Lab Report

Title: Lab 2.1

Notice: Dr. Bryan Runck

Author: Mattie Gisselbeck

Date: November 2, 2022

Repository: https://github.com/mgisselbeck/GIS5571

Time Spent: 20 hours

# Abstract

The main objectives of this lab are to build an ETL that downloads data from the Minnesota DNR, complete a side-by-side exploratory data analysis with a 2D map, and build a ETL that downloads precipitation data from PRISM. The required and input data were sourced and extracted from Minnesota Geospatial Commons and PRISM in ArcGIS Pro via a Python notebook. The methods used to complete this lab are listed below under 'Methods'. Since the lab didn't prove to be analytically intensive, the results don't need to be verified qualitatively. The visual outputs (i.e., maps) could be verified by cross-referencing with peers or with Dr. Runck. The results are shown in the figures below (see Fig. 3 through Figure 8). In this lab, I was able to learn a ton of new information and was also able to build off pre-existing knowledge such as 3D data visualization, creating a space time cube, and building an ETL.

# Problem Statement

The objectives of this lab are divided into three specifics:

(1) Build an ETL that (a) downloads .las files from the Minnesota DNR, (b) converts .las files into a DEM and a TIN, (c) saves the DEM and TIN to disk, and (d) exports PDFs of the DEM and TIN with correct visualization.

(2) Complete a side-by-side exploratory data analysis with a 2D map of the .las file on one pane and a 3D Scene of the .las file on another pane.

(3) Build an ETL that (a) downloads the annual 30-Year Normals .bil files from PRISM, (b) converts .bil files into a space time cube and exports it to a disk, and (c) export an animation of the timeseries.

| # | Requirement | Defined As | (Spatial) Data | Attribute Data | Dataset | Preparation |
|---|---|---|---|---|---|---|
| 1 | LiDAR (.las) | LiDAR for Study Extent | .las | Elevation | Minnesota DNR | ETL |
| 2 | Annual 30-Year Normals | Precipitation Normals (2021) | .bil | Precipitation | PRISM | ETL |
| 3 | NCLD Land Cover | Land Cover Classification | TIFF | Land Cover | Minnesota Geospatial Commons | ETL |
| 4 | Digital Elevation Model | Elevation (Wabasha, Winona, and Olmsted County) | TIFF | Elevation | Minnesota Geospatial Commons | ETL |

*Table 1. Required Data*

# Input Data

The table below is a collection of data from Minnesota Geospatial Commons and PRISM. Data was extracted through an ETL pipeline in ArcGIS Pro via a Python notebook. Data scraped from PRISM will be used to create a space-time cube from a multi-dimensional raster layer and visualize the annual precipitation normal as a .gif animation. Data from Minnesota DNR will be converted into a TIN and DEM, and then visualized on a map layout.

| # | Title | Purpose in Analysis | Link to Source |
|---|---|---|---|
| 1 | LiDAR (.las) | To convert data to a TIN and DEM and visualize the output | Minnesota DNR |
| 2 | Annual 30-Year Normals | To create a space-time cube from a multidimensional raster layer and visualize it as an .gif animation | PRISM |

*Table 2. Input Data*

# Methods

## Part 1.1

To download the LAS files from the Minnesota DNR, I used the ‘requests.get’ function in the requests package. Following that, I implemented the ‘LAS Dataset to Raster’ conversion command from ArcPy to convert the LAS to a DEM. To convert the LAS to a TIN, I used the ‘LAS Dataset to TIN’ conversion command. To export the TIN and DEM with correct visualization, I used the Python sub module, Arcpy.mp. Then I applied the ‘ListLayouts()’ in the project workspace and ‘exportToPDF’ functions which gave a path for the PDF outputs (This module was repeated for both DEM and TIN files). The results are shown below (see Fig. 3 and Fig. 4).

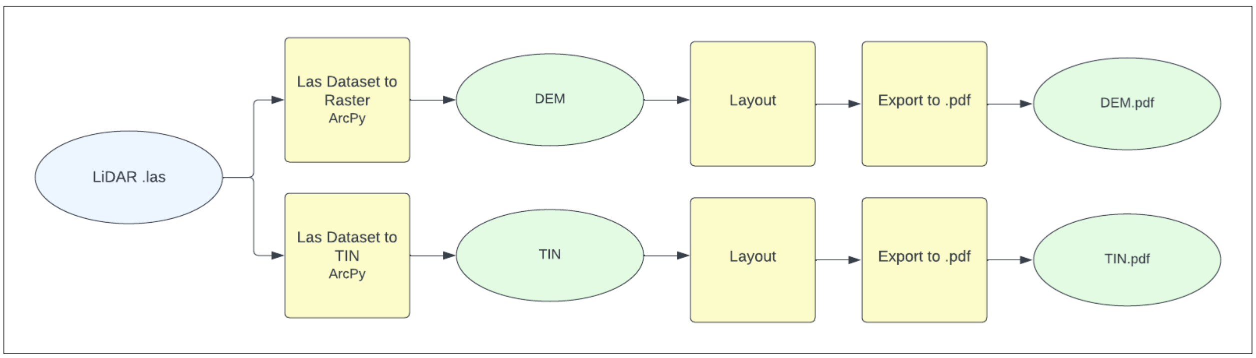


*Figure 1. Data Flow Diagram - Part 1.1. (Note:* .ipynb for Part 1.3).

## Part 1.2

For the 3D and 2D side-by-side exploratory data analysis, I created a new map and local scene in a parallel pane (see Fig. 5 and Fig. 6). In comparison to the 2D pane, the 3D pane provides more options to visualize an las dataset such as symbology. The symbology tab in the 3D pane, lists four options a user can choose from: (1) ‘symbolize your layer using point’, (2) ‘symbolize your layer using contour’, (3) ‘symbolize your layer using edges’, and (4) ‘symbolize your layer using a surface’. The ‘symbolize your layer using point’ options allow you to draw using a variety of different variables such as elevation, intensity, and scan angle. ‘Symbolize your layer using contour’ allows you to draw contours, manipulate symbology of the contours, and edit the type, index factor, and height. The ‘symbolize your layer using edges’ allows you to change the type of edge drawing and symbology. ‘Symbolize your layer using a surface’ allows a user to edit the drawing variable, method, classes, and color scheme. The explore mouse in a 3D scene allow a user to change the view angle, zoom in and out, and scroll up and down. Visualizations of the side-by-side comparisons are shown below in Fig. 5 through Fig. 7.

## Part 1.3

The objective of this specific was to build an ETL that downloads the annual 30-Year Normals .bil files from PRISM, converts the .bil files into a space time cube and exports it to a disk, and exports an animation of the timeseries. I built an ETL that automates the downloading from PRISM and unzips the file. The following steps were applied to create a Space Time Cube:

### Part 1.3.1

**1.3.1: Downloads the Annual 30-Year Normal .bil files from PRISM**

In [7]:

```
# Import Packages
import arcpy
import os
import requests
from zipfile import ZipFile
```

In [9]:

```
# Setting File Paths for Outputs
zip_path = os.path.join(os.getcwd(), "data/PRISM")
file_name = os.path.join(zip_path, "PRISM_ppt_30yr_normal_4kmM3_all_bil.zip")

# Check if Path Exists
if os.path.exists(zip_path) == False:
    os.mkdir(zip_path)

prism_url = "https://ftp.prism.oregonstate.edu/normals_4km/ppt/PRISM_ppt_30yr_normal_4kmM3_all_bil.zip"

resp = requests.get(prism_url)

if os.path.exists(file_name) == False:
    with open(file_name, "wb") as z:
        z.write(resp.content)
else:
    print("ZIP file already exists.")

# Unzipping File
if os.path.exists(file_name[:-3]) == False:
    with ZipFile(file_name, "r") as zipped:
        zipped.extractall(file_name[:-3])
else:
    print("File has already been unzipped.")
```

Part 1.3.2

Created a Mosaic Dataset > “PRISM” and Added Rasters to Mosaic Dataset (Input: All .bil files)

1.3.2: Converts the Annual 30-Year .bil data into a Space Time Cube and Exports it to a Disk

In [12]:

```
# Create Mosaic Dataset
sr = arcpy.SpatialReference(3857)
empty_mosaic = arcpy.management.CreateMosaicDataset(r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2
```

In [13]:

```
# Add Rasters to Mosaic Dataset
arcpy.management.AddRastersToMosaicDataset("PRISM_ALL_1_12", "Raster Dataset", r"C:\Users\gisse015\Docur
```

Part 1.3.3

Calculate Field (Add Field: Variable – PPT) and Calculate Field (Add Field: Timestamp – PPT)

In [17]:

```
# Calculate Field for Variable and Timestamp
arcpy.management.CalculateField(r"PRISM_ALL_1_12\Footprint", "Variable", '"PPT"', "PYTHON3", '', "TEXT",
arcpy.management.CalculateField(r"PRISM_ALL_1_12", "Timestamp", 'DateAdd(Date(2021, 0, 1), $feature.OBJI
```

Part 1.3.4

Build Multidimensional Info

In [25]:

```
# Build Multidimensional Info
arcpy.md.BuildMultidimensionalInfo("PRISM_ALL_1_12", "Variable", "Timestamp # #", "PPT # #", "NO_DELETE_
```

Part 1.3.5

Make Multidimensional Raster Layer > PRISM_MultidimensionalRaster

In [27]:

```
# Make Multidimensional Raster Later
arcpy.md.MakeMultidimensionalRasterLayer("PRISM_ALL_1_12", "PRISM_ALL_1_12_MultidimLayer1", "PPT", "ALL'
```

Part 1.3.6

Create Space Time Cube from Multidimensional Raster Layer > “C:\Lab2_2\PRISM_STC.nc” (See Figure 8).

In [28]:

```
# Create Space Time Cube from Multidimensional Raster Layer
arcpy.stpm.CreateSpaceTimeCubeMDRasterLayer("PRISM_ALL_1_12_MultidimLayer1", r"C:\Users\gisse015\Documer
```

Part 1.3.7

To create an animation, I created key frames for the corresponding 12 months under the multidimensional tab. Then under the animation tab, I exported the animation as a .gif. (See Figure 9).

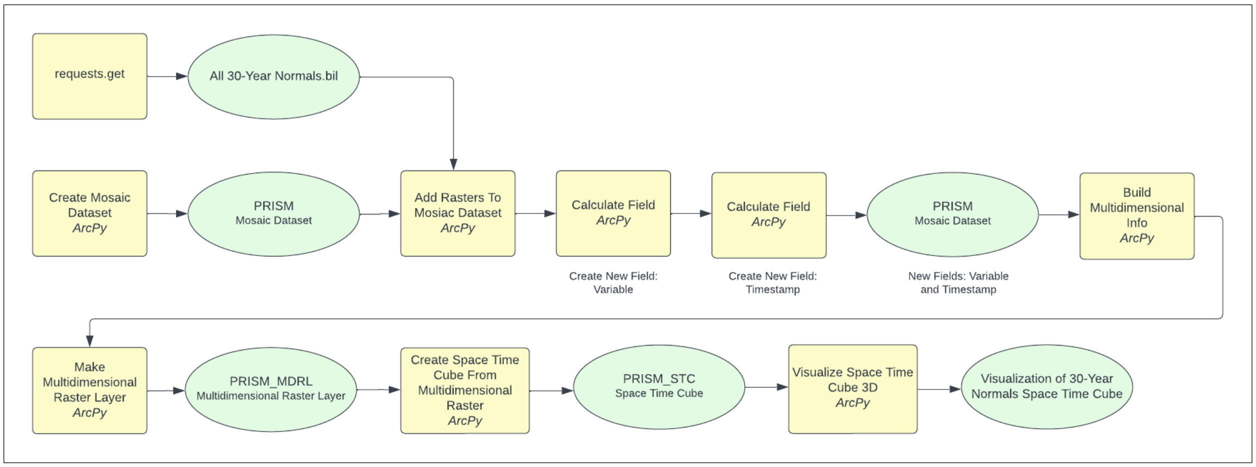


*Figure 2. Data Flow Diagram - Part 1.3.*

# Results

The results are shown in the figures below (see Fig. 3 through Figure 8). The main themes of the lab were data conversion, visualization of raster datasets, creating a space time cube, comparing 3D to 2D visualization, and visualizing multidimensional raster layers. Two maps were created to visualize the TIN and DEM output in correct cartographic form.

# Results Verification

This lab didn’t prove to be analytically intensive but did emphasize on data visualization and exploration. Therefore, the results don’t need to be verified qualitatively. The visual outputs (i.e., maps) could be verified by cross-referencing with peers or with Dr. Runck.

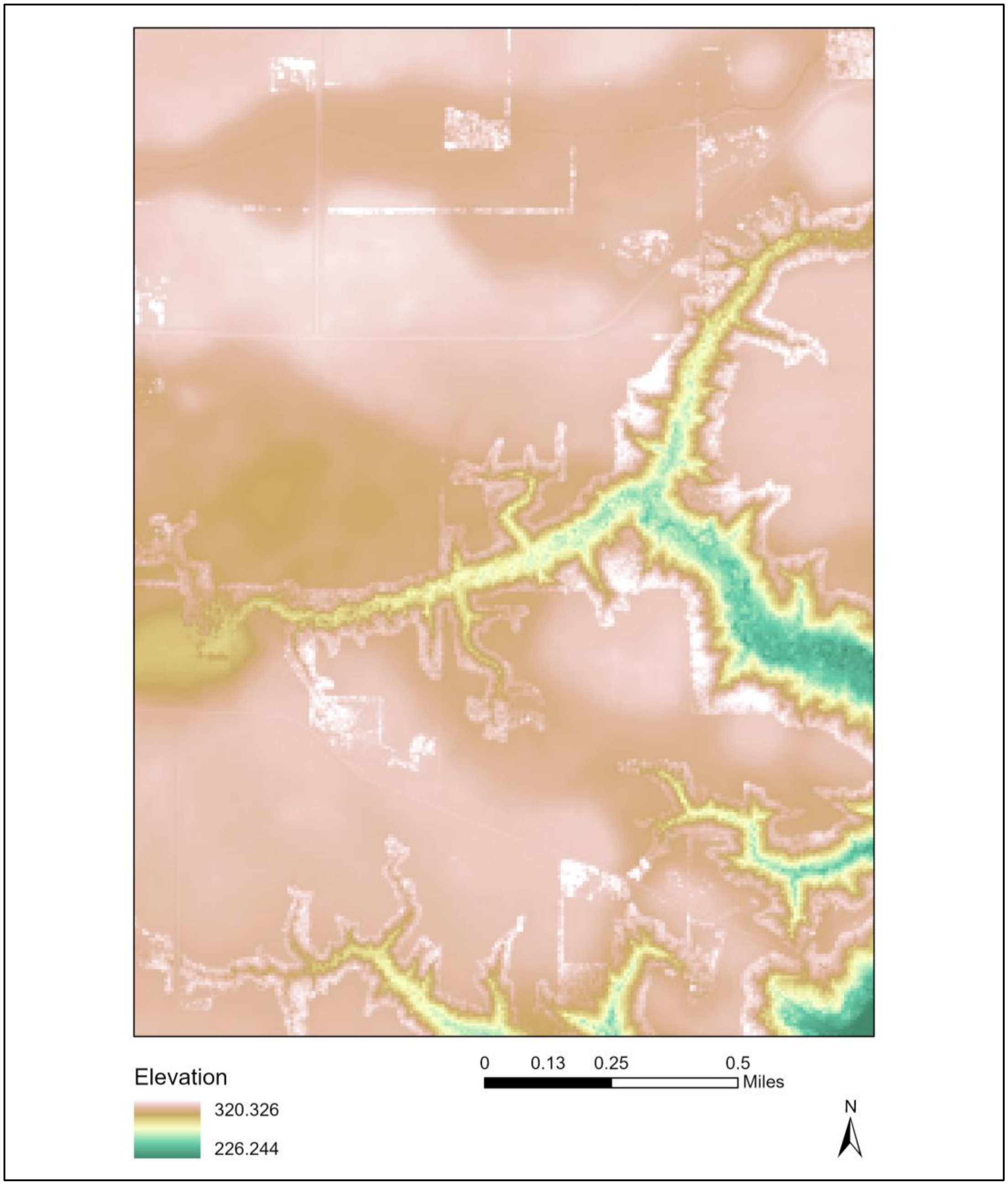


*Figure 3. LAS to DEM Output Map*

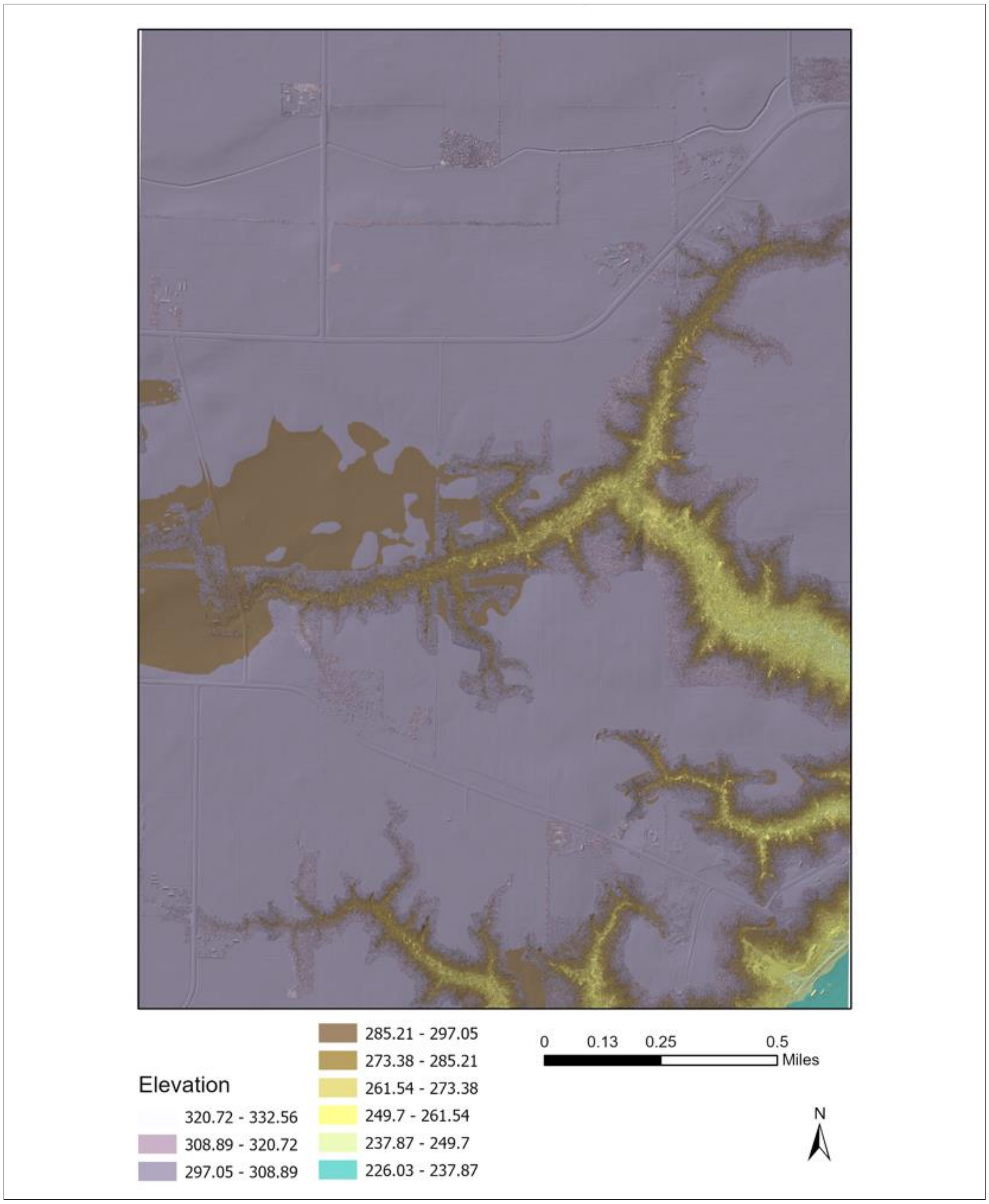


*Figure 4. LAS to TIN Output Map*

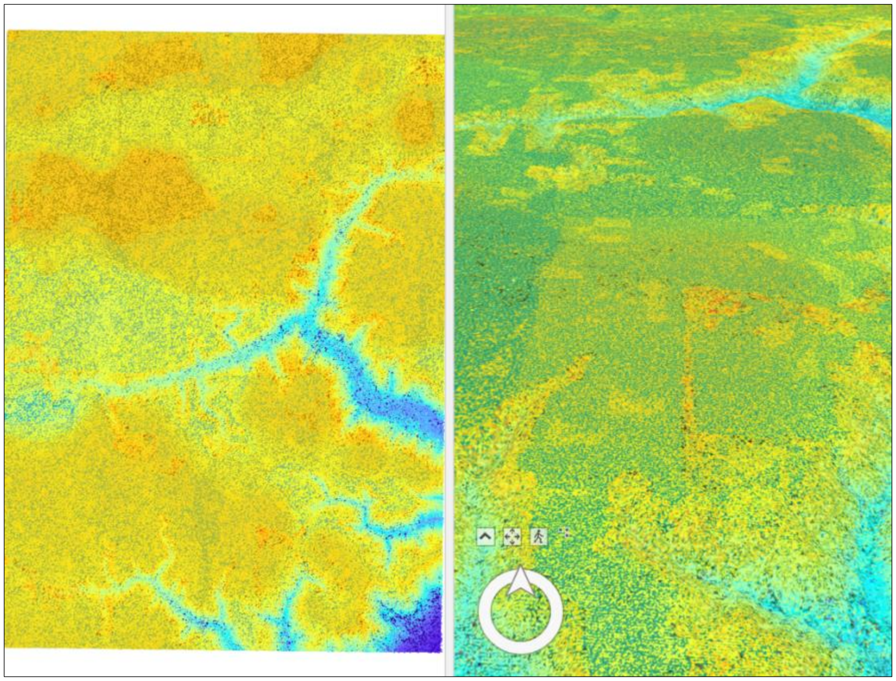

*Figure 5. 3D and 2D Data Exploratory.*

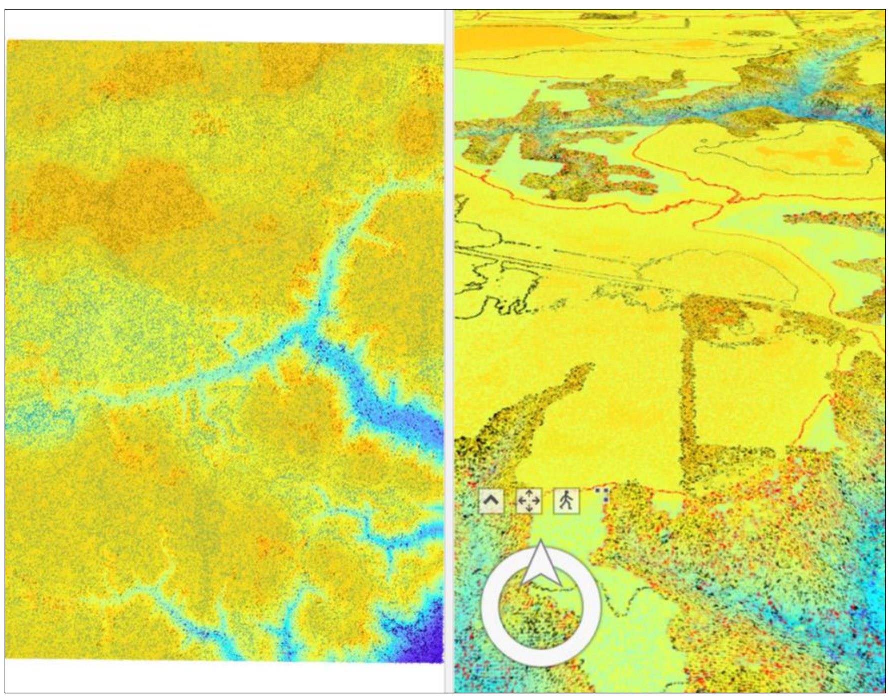

Figure 6. *3D and 2D Data Exploratory - Contours.*

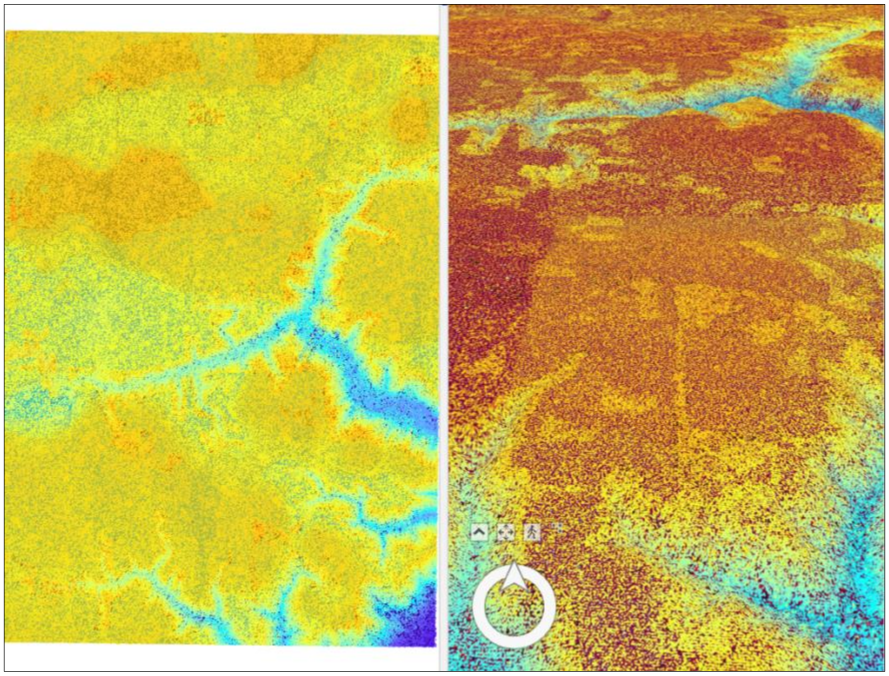

Figure 7. *3D and 2D Data Exploratory - Edges.*

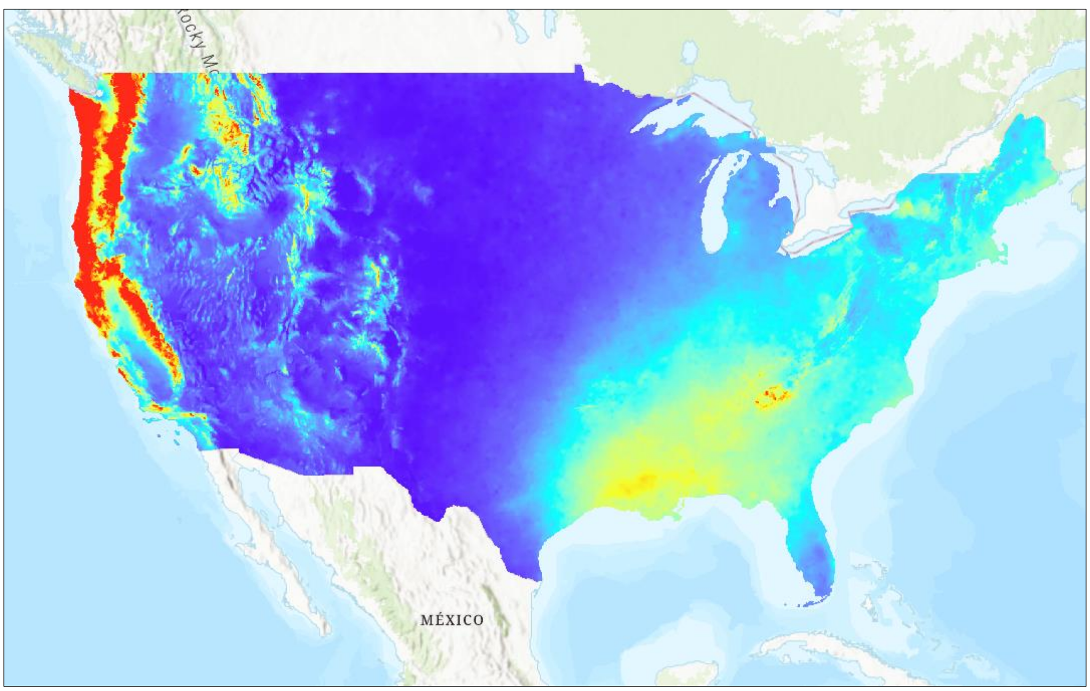


*Figure 8. PRISM Annual Precipitation Multidimensional Raster Layer*

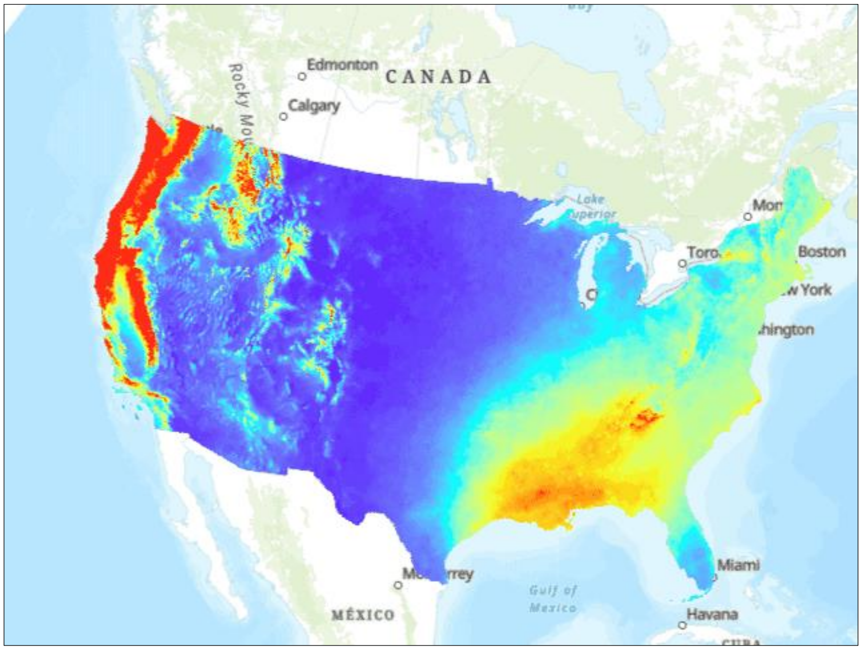


*Figure 9. PRISM Annual Precipitation Multidimensional Raster Layer (.gif)*

# Discussion and Conclusion

In this lab, I was able to learn a ton of new information and was also able to build off pre-existing knowledge. I learned how to create a space time cube, build multidimensional info, make a multidimensional raster, create and visualize a timeseries, and export a timeseries animation as a .gif. I gained supplementary experience working with LiDAR LAS datasets in a 3D scene. Despite Ersi's documentation on 'LAS dataset to DEM' command, I found it difficult to get the code to run without error. The space time cube also proved to be time intensive. I found the 3D and 2D side-by-side analysis useful since I haven't opened a LAS dataset in a scene.

# References

Ersi. 2022. Create Space Time Cube From Multidimensional Raster Layer (Space Time Pattern Mining). https://pro.arcgis.com/en/pro-app/latest/tool-reference/space-time-pattern-mining/createcubefrommdrasterlayer.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 | **23** |
| **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 | **27** |
| **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 | **19** |
| | | 100 | **97** |