Lab Report

Title: Lab 2 Part 1
Notice: Dr. Bryan Runck
Author: Eric Gibson
Date: 10/31/2023

Project Repository: https://github.com/gibso632/GIS5571/tree/main/Lab2
Google Drive Link:
Time Spent: 24 hours

Abstract

This lab describes multiple analyses performed to visualize multidimensional, including 2-dimensional, 3-dimensional, and 4-dimensional, data. The goal is to provide two PDFs depicting a digital elevation model (DEM) and a triangular irregular network (TIN) of analyzed LiDAR data, a 3D scene of the same LiDAR data, and an animation of monthly precipitation averages based on data taken over 30 years from 1991 to 2020. The LiDAR data gathered is of point cloud data captured in an area near Belle Plaine, Minnesota. The monthly precipitation raster data used to create the animation is based on PRISM 30-year normals with a 4-kilometer spatial resolution. In order to create an automated and reproduceable analysis, an ArcGIS Pro Jupyter Notebook was used and any "tools" mentioned in this lab were performed through Arcpy unless otherwise specified. All deliverables were eventually produced showing the different ways multidimensional information can be illustrated within a GIS framework. It was concluded this lab provided much needed experience and knowledge of manipulation of multidimensional data and provided support for moving forward with these ideas.

Problem Statement

This lab will be divided into three sections to combine various concepts from the class in an effort to build experience and knowledge for building an ETL pipeline using raster data from Landsat imagery. This includes converting a Landsat DEM to a TIN and exporting it to PDF using Arcpy, making a side-by-side comparison of a 2D map and 3D scene including the various features provided for both, and creating a spacetime cube with annual 30-Year Normals from PRISM. Converting the DEM to a TIN works as an introduction to analysis on rasters using Arcpy, as most of the analysis done so far has been on vector data. Performing the comparison between both a straightforward 2D map and a 3D Arcscene map provides useful information of different performance and analysis capabilities of either Arc product. Finally, creating a spacetime cube will allow for an understanding of the temporal aspects of data and allow for a clear visualization of geographical changes over time.

Table 1. Requirements for performing Lab 2 Part 1.

#	Requirement	Defined As	(Spatial) Data	Attribute Data	Dataset	Preparation
1	Two PDFs depicting a DEM and a TIN created via LiDAR data	One PDF illustrating the result of using LiDAR point cloud data to create a DEM and another PDF illustrating a TIN created from the DEM.	Point cloud elevation data	Elevation and time	Minnesota LiDAR Data	Gather data through the Minnesota Geospatial Commons API and display in ArcGIS Pro using Arcpy.
2	2D and 3D map created from LiDAR data	Screenshot of the ArcScene visualization of the 3D map.	Point cloud elevation data within an ArcScene 3D visualization	Elevation and Time	Minnesota LiDAR Data	Prepare ArcScene window for adding 3D data.
3	PRISM Annual Normals Precipitation Data	Spacetime Cube and a GIF depicting the change in monthly precipitation averages throughout the United States between 1991 and 2020.	Raster depicting precipitation averages with a 4-kilometer spatial resolution throughout the contiguous US	Monthly precipitation averages and date data	PRISM Data	Add any libraries to perform the analyses.

Input Data

There are two datasets needed for this lab and both will be utilized to present a raster dataset once the analyses are completed. The first dataset is LiDAR data taken from the Minnesota Geospatial Commons in a .las file. The specific .las dataset used was gathered from an area just west of Belle Plaine, Minnesota, located a few miles southwest of Minneapolis. The data contains point cloud data with elevations matching each point for possibly converting to a DEM and a TIN, which will be done within this lab. This data was captured by various individual pulses of light from the LiDAR scanner, which reflected off the ground back to the scanner and the distance from the ground was then calculated to find the ground surface elevation. This is why LiDAR data creates point clouds instead of a raster, each point represents one of these energy pulses.

The other dataset is a raster containing annual data from 1991-2020 in .bil format downloaded from PRISM containing precipitation information for the entire contiguous United States. This dataset is a typical raster with pixels containing precipitation values at a 4 km spatial resolution. There are also multiple rasters needed within this dataset in order to create the mosaic dataset needed later in the lab. Each raster corresponds to each year between 1991 to 2020, to create a space-time cube after forming the mosaic dataset. This means the required attribute information within each raster are the precipitation information and the year in which the raster was created.

Table 2. Datasets used for analyses within this lab.

#	Title	Purpose in Analysis	Link to Source
1	4342-12-05.las	Point cloud data with elevations for creating a DEM and TIN.	[LiDAR Data]
2	Various PRISM Precipitation .bil files	Creating a mosaic dataset to be converted to a spacetime cube for an animated visualization of precipitation data from 1991 to 2020	[PRISM Precipitation Annual Normals]

Methods

Figure 1. Data flow diagram illustrating the process for creating a PDF of a DEM and a TIN using a LiDAR dataset.

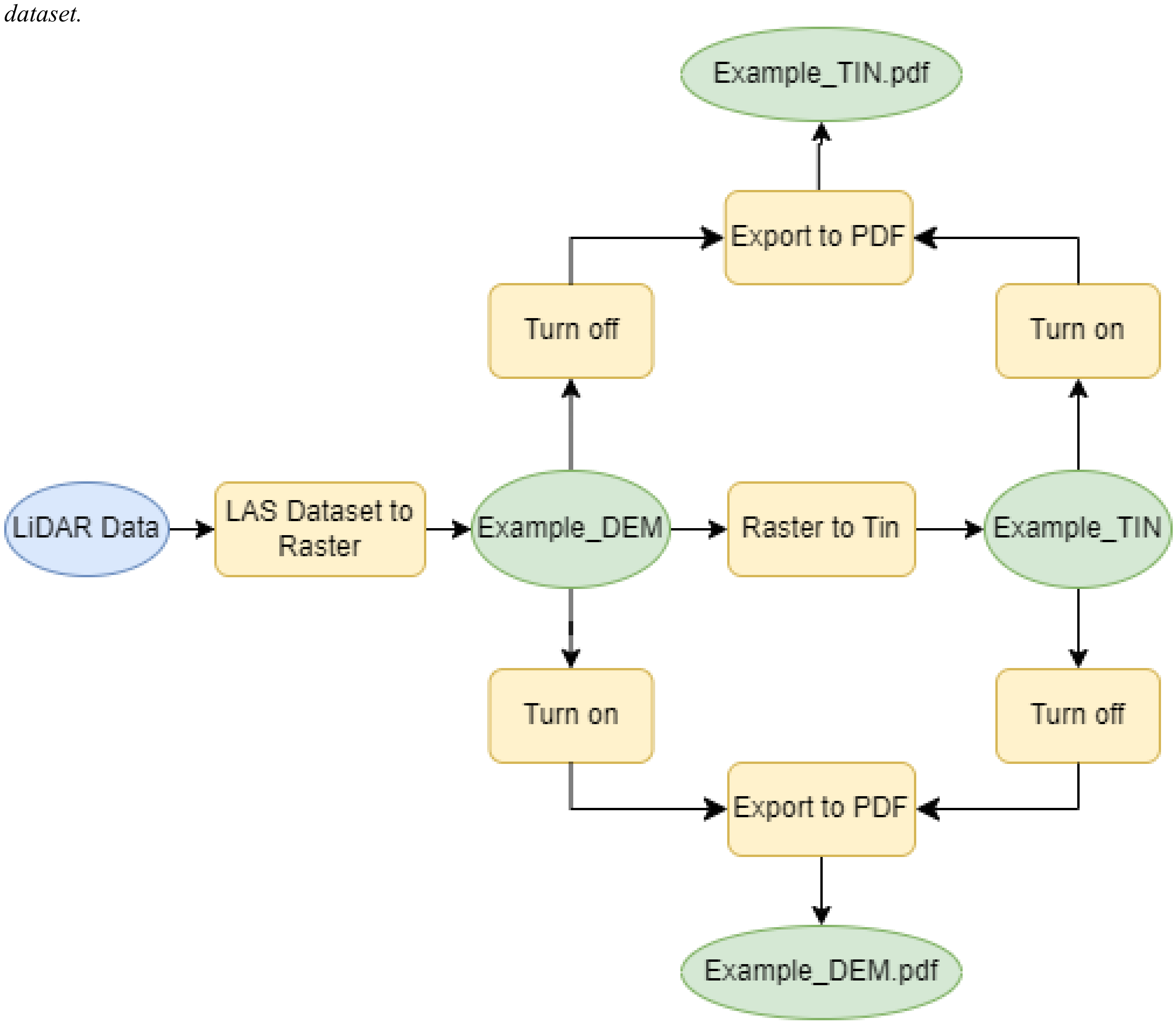


Figure 2. Data flow diagram illustrating the process for creating a 3D representation of LiDAR data.

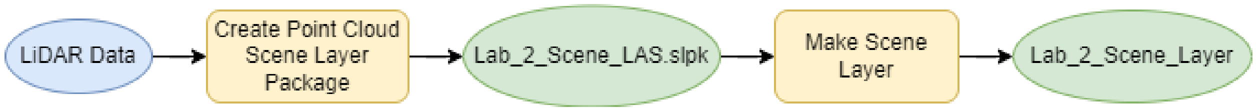
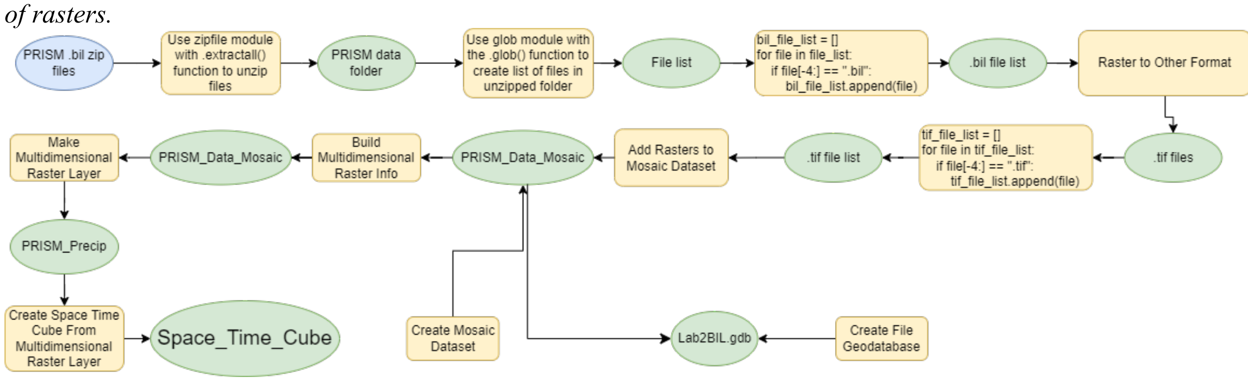


Figure 3. Data flow diagram illustrating the process for creating an animation from a spacetime cube using a series of rasters.



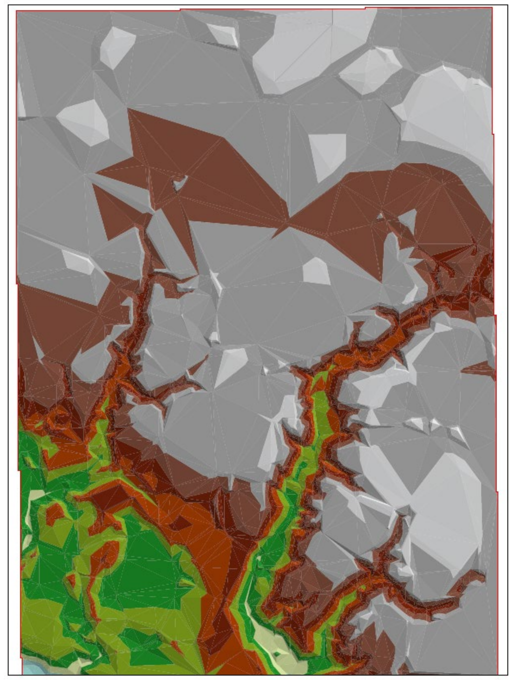
In addition to what is depicted within **Figure 3**, some functions were added to clean up the mosaic dataset and prepare the attribute table for creating the multidimensional raster layer. This included removing excess rows in the attribute table of the mosaic dataset footprint which were not needed and adding a variable field (each value was simply "PRISM") and a time dimension field (each value was the month date corresponding to each raster) within the attribute table. The addition of these two fields enabled the mosaic dataset to be used within the "Build Multidimensional Raster Info" tool and finally create the Spacetime Cube.

Results

<u>Section 1</u>

Beginning with Section 1 in Part 1 of Lab 2 (depicted by **Figure 1**), the process was fairly straightforward, especially since LiDAR data already contains accurate information for creating a DEM and a TIN. Once the "LAS Dataset to Raster Tool" was run through Arcpy, a DEM was created and, from there, it is fairly simple to just run the "Raster to TIN" tool via Arcpy as well to output a TIN. Finally, after adding a layout, first the TIN was turned off, while the DEM was turned on in the Contents pane using an "if" statement and the layer.visible function. The result was then exported to a PDF using the "Export to PDF" tool through arcpy once again. The TIN layer was then turned on and the DEM was then turned off to create another PDF depicting the TIN instead. The PDFs can be viewed in GitHub using the link at the beginning of the lab, but below are screenshots of the created PDFs.

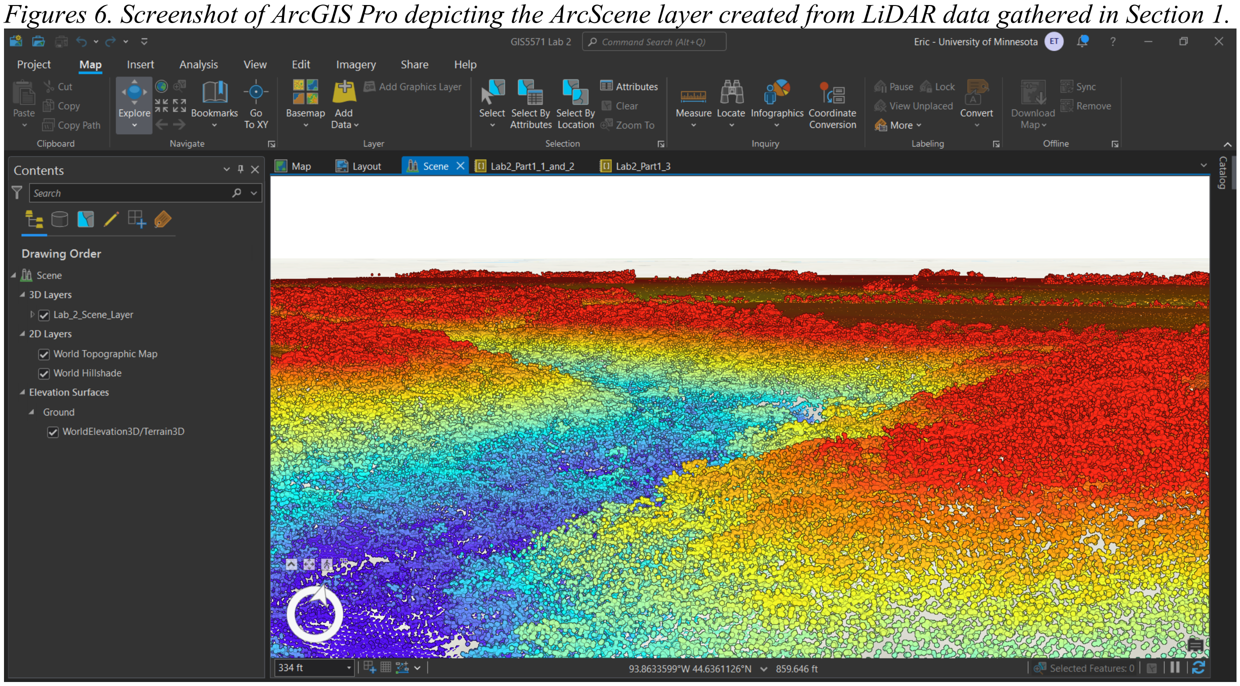
Figures 4 and 5. Screenshots of PDFs depicting a DEM and a TIN created via the acquired LiDAR data from the Minnesota Geospatial Commons.



<u>Section 2</u>

Moving on to Section 2 in Part 1 of Lab 2, the setup for creating the 3-dimensional map within ArcScene was not too much. Through using the “Create Point Cloud Scene Layer Package” and “Make Scene Layer” tools using the gathered LiDAR data, a 3D map was created and depicted below in a screenshot:

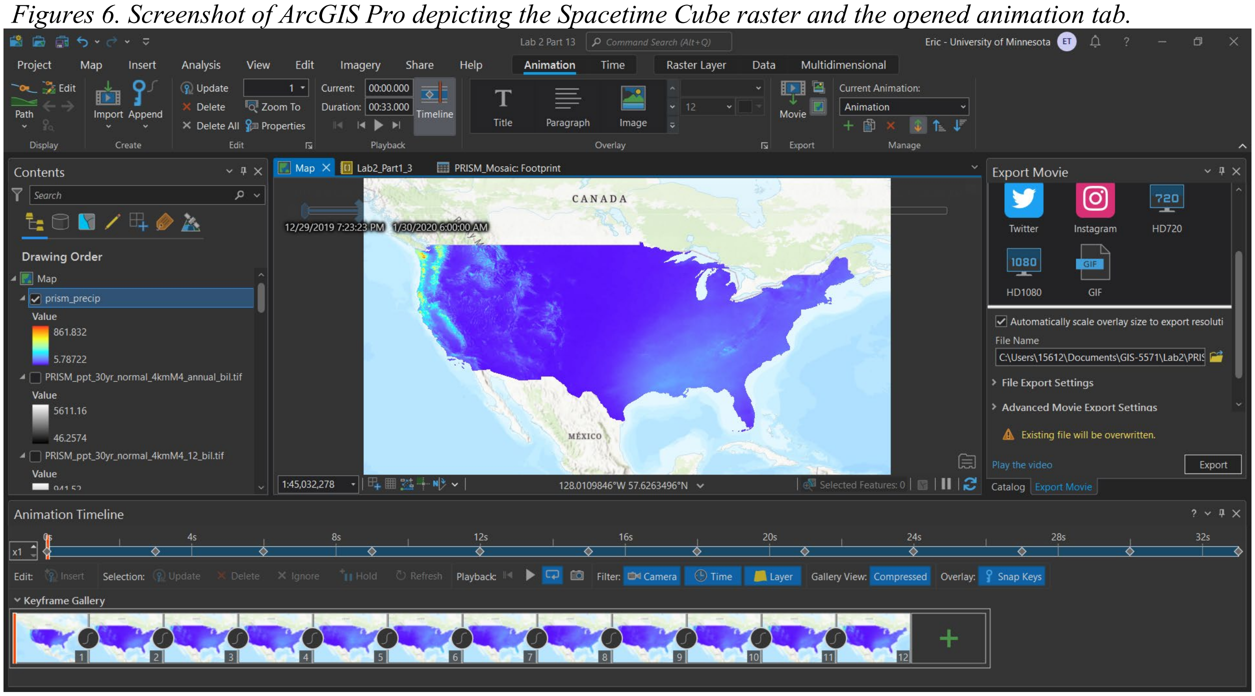
Figures 6. Screenshot of ArcGIS Pro depicting the ArcScene layer created from LiDAR data gathered in Section 1.



Section 3

Finally, the most difficult section within the first part of Lab 2 was Section 3 by far. There were many errors throughout creating the code and crashing was a huge issue when trying to run the “Make Multidimensional Raster Layer” tool through Arcpy. Once precipitation data was gathered from PRISM through an API call using the “requests.get()” function, the raster were changed from .bil files to .tif files simply because the geographic information can be embedded within a .tif file and they provide more GIS functionality overall. Once this was finished, a geodatabase was created and a mosaic dataset was added to the geodatabase using the “Create File Geodatabase” and “Create Mosaic Dataset” tools respectively, again via Arcpy. All .tif files were put into a list and added to the mosaic dataset through the “Add Rasters to Mosaic Dataset” tool. As mentioned in the **Methods** section, the mosaic dataset was cleaned up and prepared to make it multidimensional using the “Build Multidimensional Info” and “Make Multidimensional Raster Layer” tool. Running the latter tool in Arcpy caused massive crashing issues, so there was a delay before moving on to create the Spacetime Cube. Fortunately, this issue was fixed fairly quickly and an animation was made using the Spacetime Cube formed from the “Create Space Time Cube Multidimensional Raster Layer” tool. Once again, the script for this entire process as well as the created GIF can be viewed in the GitHub link above and a screenshot of the Spacetime Cube raster with the animation tab open in ArcGIS Pro is shown below:

Figures 6. Screenshot of ArcGIS Pro depicting the Spacetime Cube raster and the opened animation tab.



Results Verification

After completing the steps necessary for outputting two PDFs (one depicting a DEM and one depicting a TIN), a 3D scene layer, and an animation of monthly precipitation averages, it seems the results are correct, although the code used may not be the most organized as shown in GitHub. Initially, Section 1 took a little time to find the point cloud data captured via LiDAR, but the process through the ArcGIS Notebook was relatively simple using Arcpy, especially when compared to Section 3. Ultimately, after using a series of for loops and if statements to turn on and off the correct layers in the "Contents" pane, the "Export to PDF" tool created separate PDFs of both the DEM and TIN layers created using the same LiDAR data, so the goal of this section was achieved via the created code.

Section 2, as mentioned above was essentially just couple extra steps for converting the point cloud data to a scene layer. After opening an ArcGIS Scene layout in ArcGIS Pro, it was easy to visualize the 3D nature of the scene layer once the angle of the camera was no longer perpendicular to the ground. This allowed for a unique depiction of topography captured by LiDAR. The comparison between the 2D and 3D visualizations will be made in the **Discussion and Conclusion** section of this lab report.

Section 3, while the most difficult, was also the most rewarding and provided an interesting visual of precipitation data. The Spacetime Cube raster was eventually created using a multidimensional mosaic dataset and converted to an animated GIF which shifts through monthly precipitation averages taken from data recorded between 1991 and 2020. While far from perfect, the GIF cycles through all months from January to December and does change based on the precipitation averages recorded within each region (shown as pixels with a 4-kilometer spatial resolution). Unlike nearly everything else within this lab, the animation could not be created based using an ArcGIS Pro Notebook and was instead created through the "Animation" tool in ArcGIS Pro. This is because the only way to export an animation in ArcGIS Pro is through using this tool, not through coding. The Spacetime Cube as well as the timeline along with it was still made via Python coding in an ArcGIS Pro Notebook, however.

Discussion and Conclusion

Throughout the process of the first part of this lab, I learned many different procedures for making different visualizations of multidimensional data. First, I created a raster depicting 2-dimensional data, then formed a Scene layer depicting 3-dimensional data, and finally I created a spacetime cube depicting 4-dimensional data to make an animation based on time. When comparing a 2D visualization to 3D, they are relatively similar in that they essentially portray the same data and provide a clear illustration of depth to that third dimension (in this case, elevation). However, where they differ is in how they can describe the data to the viewer. Data visualized in a 2D form provides more simplicity and, therefore, is likely better for presenting to a broader audience unfamiliar with GIS. The symbology can provide a general understanding of the difference in elevation and topography within an area. 3D depictions can provide a much greater understanding and a more advanced portrayal of the topography, however. For example, a difference of one hundred meters can only be inferred when examining a 2D map, but the difference can easily be seen when viewing a 3D map at an angle. There are still advantages to the 2D visualization of a .las dataset including it is much less computationally intensive and there are more functions which can be performed on the dataset. An ArcScene layer is completely separate from a .las file, so any of the analyses which can be performed on a point cloud layer cannot be performed on an ArcScene layer. Although I am unsure, I believe this would have to do with how computationally intensive 3D scenes are.

This lab provided me with the tools necessary for near complete automation of 3D scene layers and spacetime cubes, which are important for analyses of raster data. There were a few elements which could not be performed using code, mainly including the animation formed from the spacetime cube. Another one of these elements includes the addition of maps and layouts within an ArcGIS Pro project. While it is fairly easy to add these, I initially believed maps and layouts could be added through Arcpy or at least via code within the notebook, but it was found this was not the case. Moments like these within my coding analysis created a few walls for me to break through along with discovering steps needed to move from one set of code to the next. Still, this is to be expected and, in the end, I learned many skills and am proud to move forward and use them later in this class and beyond.

References

Minnesota Geospatial Commons. "LiDAR Samples." *Index of /Pub/Data/Elevation/LIDAR/Examples/Lidar_sample/Las*, 2012, resources.gisdata.mn.gov/pub/data/elevation/lidar/examples/lidar_sample/las/. Accessed 21 Oct. 2023.

PRISM. "PRISM 30-Year Normals." *Index of /Normals_4km/PPT*, Oregon State, 2020, ftp.prism.oregonstate.edu/normals_4km/ppt/. Accessed 21 Oct. 2023.

Multiple prompts. *ChatGPT*, 13 Feb. version, OpenAI, Sept. - Oct. 2023, chat.openai.com.

ESRI. "ArcPy: Documentation: Arcgis Developers." *Documentation | ArcGIS Developers*, 2023, developers.arcgis.com/documentation/arcgis-add-ins-and-automation/arcpy/.

ESRI. "ArcGIS pro Geoprocessing Tool Reference." *ArcGIS Pro Geoprocessing Tool Reference-ArcGIS Pro | Documentation*, 2023, pro.arcgis.com/en/pro-app/latest/tool-reference/main/arcgis-pro-tool-reference.htm.

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	**20**
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	**20**
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	**18**

		100	**86**