State Forest Roads":

State\_Forest = item

print(Rice\_County,Dakota\_County,State\_Forest)

display(Rice\_County,Dakota\_County,State\_Forest)

Rice\_County\_buf = use\_proximity.create\_buffers(Rice\_County,distances=[100],units='Meters')

Dakota\_County\_buf = use\_proximity.create\_buffers(Dakota\_County,distances=[100],units='Meters')

State\_Forest\_buf = use\_proximity.create\_buffers(State\_Forest,distances=[100],units='Meters')

roads\_map = gis.map("Minnesota")

display(roads\_map)

buffer\_list = ['Rice\_County\_buf','Dakota\_County\_buf','State\_Forest\_buf']

for buffer in buffer\_list:

roads\_map.add\_layer(buffer)

display(roads\_map)

**Results**

ModelBuilder

Figure 4. Map of 100-meter buffers around roads in Rice County, Minnesota using ModelBuilder

A map of a city

Description automatically generated

Figure 5. Map of 100-meter buffers around State Forest roads in Aitkin County, Minnesota using ModelBuilder

A map of a road

Description automatically generated

ArcGIS Pro Notebook

Figure 6. Map of 100-meter buffers around roads in Dakota and Rice Counties, Minnesota using an ArcGIS Pro Notebook

A map of a city

Description automatically generated

Figure 7. Map of 100-meter buffers around State Forest roads in Aitkin County, Minnesota using an ArcGIS Pro Notebook

A map of a road

Description automatically generated

The four figures above where created two different ways: two were created via ModelBuilder and two were created via an ArcGIS Pro Notebook. I then color-coded the datasets so they would stand out from each other and all of them seemed to output the same features: a polygon shapefile extending 100-meters from the roads. This was fairly straightforward on both the ModelBuilder and Python in ArcGIS Pro sides allowing for various ways to perform this function.

I was unfortunately unable to create a clean output using ArcGIS Online and instead got an error message for having too many features in a feature class and I believe not using the correct data type for the Python API in ArcGIS Online. I believe it could be corrected by referencing the location of the feature class using the gis.content.get() function, but I am unsure and ran out of time.

**Results Verification**

After measuring and visualizing the buffers for the three different ways of performing the function, it is obvious the first two ways (ArcGIS Pro ModelBuilder and ArcGIS Pro Notebook) went smoothly without too many hiccups in the model and code. They also both output the same buffer feature classes displaying polygons which create a 100-meter buffers around the roads. I was able to determine this by using the measuring tool in ArcGIS Pro and measuring the width of the output polygons. Both of the outputs also verified each other, in a way, since they both output the exact same polygons.

Unfortunately, I was unable to output the correct buffer feature classes using the Notebooks in ArcGIS Online due to complications with my authorizations as well as how Python API recognizes certain types of files. I unfortunately got an error, but I am going to keep trying to figure out how to correctly run and output the right buffers.

**Discussion and Conclusion**

ArcGIS

The main thing I learned in this lab was how to perform Python functions in ArcGIS Pro. I had used ModelBuilder previously and I always knew about the different Python functions, but had never really used Python in my previous GIS experience. I am sure Python can help a lot with certain complex functions that cannot be performed by simply using a tool in ArcGIS Pro. I also like how you can save the Notebook as a Jupyter Notebook or essentially create a tool that performs a complex function you need to perform. I really wish I was able to figure out how to use the Notebooks in ArcGIS Online, but unfortunately I just ran out of time to figure it out. I will definitely continue to move forward in finding a solution.

The problem was a relatively straightforward one in the GIS community: create a buffer. I did in different ways to simply get an introduction on how to use various coding notebooks and models. This should help me to move forward with using Python for ArcGIS among other ways of performing functions.

GitHub

While fairly straightforward, I did encounter some issues when trying to get set up with GitHub. One of the largest issues I had was figuring out how to create folders for each lab within my GIS-5571 repository, as I did not see anything in the tutorials on how to do that, unless I missed something. It was easy to find the solution, though. When I searched how to create a folder in a GitHub repository, one of the first links was to a Stack Overflow help page where user ShadowFax mentioned to click on “Add File” on the repository’s page in GitHub and then input the folder name with a “/” at the end to make it a folder. They also mentioned you need to create a file to add into the folder since GitHub does not monitor empty folders. I also had some issues with some of the commands in Git Bash, as some of the commands mentioned in the tutorial did not seem to work, but I managed to find some work-arounds and come to a solution for creating the repository and the folders within it. Overall, however, setting up GitHub went well at least as far as I know with my inexperience in GitHub.

Link to repository: https://github.com/gibso632/GIS-5571.git

**References**

YouTube. (2021). *Getting Started with ArcGIS Notebooks (Part 1)*. *YouTube*. Retrieved September 12, 2023, from https://www.youtube.com/watch?v=QYRPH71TgV0.

**Self-score**

*Fill out this rubric for yourself and include it in your lab report. The same rubric will be used to generate a grade in proportion to the points assigned in the syllabus to the assignment.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Description** | **Points Possible** | **Score** |
| **Structural Elements** | All elements of a lab report are included **(2 points each)**:  Title, Notice: Dr. Bryan Runck, Author, Project Repository, Date, Abstract, Problem Statement, Input Data w/ tables, Methods w/ Data, Flow Diagrams, Results, Results Verification, Discussion and Conclusion, References in common format, Self-score | 28 | 28 |
| **Clarity of Content** | Each element above is executed at a professional level so that someone can understand the goal, data, methods, results, and their validity and implications in a 5 minute reading at a cursory-level, and in a 30 minute meeting at a deep level **(12 points)**. There is a clear connection from data to results to discussion and conclusion **(12 points)**. | 24 | 18 |
| **Reproducibility** | Results are completely reproducible by someone with basic GIS training. There is no ambiguity in data flow or rationale for data operations. Every step is documented and justified. | 28 | 22 |
| **Verification** | Results are correct in that they have been verified in comparison to some standard. The standard is clearly stated **(10 points)**, the method of comparison is clearly stated **(5 points)**, and the result of verification is clearly stated **(5 points)**. | 20 | 14 |
|  |  | 100 | 82 |