**Lab 2 Pt 1 Report**

Title: Final Project Rough Draft

Notice: Dr. Bryan Runck

Author: Jacob Harris

Date: 10/31/2023

Project Repository: [*https://github.com/harr2887/GIS5571*](https://github.com/harr2887/GIS5571)

**Time Spent:** 6 hrs

**Abstract**

For this part of Lab 2, I perform a variety of computation tasks to demonstrate my comfortability working with LiDAR data. More specifically, the data types that I’ll be working with raster, cube, TIN, and and Digital Elevation Model (DEM) geodatabase. Then, I perform side-by-side exploratory spatial data analysis using 2D and Scene views. Lastly, I use ArcPy to export to a PDF a visualization of LiDAR data.

**Problem Statement**

To demonstrate proficiency in working with raster data, I first must extract, transfer, and load LiDAR data from the Minnesota DNR’s FTP server and convert those.LAS files to DEM and TIN data. Next, there are situations when working with elevation data where it is necessary to analyze data in 2D map view and other times when it is necessary to explore data within Scene views. In this lab, I perform both analyses. Lastly, when analyzing LiDAR data, it is important to demonstrate how to export PDFs to document work.

### Table 1. Input Data

| **#** | **Requirement** | **Defined As** | **(Spatial) Data** | **Attribute Data** | **Dataset** |
| --- | --- | --- | --- | --- | --- |
| 1 | DEM | MN LiDAR data by county | lidar | Elevation | MN DNR FTP Server |
| 2 | Weather | Monthly, 30 year normal 1991-2020 | .BIL | precipitation | Oregon State University PRISM FTP Server |

### Table 2. Input Data

| **#** | **Title** | **Purpose in Analysis** | **Link to Source** |
| --- | --- | --- | --- |
| 1 | PRISM Normals Precipitation Data | Construct Spacetime Cube | <https://resources.gisdata.mn.gov/pub/data/elevation/lidar/> |
| 2 | MN LidarData | Convert lidar to TIN and DEM ouput | [Climate data from Oregon State University](https://gisdata.mn.gov/dataset/elev-30m-digital-elevation-model) |

**Methods**

As seen in the flow charts below, part 1 of this lab followed a pretty standard work flow in the ETL of gathering the data and using ArcPy functions to perform the requested tasks. When converting .LAS to DEM and TIN, I just used the ArcPy LAS dataset functions to set up the dataset and then manually created layouts of the DEM and TIN data to export as PDF. When using the PRISM data, I used ArcPy functions such as Build Multidimensional Info and Make Multidimensional Raster Layer to set up the raster layer with the mosaic dataset before creating the space time cube. After exporting the cube as a .nc file, I manually screenshot each frame of the layer in ArcGIS Pro before using the Pillow package in Python to append each PNG together to create the GIF below in *Figure 4*.

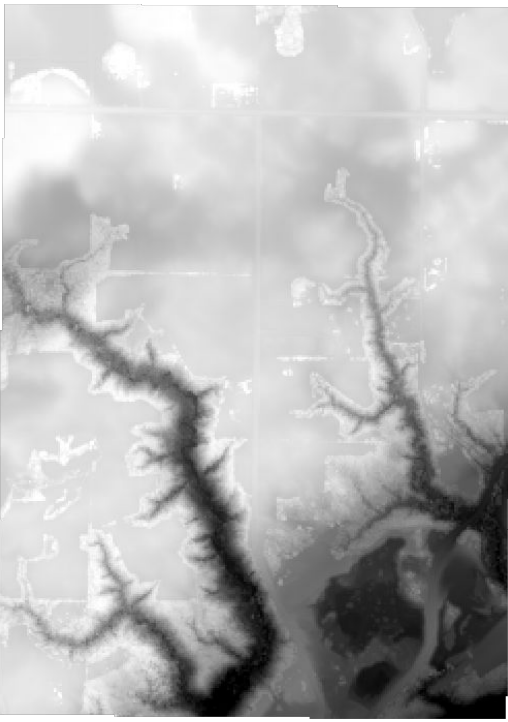
### Figure 1. Lab 2 Part 1 Flow Chart

****

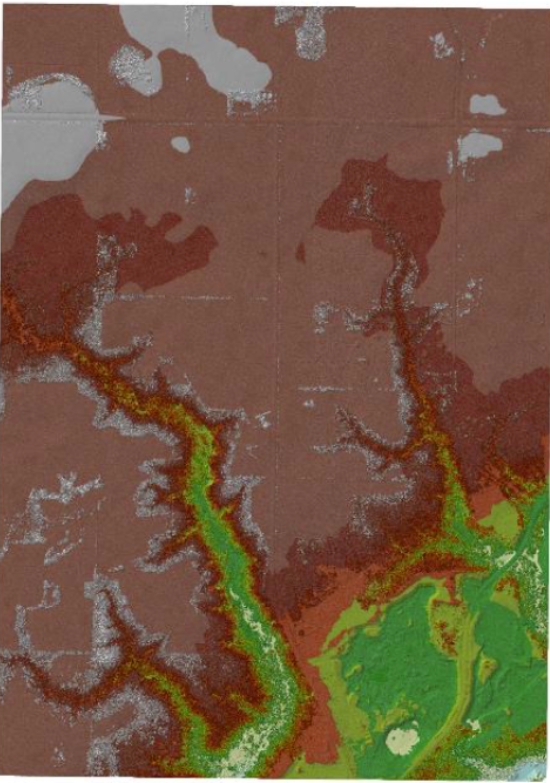
**Results**

In this part of the lab, I was able to generate the three necessary deliverables: a PDF of the DEM and a PDF of the TIN as well as a GIF of the Spacetime Cube. Unlike part 2, there wasn’t any analysis to report on for 1.1 and 1.3. In the discussion section, I’ll talk about my exploration of 2D versus Scene view.

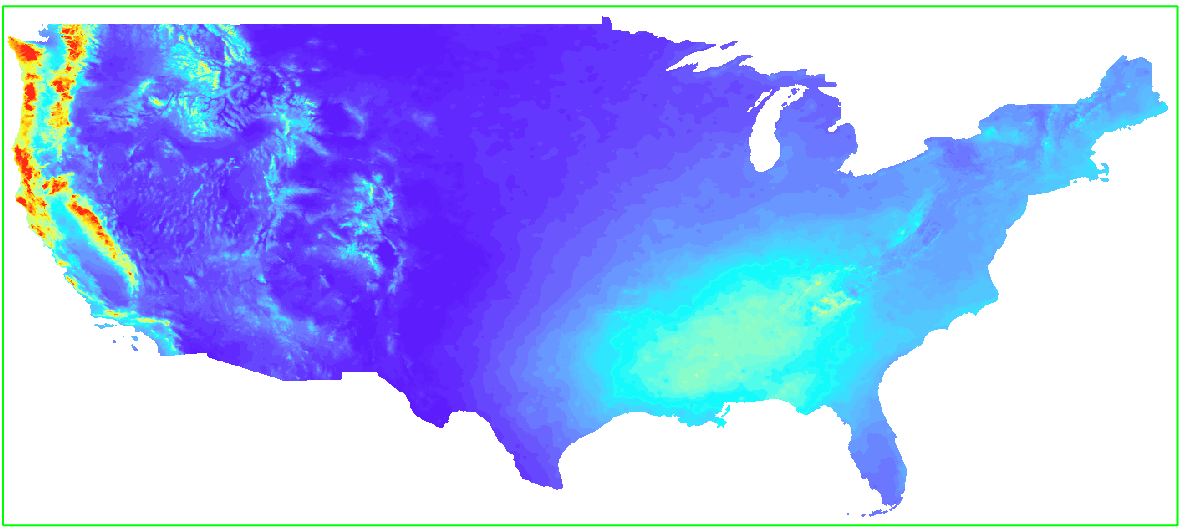
### Figure 2. DEM PDF



### Figure 3. TIN PDF



### Figure 4. Spacetime Cube Precipitation GIF



**Results Validation**

I’m not quite sure how to validate my results for this part of the lab since 1.1’s objective of exporting PDFs is fairly straightforward on whether or not it’s successful. Likewise, 1.2’s exploration didn’t require any results. Lastly, the only validation I did for my Spacetime Cube was the visual check that my .gif looked accurate and it seemed to follow the trend shown in the Spacetime Cube data.

**Discussion and Conclusion**

This lab was definitely more lengthy than the other labs since it required four distinct tasks; However, I think that part 1 went fairly well. I enjoyed learning how to create a spacetime cube since my program, Applied Economics, works with Time Series data a lot so it will probably be useful for me at some point in the future. Unfortunately, I wasn’t able to generate the .gif completely in Python since I couldn’t figure out how to get the correct zoom when exporting a .png of each frame. Instead, I manually took screenshots but used Python’s Pillow package to append them together into a .gif.

When look at the lidar data in both as a 2D map and 3D with Local Scenes, I couldn’t help but think about the situations when either would be best. In the 3D view, it was great to rotate the view/zoom of the data and visualize the details with tools in the raster analysis tool kit in ArcGIS Pro. However, I realized that if the analysis that you need to run can be done with just looking at the data in ArcPy, I wouldn’t bother navigating within Local Scene unless you are confused about some part of the dataset. Conversely, I see the benefit of viewing the data in 3D if you want to export a detailed looking map of the elevation.

**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 | 26 |
| **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 | 18 |
|  |  | 100 | 96 |