**Lab Report**

Title: Lab00

Notice: Dr. Bryan Runck

Author: Jake Ford

Date:09-02-2022

**Project Repository:** <https://github.com/ThisFord/GIS5571-arc1.git>

**Google Drive Link: n/a**

**Time Spent:** 30 hours

**Abstract**

Using a compare and contrast method, this lab report assess the effectiveness of three tools, ArcPro, ArcPro with Jupyter Notebooks, and ArcOnline with Jupyter Notebooks with a simple buffer operation. (*Get Started with Notebooks—ArcGIS Online Help | Documentation*, n.d.; *Manage Data in ArcGIS Online—ArcGIS Online Help | Documentation*, n.d.; *Notebooks in ArcGIS Pro—ArcGIS Pro | Documentation*, n.d.) The tools are compared in three categories: ease of use, efficiency and speed, and flexibility. Road center line data for all roads in Minnesota in 2012, sourced from the Minnesota Geospatial Commons, is used as the test dataset.(*County Boundaries in Minnesota - Minnesota Geospatial Commons*, n.d.; *Roads, Minnesota, 2012 - Minnesota Geospatial Commons*, n.d.) The data is then run through a buffer operation in each of the tool environments, with the goal of outputting a new buffer layer and comparing the processes and user experiences of each tool. (*Arcgis.Features.Use\_proximity Module*, n.d.) Each of the tools was able to complete the buffer operation, with significant differences in operability. Overall, this project finds that ArcPro is the most user friendly, especially for simple one-off analysis, ArcPro with Jupyter Notebooks is the most powerful, repeatable, and flexible; with ArcOnline being restricted to light use in the online environment.

**Problem Statement**

Modern GIS tools are diverse and constantly evolving. To assess the effectiveness of different technical approaches, this lab will compare and contrast performing a simple buffer operation on a single road network data set sourced from Minnesota Geospatial Commons using three different tools:  ArcGIS Pro, Jupyter Notebooks in ArcGIS Pro, and Jupyter notebooks in Arc Online. The tools will be compared in three categories: ease of use, efficiency and speed, and flexibility.

*Table 1. Material and data Requirements*

| **#** | **Requirement** | **Defined As** | **(Spatial) Data** | **Attribute Data** | **Dataset** | **Preparation** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Road network | Raw input dataset from MNDOT | Road geometry | na | [Mn GeoSpatial Commons](https://gisdata.mn.gov/dataset/trans-roads-mndot-tis) | download and unzip |
| 2 | ArcPro | standard GIS software GUI for geospatial analysis | base map and crs | na | ESRI Library | access to software and license |
| 3 | ArcPro Jupyter Notebook | Python programming interface included in ArcPro v3.0 | base map and crs | na | Esri Library | access to ArcPro v3.0 |
| 4 | ArcOnline Notebook | Jupyter labs interface for Arc Online | na | na | na | access to ArcOnline |
| 5 | Web Hosted Feature Layer | Feature layer containing centerline road data for buffer operation input | road geometry | na | [Mn GeoSpatial Commons](https://gisdata.mn.gov/dataset/trans-roads-mndot-tis)   derived web feature layer | upload shapefile used in desktop app |
| 6 | County Boundaries | Hennepin county boundary shapefile | boundary geometry | county names | [County Boundaries](https://gisdata.mn.gov/dataset/bdry-counties) | export Hennepin County as its own feature class for clipping extent |

**Input Data**

The data set used for the buffer operation compare-and-contrast is sourced from the Minnesota Geospatial Commons. The database represents historic road centerlines for the entire state of Minnesota from 2012. Roads are segmented from intersection to intersection and the data contains attributes are assigned based on the individual route. The key attributes included are Route System, Route Number and Name. [(*Roads, Minnesota, 2012 - Minnesota Geospatial Commons*, n.d.)](https://www.zotero.org/google-docs/?GppJPd) A complete metadata file is included in the downloaded geodatabase which includes a key to attribute information.  For the purposes of this comparison, the full statewide set of Federal and State roads were used for the buffer operation. Having a large data set to perform the operation on was key in testing the processing speed of the individual tools. Jupyter Notebooks in Arc Online required a smaller subset of the statewide data set to complete the buffer operation. A subset consisting of one route (route\_name = 35) was unable to handle the full data set due to credit limitations

*Table 2. Data Used*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Title** | **Purpose in Analysis** | **Link to Source** |
| 1 | DOT Roads | Raw input dataset for buffer analysis from MNDOT | [Mn GeoSpatial Commons](https://gisdata.mn.gov/dataset/trans-roads-mndot-tis) |
| 2 | MN County Boundaries | Boundary shapefile from MNGeo data clearing house as clipping mask. | [County Boundaries](https://gisdata.mn.gov/dataset/bdry-counties) |
| 3 | Web hosted Feature Layer Clipped DOT Roads | Input data set for buffer analysis in ArcOnline compatible format | [ArcOnlineContent](https://services.arcgis.com/8df8p0NlLFEShl0r/arcgis/rest/services/Hennepin_Co_Clip_Feature/FeatureServer) |

**Methods**

**ArcPro:**

Figure one shows the straight-forward data flow used in the standard ArcPro Graphical User Interface. The data was downloaded from the MN Geo website and unzipped in the Lab0 folder where the ArcPro project files are initialized. The folder containing the data was linked to the project through the catalog interface in ArcPro. The data set is then added to the map. To perform the buffer: the Federal and State feature class is selected as the input, the pairwise buffer operation is selected, and the output is a new feature class with the 20 m buffer applied to all the road centerlines. The run button executes the operation, and the output is displayed as a layer in the Map contents section of ArcPro.

*Figure 1. Data flow diagram for ArcPro GUI.*

*Diagram, chat or text message

Description automatically generated*

**ArcPro with Jupyter Notebooks:**

Figure 2 shows that the data flow for using Jupyter Notebooks in ArcPro remains exactly the same. The user interface has changed to a python script format with the input’s variables set as parameter of the function, as seen below in Figure 3. Here the input data is called in the function as a string referring to the file name, the output feature class name is defined by a string, and the buffer distance is also set using a string as the input type. Using the drop-down menu, the program is run by selecting the Run All option, the output is represented as a buffer feature layer with the assigned output name in the ArcPro contents pane.

*Figure 2. Data flow diagram for ArcPro with Jupyter Notebook.*

*Diagram, chat or text message

Description automatically generated*

*Figure 3. Jupyter Notebook interface in ArcPro with the code to run a buffer operation.*

Graphical user interface, text, application

Description automatically generated

**ArcOnline with Jupyter Notebooks:**

Figure 4 shows the data flow from ArcOnline with Jupyter Notebooks. The processing limitations required a drastic downsizing of the data set in ArcPro to be compatible with the online format. The data was uploaded along with a county boundary data set from MN Geo. The Road data was converted with a feature class to feature class process to select for only Route 35. The resulting layer was then clipped to the boundary of Hennepin County, the resulting feature layer was uploaded to ArcOnline. Once the feature layer was uploaded the buffer transformation was applied sing the code in figure 5. The process was considerably more involved than using a desktop interface and slower to the point where the operation couldn’t be performed with the existing large raw data set.

*Figure 4. Data flow diagram for ArcOnline with Jupyter Notebook.*

Diagram

Description automatically generated

*Figure 5. Notebook with code from ArcOnline with Jupyter. Graphical user interface, text, email

Description automatically generated*

**Results**

*Figure 6: 20-meter buffer results with Arc Pro*

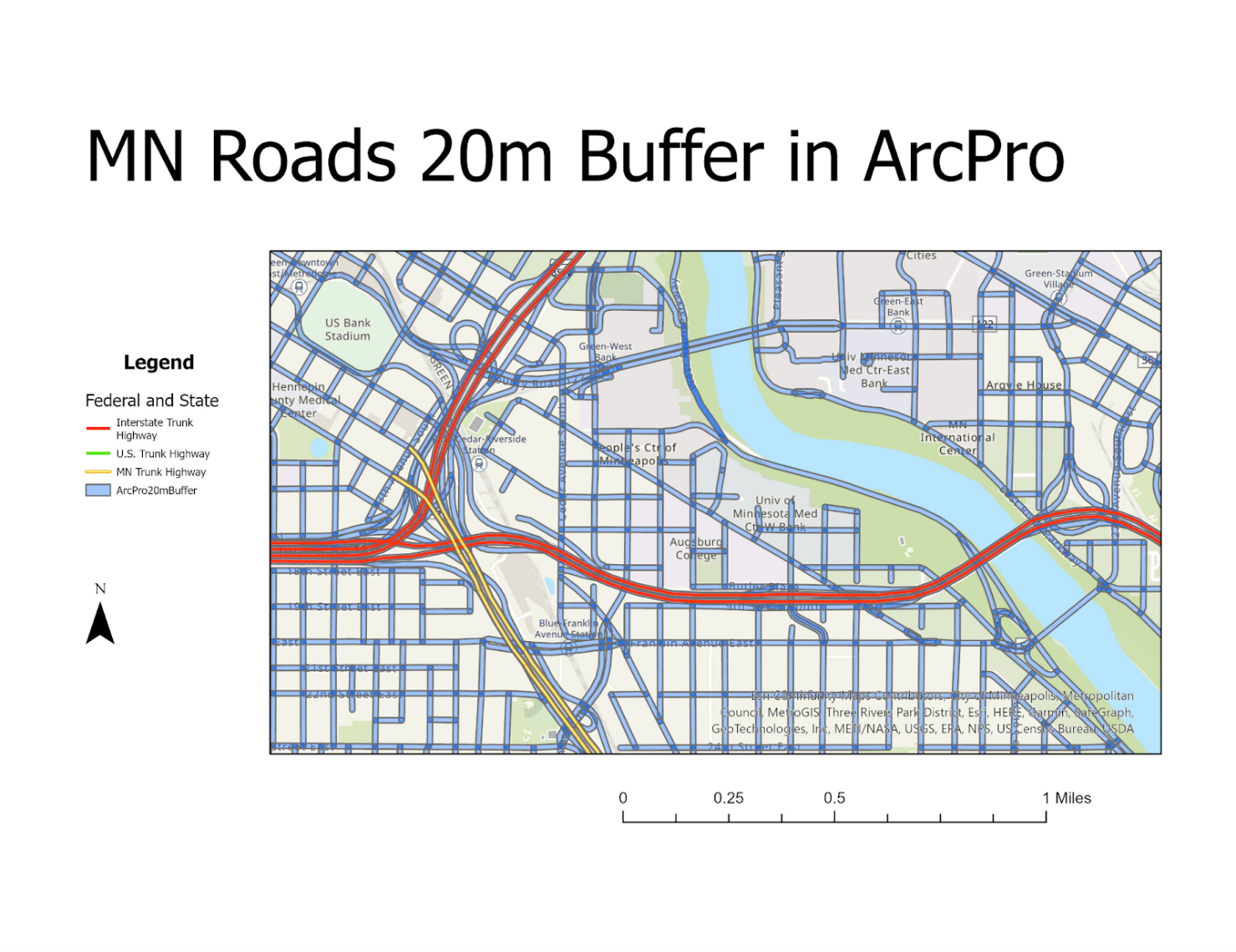
**

Figure 6 shows the results of the buffer operation in ArcPro as a static map, with the extent centering around the U of M campus. Each road segment has been successfully buffered, represented in blue grey above. The user interface was intuitive for this tool as a user with some experience using mapping software. The process was efficient and fast, taking the least amount of time to run of the three tools. The result was not programmed to be repeatable or automated. There is little flexibility beyond the drop-down menus, which contributes to the ease of use but could be limiting for more complex procedures.

*Figure 7: 10-meter buffer results with Arc Pro with Jupyter Notebooks*

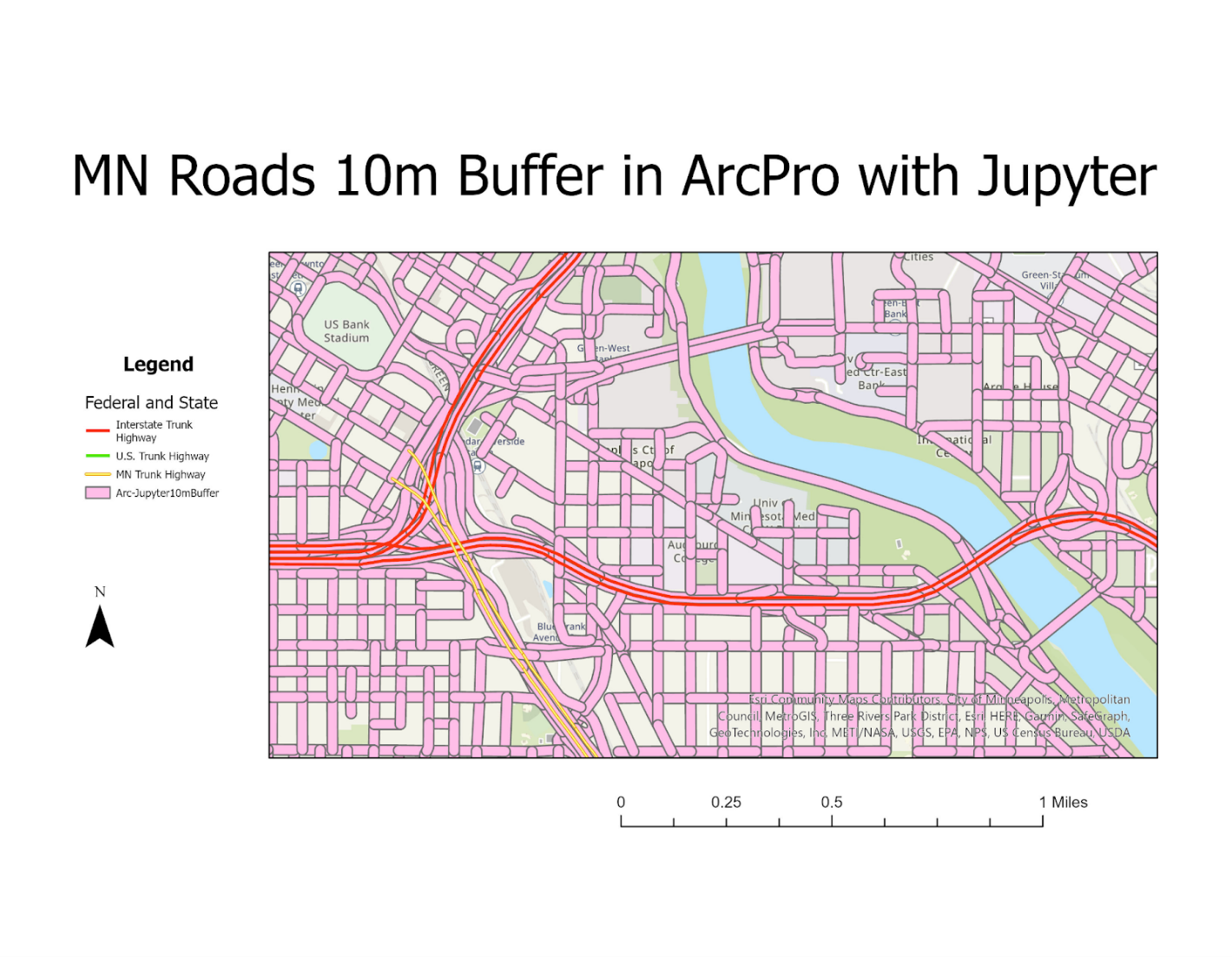
**

Figure 7 shows the results of the buffer operation in ArcPro with Jupyter Notebook as a static map, with the extent centering around the U of M campus. Each road segment has been successfully buffered, represented in pink above. These results are identical to the standard user interface as expected. The process was efficient and fast, though the operation took longer than in ArcPro standard interface, the time difference was not significant. This tool performed the operation successfully on the large data set, making it more robust than the limited ArcOnline performance. The short operation can be programmed to be repeatable and automated. The custom process can be saved externally and used in other projects as well. The various input parameters also allow for a wide range of customization making this the most flexible option of the three interfaces.

*Figure 8: buffer operation with ArcOnline and Jupyter Notebooks*

MN Roads 10m Buffer Arc Online with Jupyter Notebooks

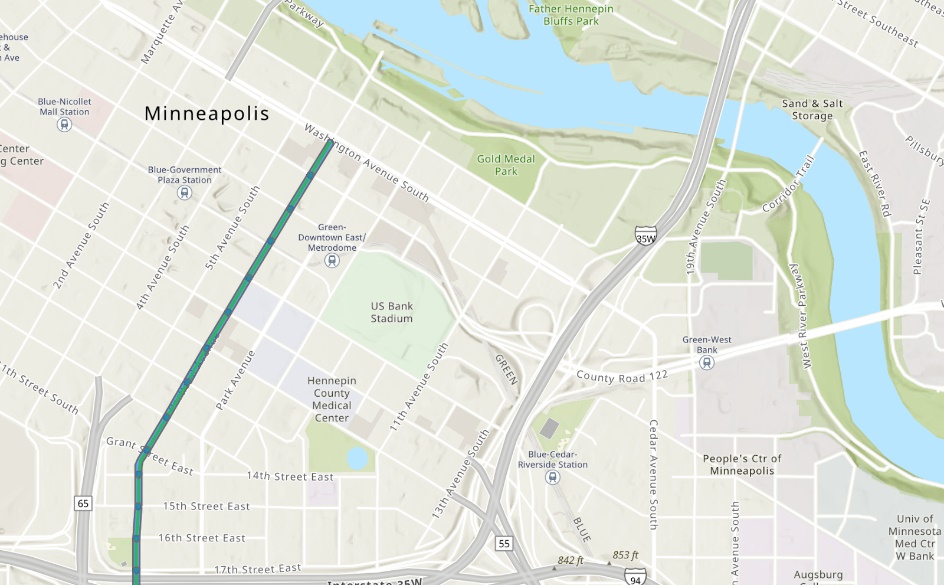
**

Figure 8 shows the results of the buffer operation in ArcOnline with Jupyter Notebooks as a clipping from an online dynamic map, with the extent centering around the U of M campus. Each road segment in the significantly reduced input data set has been successfully buffered, represented in green above. These results are similar to the two previous methods, but with great limitations on input data set size imposed by Esri credit restrictions. The interface is relative friendly with point and click coding producing readymade functions and methods. This tool took the longest to perform the operation, even on a much smaller data set, and has limited functionality as all data input must come through a web hosted feature layer.

**Results Verification**

Successful completion of the buffering operation under the given parameters with each tool demonstrates that each tool is capable of the given task, and that the tasks are repeatable. Direct experience as a user of each of the tools further verifies the qualitative assessments of the experiment.

**Discussion and Conclusion**

All three tools were able to complete the operation with ArcOnline and Jupyter Notebooks requiring a significant reduction to the size of the input data set. The first two tools are nearly identical in out put and run time, with the ArcPro GUI being the most user friendly, with point and click and drop-down menu options. ArcPro with Jupyter Notebooks has greater flexibility and functionality, with the bonus of being automatable and reproducible. However, it is not as ready to use by a novice. Some understanding of python is required, the auto complete options are helpful, but not exhaustive. ArcOnline ended up being limited in functionality and costly to run, both in time and ESRI credits. The online tool seems to be tailored to light operations on preexisting data sets for a user with some python familiarity. The point and click functions here present as many problems as they solve with dense documentation obscuring the ease and functionality. Overall, ArcPro is the most user friendly, especially for light analysis, ArcPro Notebooks is the most powerful and flexible, with ArcOnline being restricted to light use in the online environment.

**Git Hub**

Getting started with git hub was a huge learning curve. I went off script as I had WSL/ ubuntu installed, and had to figure out what the differences were with that and git for windows, it’s been a journey. Had to set up vs code in the command line, and think I got it working so I can use the regular windows gui interface to program remotely in wsl. However after much exploration I decided to scrap running everything in a separate operating system and switched to the Git for Windows program and have had much more success. Using multiple machines has allowed me to explore the usefulness of git and practice the install and set up features. Originally I had mis-named my clone file and didn’t realize it, so I kept getting “no parent/master branch” warnings, which was hard to debug. I kept plugging away, realized my mistake, re-did the clone and things worked well from there! Must pay careful attention to what directory you are in otherwise commands won’t work.

Lab0 Git Repository: <https://github.com/ThisFord/GIS5571-arc1.git>

**References**

*Arcgis.features.use\_proximity module*. (n.d.). ArcGIS API for Python. Retrieved September 21, 2022, from https://developers.arcgis.com/python/api-reference/arcgis.features.use\_proximity.html

*County Boundaries in Minnesota—Minnesota Geospatial Commons*. (n.d.). Retrieved September 21, 2022, from https://gisdata.mn.gov/dataset/bdry-counties

*Get started with notebooks—ArcGIS Online Help | Documentation*. (n.d.). Retrieved September 20, 2022, from https://doc.arcgis.com/en/arcgis-online/get-started/components-of-the-notebook-editor.htm

*Manage data in ArcGIS Online—ArcGIS Online Help | Documentation*. (n.d.). Retrieved September 20, 2022, from https://doc.arcgis.com/en/arcgis-online/manage-data/data-in-online.htm

*Notebooks in ArcGIS Pro—ArcGIS Pro | Documentation*. (n.d.). Retrieved September 20, 2022, from https://pro.arcgis.com/en/pro-app/latest/arcpy/get-started/pro-notebooks.htm

*Roads, Minnesota, 2012—Minnesota Geospatial Commons*. (n.d.). Retrieved September 20, 2022, from https://gisdata.mn.gov/dataset/trans-roads-mndot-tis

**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 |