Lab Report

Title: Apples And Oranges: Comparing and Contrasting Esri Tools For Lab 0
Notice: Dr. Bryan Runck
Author: Cecelia Isaac
Date: 2/4/2021

Project Repository: https://github.com/CeceliaAi/GIS5572/tree/master

Abstract
This lab's objective is to use three separate ESRI tools to create the same output. We will then compare and contrast the process using models. The data will be obtained through the Minnesota Geospatial Commons. We will use the Rail Lines, Minnesota network data from MnDOT, Office of Freight and Commercial Vehicle Operations (OFCVO), Freight, Railroads, and Waterways Section. The three tools we will use are: ArcPro, Jupyter Notebooks in ArcPro, and Jupyter Notebooks in ArcOnline. The process for each of these tools will be recorded in the Methods section. A rose by any other name should smell as sweet, so the results should be the same, though they may look different in the various environments. In the conclusion, we will discuss what we learned throughout the process and from the outputs. Additionally, the conclusion will discuss part one of the lab, getting set up in GitHub.

Problem Statement
Esri tools allow for more than one way to skin a cat. This lab will use three of them to perform the same task, buffering a network dataset. Each environment will require the same basic steps (setting the environment, data entry, and buffering), but the details of each will be different and will require some small changes to certain steps. In ArcPro, we will have to properly set the conditions of the buffer tool. In ArcPy, we will have to set the environment to use the buffer function. In ArcGIS Online Notebooks, we will have to bring the tools into the coding environment. There is not much context for this lab, so we will buffer at a range of 10 nautical miles since that output will be easily identified.

Table 1. Analysis Requirements

#	Requirement	Defined As	Spatial Data	Attribute Data	Dataset	Preparation
1	Rail network	Raw input dataset from MNDOT	Rail road geometry		https://gisdata.mn.gov/dataset/trans-rail-lines	
2	Buffer tool or code	Tool in ArcPro or Python code		Distance of buffer		
3	Model Builder	Lucidchart				
4						

Input Data

This dataset was put together by the MnDOT Office of Freight and Commercial Vehicle Operations, Freight, Railroads, and Waterways Section. It is a current dataset that is continuously updated. The features were originally digitized from 1990 to 1995, and the data has been edited for accuracy. The data does have some gaps, and so it is best viewed at a smaller scale. These gaps in the polylines will not affect our work in this lab. We will download a zip file of the shapefile.

The attributes in this dataset include such columns as route number, railroad operator, railroad section and subdivision, and comments. For this lab, we will not use the attribute data in our analysis.

Table 2. Data

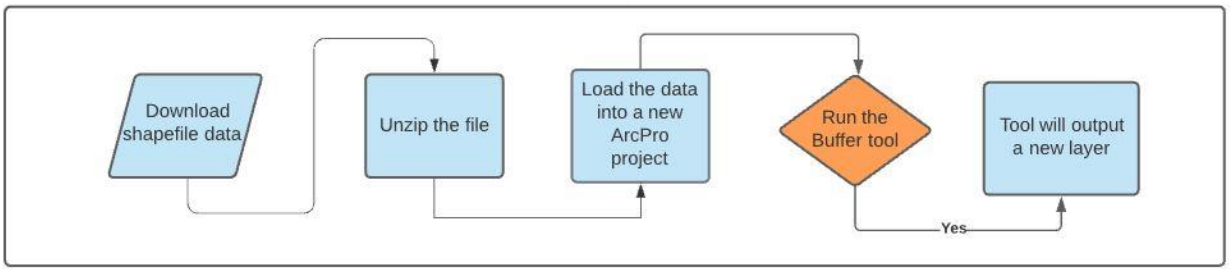
#	Title	Purpose in Analysis	Link to Source
1	Rail Lines, Minnesota	Raw input dataset for network analysis from MNDOT	https://gisdata.mn.gov/dataset/trans-rail-lines
2			
3			

Methods

ArcPro

To work with the data in ArcGIS Pro, download the Rail Lines, Minnesota shapefile (MnDOT, 2007) as a zip file and unzip it. We have chosen to represent these initial steps in the model (Figure 1) because even though they are basic, they are different for one of the methods. Create a new project in ArcPro, and open a new map. Load the data into the map. It will also appear in the geodatabase. In the Geoprocessing pane, open the Buffer Tool. Use the rail lines layer as the input, name the new output layer, and select 10 for the distance and Nautical Miles for the units. Keep all other defaults the same. Run the tool. The output will appear both in the geodatabase, and as a layer on the map. Symbolize the layer so the rail line is still visible underneath.

Figure 1. Data flow diagram.

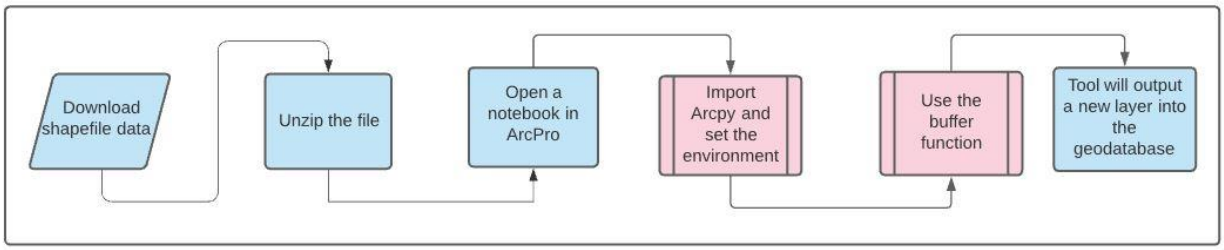


Jupyter Notebooks in ArcPro

As above, download and unzip the shapefile. In ArcPro, open a new notebook from the Analysis tab. In the cell, import the arcpy module and set the folder environment if necessary. I used the same project and geodatabase as the previous step, so I did not need to set an environment. Run the cell. In the next cell, add the buffer function (Esri, 2021) with the appropriate parameters. In this case, that means specifying the layer, the output layer name, the distance and the units. The other parameters have default settings that do not need to be altered. Run the cell. The output

should confirm that the program was successful. Additionally, the project geodatabase will have the new layer added to it automatically (Figure 2).

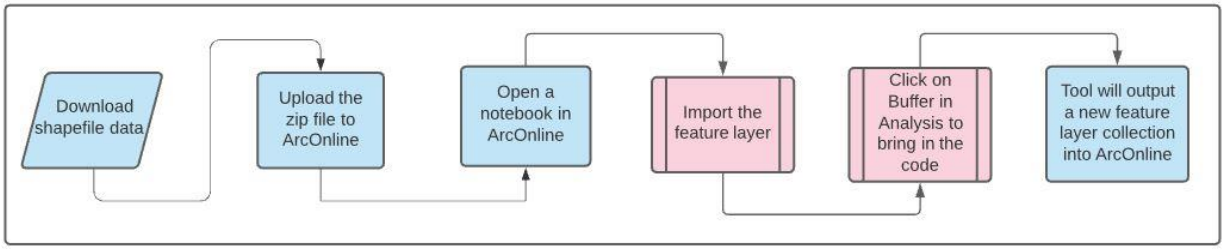
Figure 2. Data flow diagram



Jupyter Notebooks in ArcOnline

In order to use the data in the ArcOnline environment, it must be downloaded and then uploaded into the Content section (Figure 3). Once this is done, a new notebook can be opened. The new notebook automatically runs an import function for certain features. Using the add button, search the zip file location. The file will have become two files, a shapefile and a feature layer. Bring the feature layer in to the notebook. Adding the feature will bring the correct code into the cell. Change the variable names to something relevant. Hit run to actually execute the code. To buffer, search for the buffer tool in the Analysis tab. As with above, the correct code will appear in a new cell, but some details will need to be changed (Esri, 2021). In this case, the parameters must be set to those relevant to our data. Then the cell can be run and the output (a feature layer collection) will appear in the Content section.

Figure 3. Data flow diagram



Results

In this section we will compare the output of the three formats. Table 3 (below) offers a quick overview of the options and their processes.

ArcPro

In ArcPro the tool produced a buffer layer of 10 nautical miles around the rail line features. As specified, the output did not dissolve and the ends were rounded (Figure 4). The original rail line layer was symbolized with railroad track symbology, and the buffer layer was put into a light pink so the rail lines would be visible underneath. The buffer also had a light gray border in order to distinguish the places were segments ended and overlapped. Compared to the other ways to complete this analysis, ArcPro is the easiest to use.

Figure 4. ArcPro Output

Jupyter Notebooks in ArcPro

In Jupyter for ArcPro, the code produced a buffer layer identical to the first part of the lab. The only difference was that this layer had no specific symbology. The code is displayed below (Figure 5). This part of the lab output into the same geodatabase as the first part, and so we were able to verify that the code ran correctly because the two layers were visually identical. This code was simple, but out of the three options, writing code will likely still have the highest barrier to entry, so this format is advanced.

Figure 5. Jupyter for ArcPro Output

```
In [1]:
1 import arcpy
2 #arcpy.env.workspace = "C:\Users\celia\Desktop\shp_trans_rail_lines"

In [6]:
1 arcpy.analysis.Buffer("rail_lines",
2                       "rail_lines_Buffer3", "10 NauticalMiles")

Out[6]:
```

Output

C:\Users\celia\Documents\ArcGIS\Projects\GIS5572_Lab0\GIS5572_Lab0.gdb\rail_lines_Buffer3

Messages

Start Time: Wednesday, February 3, 2021 11:06:48 PM
WARNING 000635: Skipping feature 149 because of NULL or EMPTY geometry.
WARNING 000635: Skipping feature 222 because of NULL or EMPTY geometry.
WARNING 000635: Skipping feature 223 because of NULL or EMPTY geometry.
WARNING 000635: Skipping feature 243 because of NULL or EMPTY geometry.
WARNING 000635: Skipping feature 244 because of NULL or EMPTY geometry.
WARNING 000635: Skipping feature 245 because of NULL or EMPTY geometry.
WARNING 000635: Skipping feature 246 because of NULL or EMPTY geometry.
Succeeded at Wednesday, February 3, 2021 11:06:49 PM (Elapsed Time: 0.97 seconds)

```
In [ ]:
1
```

Jupyter Notebooks in ArcOnline

The code in the ArcOnline environment produced a feature layer collection as the output (Figure 6). It displayed inline as well as appearing in the Content section. When opened in the map viewer on AGOL, the map output matched what was produced in the other sections of the lab. It is also possible to edit the layer from the map viewer, or to return to the notebook to write more code. This environment is designed to be more user-friendly, and it does lower the barrier to entry, so it is intermediate difficulty compared to the other options.

Figure 6. Jupyter for AGOL Output

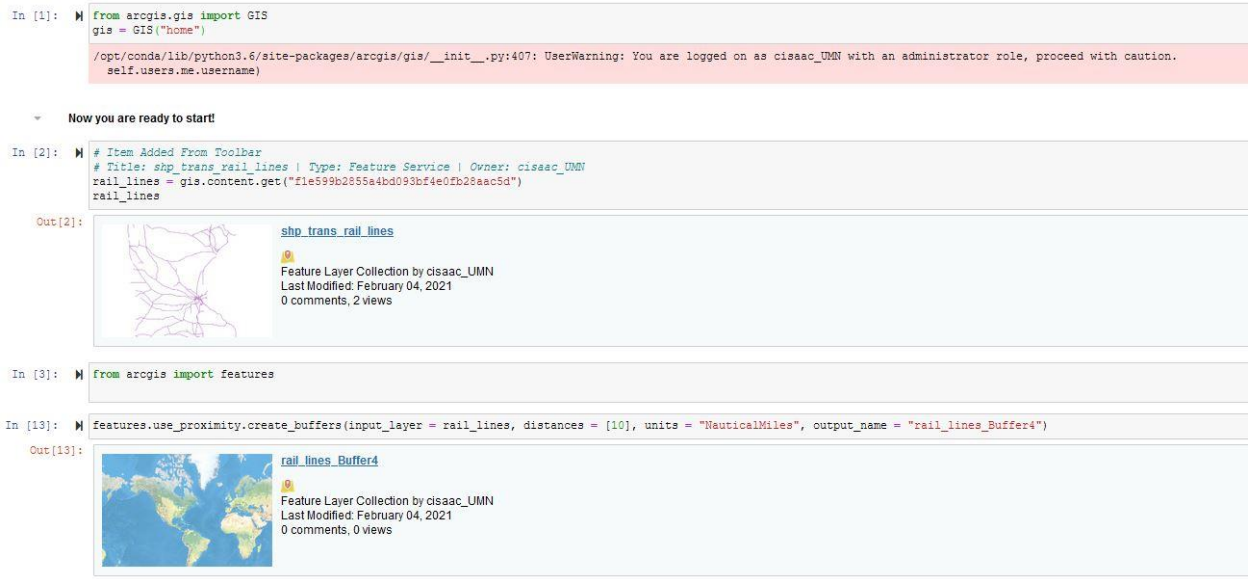


Table 3. Results

Method	Tool	Output	Ease of interpretation
ArcPro	Buffer tool in the software	Layer into the map and geodatabase	Easy; layer automatically outputs, errors are in plain text
Jupyter for ArcPro	Buffer function from Arcpy	Layer into the geodatabase	Advanced; errors are in coding jargon, output does not automatically visualize for assessment
Jupyter for AGOL	Buffer function in AGOL notebooks	Layer automatically adds to user's content	Intermediate; tool is easy to use but help for errors is hard to find

Results Verification

The first sign of correct results is that no errors resulted from running the tools or the code. The next step is to check the output. Whether it appeared in the geodatabase or the ArcOnline Content section, a new layer should be visible. Opening up that layer, it should be possible to click the feature, and a popup with the attribute data will appear. In the case of this lab, it is also possible to compare the three outputs, and verify that they look the same.

Whether or not the results are good is another question. This lab used nautical miles, a likely impractical way to buffer railroads. When creating the buffer, the default of No Dissolve was used, but it is possible dissolving the features might be most helpful, depending on the context.

Discussion and Conclusion

Mapping

This lab covered the processes for spatial data analysis in three different formats. I learned the most in the Jupyter for AGOL section, since I had not used it before. I believe the intention is to make coding more accessible, but one barrier was finding documentation. Since the project is newer, I was not sure where to look at first when I ran into an error.

The main problem of the lab was navigating the different platforms. This was not much of an issue since the actual task was straightforward and I have preformed it in two of the three formats already. In terms of comparing and contrasting the methods, my preference is still for ArcPro, since it is easy for me not only to complete the buffer, but also to symbolize and otherwise put finishing touches on the map. In the other formats, I would have had to look up the code I wanted, so it would have taken much longer to do something that is simple in Pro.

I do like that AGOL notebooks provided a middle ground for new coders, but don't count your chickens before they hatch, because the pre-made functions are not very powerful in and of themselves. It is my preference to become more comfortable with ArcPy scripting so that I can make customizable, complicated maps, not just use what is provided by Esri in a canned format.

GitHub

I was already partially set up with GitHub from a Unix Shell workshop. That workshop was only two days though, so while I have an account, I am still getting used to navigating it. Organizing a repo and pushing/pulling are straightforward. For me the tricky part is the Git Bash/CMD/etc programming. I have Git Desktop as well for when I am not able to figure something out, but am looking forward to becoming more comfortable with accessing data through Bash.

References

Esri. *Buffer (Analysis)* [Documentation]. Retrieved February 4, 2021, from https://pro.arcgis.com/en/pro-app/latest/tool-reference/analysis/buffer.htm

Esri. *Get Started With Notebooks* [Documentation]. Retrieved February 4, 2021, from https://doc.arcgis.com/en/arcgis-online/get-started/components-of-the-notebook-editor.htm

Esri. *Notebooks in ArcGIS Pro* [Documentation]. Retrieved February 4, 2021, from https://pro.arcgis.com/en/pro-app/latest/arcpy/get-started/pro-notebooks.htm

MnDOT. *Rail Lines, Minnesota* [Shapefile]. Minnesota Geospatial Commons. Retrieved February 4, 2021, from https://gisdata.mn.gov/dataset/trans-rail-lines

Self-score

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	**24**
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	**28**
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	**20**
		100	**100**