

Lab Report

Title: Lab 2

Notice: Dr. Bryan Runck

Author: Cole Anderson

Date: 3/4/21

Project Repository: <https://github.com/and04671/GIS5572/tree/main/Lab2>

Abstract.

The purpose of this lab was to build ETLs to retrieve raster data from various public servers, and transform it into various visualization types, including TIN, LAS, and Space-Time Cube. The input data is a sample raster from the DNR (to transform into TIN/DEM) and a set of 30 year normals files from PRISM Climate Group. Requests and the arcpy.conversion toolbox were used to turn LAS files into TIN/DEM files and export them. Ftplib, arcpy.management, arcpy.md, and arcpy.stpm were used to gather BIL files and transform them into a space time cube via a mosaic dataset. Results in Figure 2-5 show a successful TIN and DEM version of an LAS file, how LAS files themselves can be symbolized in ArcGIS Pro, and a time slice of the final cube model. These results are determined to be the desired ones. In conclusion, LAS and BIL files were utilized to illustrate how to use raster or variable data in the context of different outputs or modification tools.

Problem Statement

Use skills in API queries with raster, TIN, and Terrain data transformation steps to extract, convert, and export LAS files for LiDAR data from the Minnesota DNR's FTP server. Convert LAS data to TIN/DEM, and export TIN and DEM datasets from the LAS data. Explore 2D and 3D visualizations of these DNR LAS datasets in ArcGIS Pro and describe the tools and features of each. Finally, use skills with cubes and APIs to retrieve BIL files from PRISM and use them to create a space-time cube visualization of a 30 year normals variable.

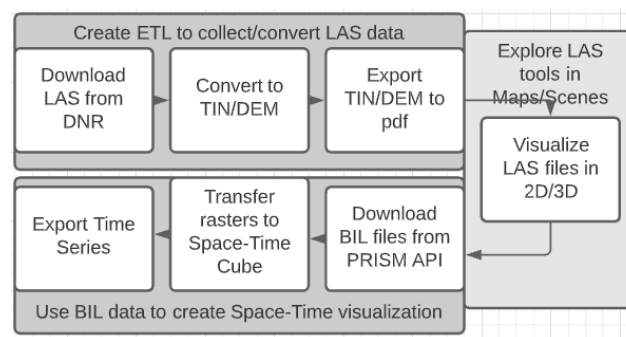


Table 1. Problem Statement Components

#	Requirement	Defined As	Spatial Data	Attribute Data	Dataset	Preparation
1	Download LAS file from DNR	Raw API pull from DNR FTP server	LAS file		LAS File net location	

2	Convert LAS to TIN/DEM and save to disk	Arcpy.LasDatasetToTin_3D Arcpy.LasDatasetToRaster	LAS file data		LAS saved file	
3	Export TIN/DEM properly visualized	(layout).exportToPDF		Elevation	TIN/DEM saved file	
4	Data Analysis 2D and 3D visualizations of LAS files	Tool and Visualization options		Elevation	TIN/DEM saved file	
5	Download BIL Files from PRISM	FTP module to retrieve from PRISM server and extract to file		Precipitation	30 Year Annuals	
6	Transform to Space-cube, Export	TIF > Mosaic Set > Variables > MultiDem. Raster > Create Space Time Cube > Visualize Cube			BIL saved files	Convert BIL to TIF, Create Mosaic Dataset
7	Export Time Series Animation	Time Slider?/Export to animation			BIL saved files	

Input Data

The PRISM file shows an approximation of precipitation over the last 30 years for each month of the year, at a number of weather stations across the US. The DNR LAS file is a small sample file of an area in the southwestern corner of the US.

Table 2. Input Data

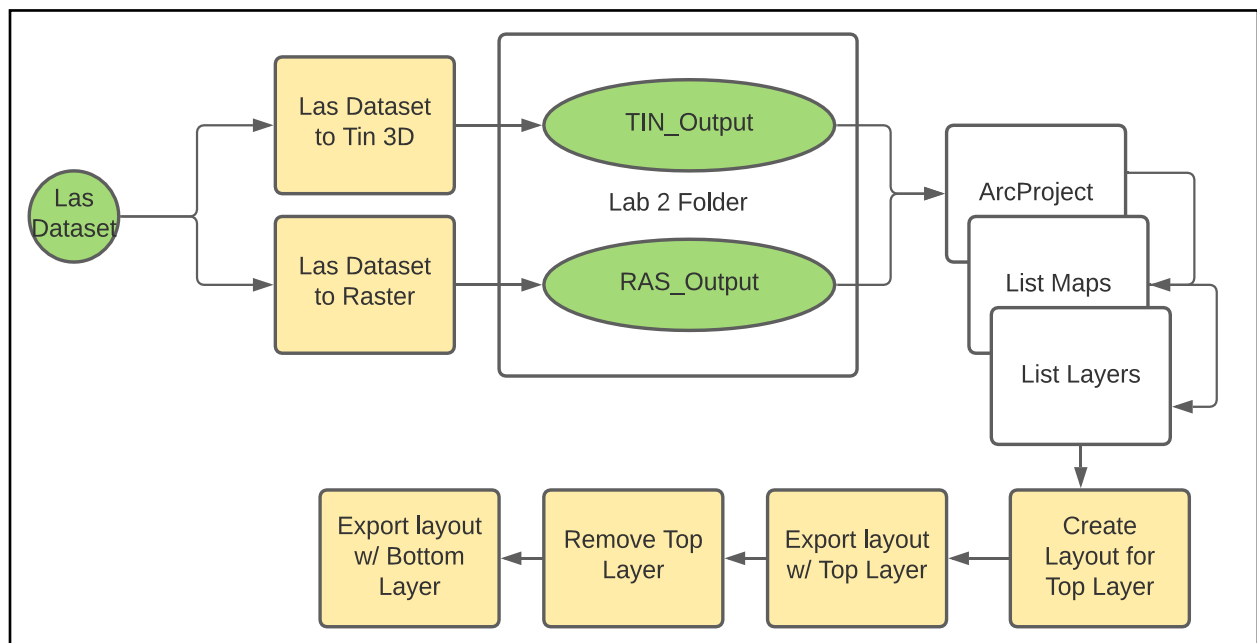
#	Title	Purpose in Analysis	Link to Source
1	DNR LAS example file	Create TIN and DEM visualization	https://resources.gisdata.mn.gov/pub/data/elevation/lidar/examples/lidar_sample/las/4342-12-05.las
2	PRISM 30 Year Normals	Creation of space-time cube	<a href="http://prism.nacse.org/normals_4km/ppt/PRISM_ppt_30yr_normal_4kmM2_<xx>_bil.zip">prism.nacse.org/normals_4km/ppt/PRISM_ppt_30yr_normal_4kmM2_<xx>_bil.zip

Methods

LAS ETL retrieval

- Import requests library to write HTTP requests. Enter the resources link to the DNR FTP, at <https://resources.gisdata.mn.gov/pub/data/elevation/lidar>. Navigate to examples/lidar_sample/las/4342-12-05.las, and copy the URL. Send a GET request to this URL and save to a variable. Open a blank file in write binary mode, writing the contents of the response variable to the file. Close the written file. There will now be an LAS file in the notebook directory, ready for format conversion.

- Import the arcpy library. Convert the LAS file to a TIN file using `arcpy.LasDatasetToTin_3D`. Set the thinning type to 'RANDOM', using thinning method 'PERCENT' and thinning value 30. This will thin the number of points in the TIN dataset randomly, at a rate of 30% of the LAS dataset. Convert the same LAS file to a raster using `arcpy.conversion.LasDatasetToRaster` using the elevation attribute as the variable value. Both functions will output respective files to the notebook directory, in this case TINoutput and RASoutput.
- Add both layers, TINoutput and RASoutput to an .aprx ArcGIS Pro document with the TIN on top. With this file open, establish it as the current ArcGIS Project using `arcpy.mp.ArcGISProject('CURRENT')` and save to a variable `aprx_file`. Search through the aprx file to find the 0th map using method `.listmaps`, and save to a variable `map_in_aprx`.
- Create a blank layout in the aprx file manually, by opening ArcPro and creating a layout. Select the map with the TIN/RAS layers as the map frame and scale appropriately. Going back to arcpy, find this new layout using `aprx_file.listLayouts()`, and save to variable `Layout1`. The new layout should be the 0th.
- Export the existing layout to a PDF using `Layout1.exportToPDF`. This should be the TIN file PDF. Remove the top layer from the layout by modifying the corresponding map through `map_in_aprx.removeLayer(map_in_aprx.listLayers()[0])`. The inner method here finds the top layer in the map. Export the layout again using `Layout1.exportToPDF`. This should be the RAS file PDF.



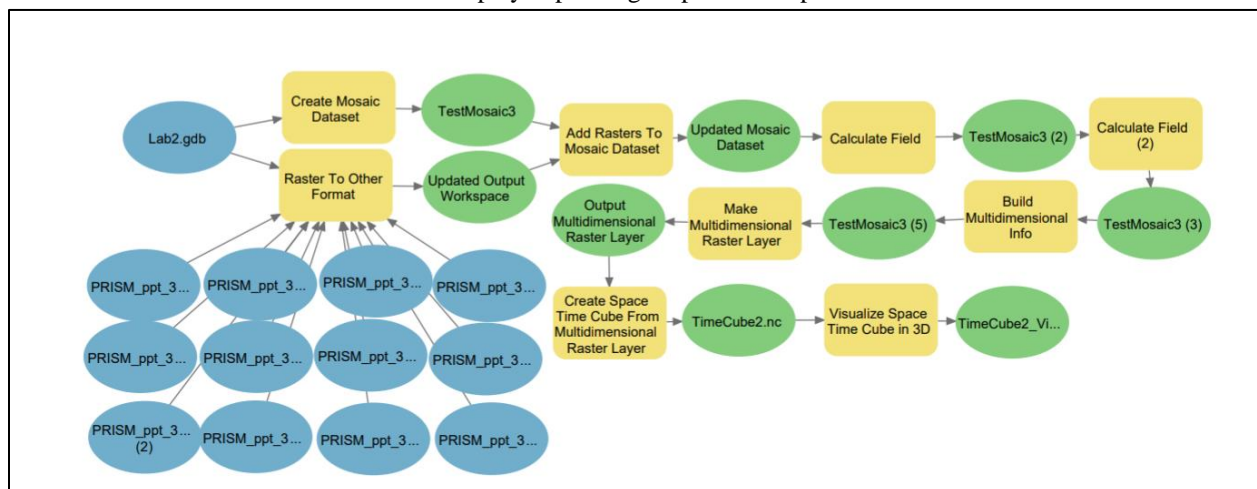
Exploring 2D/3D LAS visualizations

- Most of the tools and symbology options for 2D Map and 3D Scene visualizations will be discussed in the Results section.
- To create a scene visualization, open the project file. Right click on "New Map" in the Insert tab and click on "New Local Scene." Drag the LAS file onto the scene. Click the mouse wheel and drag to see the scene orthogonally.
- To create a map visualization, open the project file. Click "New Map" in the Insert tab. Drag the LAS file onto the map. Select symbology > elevation in the new Appearance Tab in order to see the visualization.

Space-Time Cube ETL

- Import the requests library, FTP from `ftplib`, and ZipFile from `zipfile`.

- Function `FTPNormalsDownload` retrieves files from the PRISM FTP server with parameter `server_filename`. Inside the function, object `ftp` is created from `FTP('prism.nacse.org')` to open a portal to the web address. Use the `.login()` method and enter the credentials given on the PRISM site (user = 'anonymous', passwd = 'and04671@umn.edu') to gain access to the FTP server. Navigate the FTP directory using the `.cwd` method. 'Normals_4km/ppt' to find the precipitation normal files at 4 km resolution. Inside this directory there are 12 normals files. The server file name is only established as 'server_filename' in the function. Open a new file with the same file name in write binary mode. Write the desired server file to the new save file using the method `.retrbinary('RETR ' + (server_filename>, <local_filename>.write)`. Close the FTP and save file using the `.close()` method. Run the function for every desired .zip file. This can be done in a loop since the files follow the same naming convention. For each file, unzip to the PRISM_ZIPS folder using `ZipFile(desired file)` and the `.extractall()` method.
- Create a blank mosaic dataset 'Mosaic3' in the geodatabase with a NAD83 UTM 15N projection using `arcpy.management.CreateMosaicDataset`. Convert .bil rasters in the PRISM_ZIPS to .tif files, simultaneously entering them in the geodatabase using the `arcpy.conversion.RasterToOtherFormat` method. Add these new .tif files to the mosaic with `arcpy.AddRastersToMosaicDataset`.
- Calculate a new text field in the mosaic, Variable. Set equal to Python 3 expression 'precipitation.' Calculate a new date field in the mosaic, Timestamp. Set this equal to an Arcade expression for the respective month in the series of rasters.
- Create multidimensional information from the mosaic, making sure the `variable_field` parameter is set to the field Variable. Set the `dimension_field` to field Timestamp. Use method `arcpy.md.BuildMultidimensionalInfo`. Make a multidimensional raster layer from the now multidimensional mosaic using `arcpy.md.MakeMultidimensionalRasterLayer`. Set parameter 'variables' as "precipitation." This will output a new raster layer.
- Derive a space-time cube can from the raster layer with `arcpy.stpm.CreateSpaceTimeCubeMDRasterLayer`. Set parameter `fill_empty_bins` as 'ZEROS' to turn no data areas to zeros. The output .nc file must be visualized to open in ArcPro. Use `arcpy.stpm.VisualizeSpaceTimeCube3D` to display the time cube on the map. The cube_variable for precipitation is "PRECIPITATION_NONE_ZEROS." The Time Cube will take several minutes to hours to display depending on processor speed.

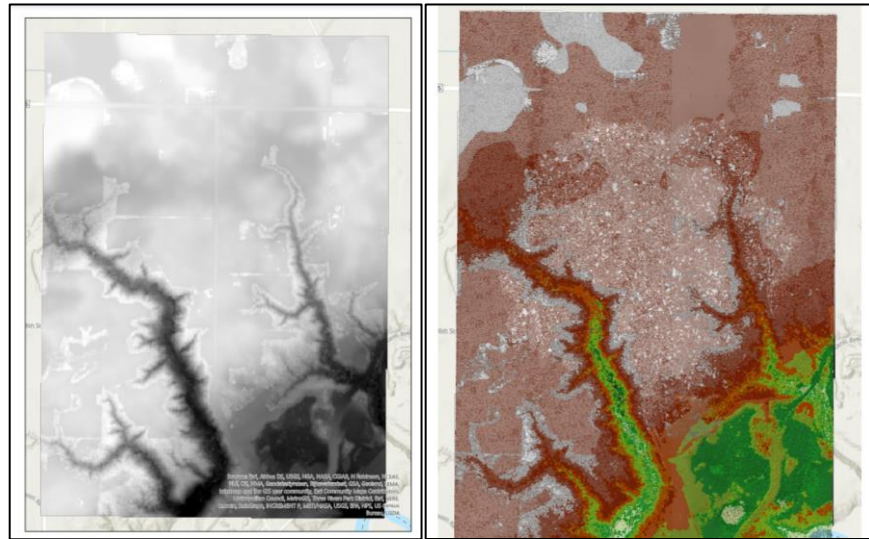


Results

LAS to DEM/TIN outputs

In Figure 3, the DEM output from the LAS file is on the left, and the TIN output on the right. The darker areas in the DEM correspond to the greener areas in the TIN, and represent lower elevation. The TIN picture consists of irregular triangles arranged in a network. The DEM consists of pixels arranged in a raster.

Figure 3: DEM and TIN outputs from LAS data



Exploratory 3D and 2D LAS visualizations

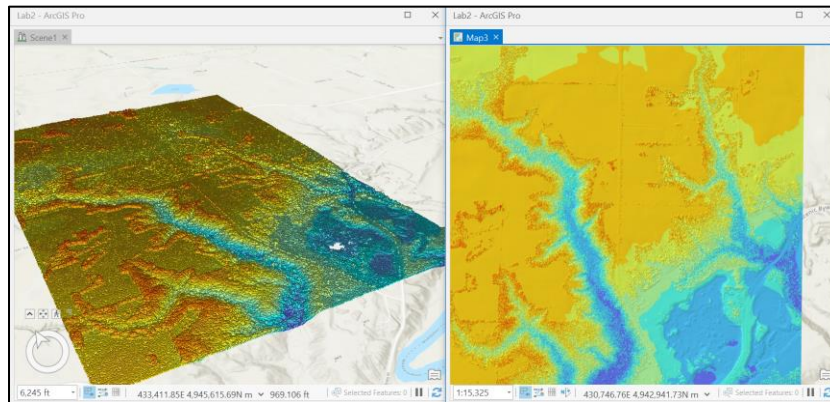
Figure 4 shows the 3D and 2D representations of the downloaded LAS file. Both visualization options show the LAS data in a gradient display of elevation or elevation related attributes. The Scene version has the interesting ability to move around a 3D version of the terrain, making it easier to see in space than a 2D surface, but requiring more processing power. Dragging an LAS file onto the map or scene will show 3 new tabs in the banner about the LAS dataset: 'Appearance', 'Data', and 'Classification'.

Under 'Appearance', the symbology can be adjusted to show different attributes in surfaces, lines, or points. Surface symbology can show elevation, slope, or the aspect value of the terrain. Line symbology can show edges or contour lines. Point symbology can show the elevation, the LAS class, the intensity value, or the lidar pulse return value. The point density can also be adjusted in the appearance tab to create coarser but faster rendering, or detailed slow rendering. Points can also be filtered by various criteria. Scenes have the additional option to adjust the radius and strength of eye-dome lighting.

Under 'Data', there are several function options for both scenes and maps. 'Area and Volume' finds these respective values between two surfaces or inside a bounding box. 'Height Metrics' will output measures of central tendency, skew, and kurtosis for the LAS dataset. 'Locate Outliers' finds points in the dataset outside of the defined elevation range or with inconsistent slopes. 'Surface Derivatives' will find the derivatives of the slope, aspect, and contour. Finally, the 'Visibility' tool will illustrate the line of sight from any point on the surface, or from the horizon.

Under 'Classification', ground points can be re-classified, as well as buildings and locations. Points can also be classified by their proximity to specific features. Class codes and flags can also be reassigned. With Scenes, there is an option to create a profile slice through the surface.

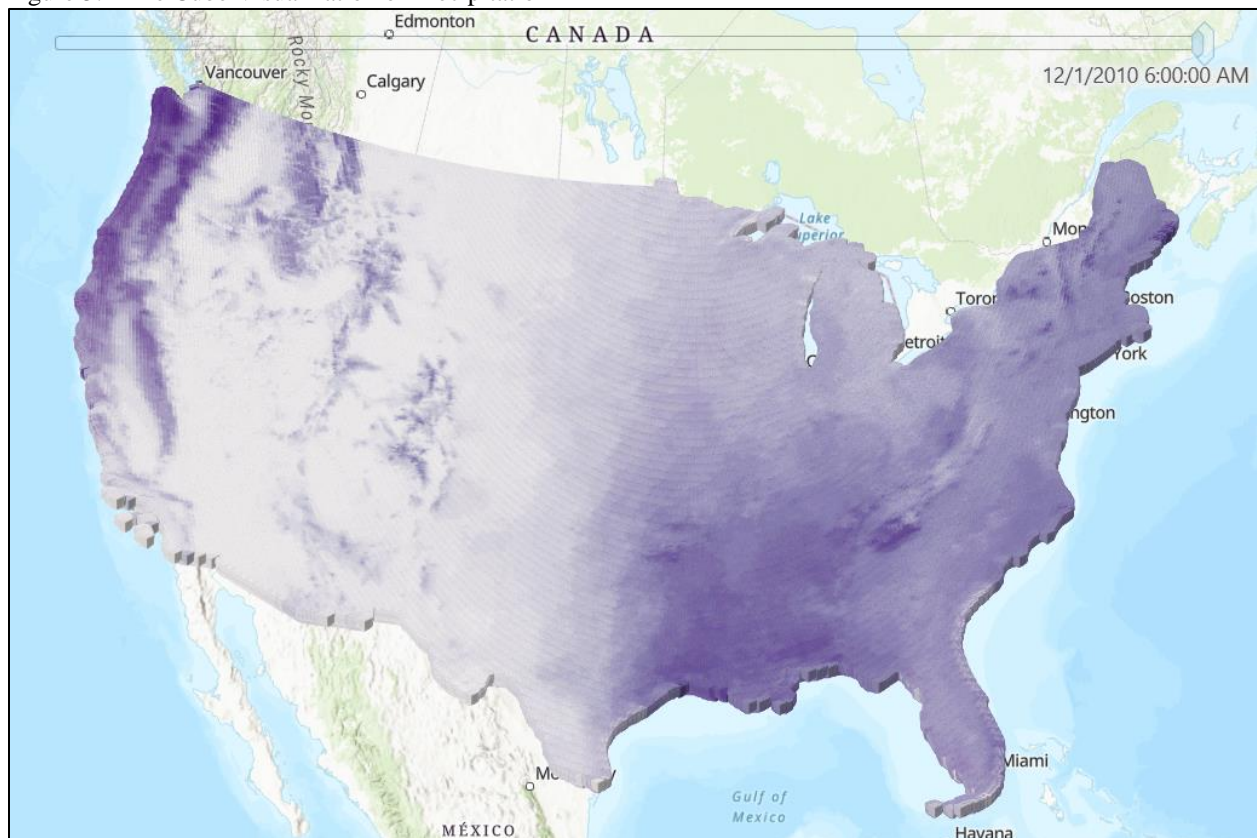
Figure 4: 3D and 2D Visualizations of LAS data



Time Cube Results

Figure 5 shows the time cube visualization of 30-year precipitation normals in the month of December, from 1981 to current. More purple areas have higher precipitation values. The attribute table for this layer shows a time and value for each location ID. See this link for the animation:

Figure 5: Time Cube Visualization of Precipitation



Results Verification

LAS ETL

A good way to test if the code is retrieving the right file is to download the same file manually and compare with the automated download. These files appear identical. To test the LAS to Raster/Tin conversions, run the same tools in the GUI. With the same parameters, the GUI and code produce the same results.

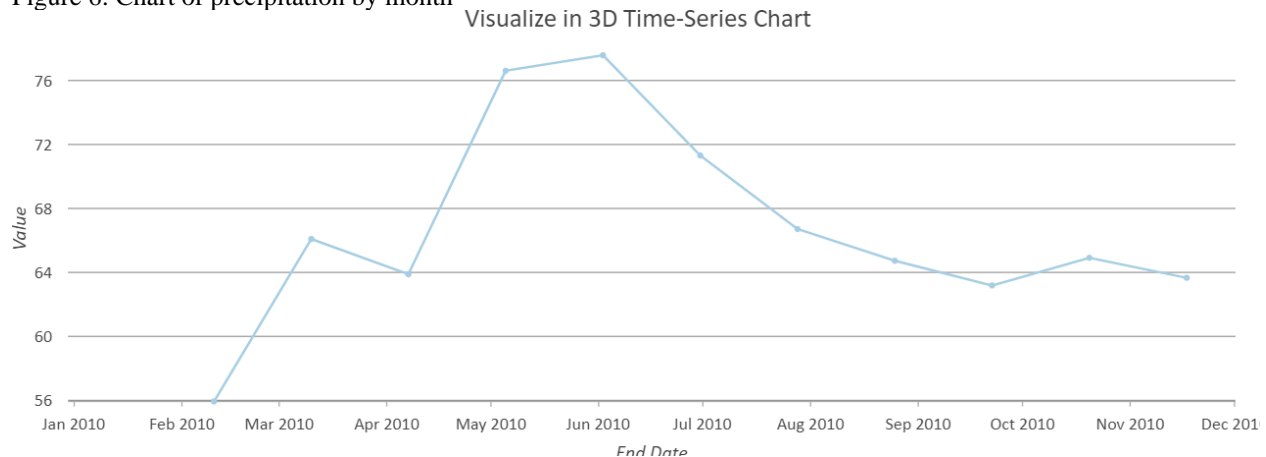
Exploratory 3D/2D LAS

The 2D and 3D visualizations match with the TIN version from the LAS ET, and each other, confirming that they have not been corrupted.

Time Cube

While the image in Figure 5 shows just one time slice, Figure 6 confirms the value does change with month in a sensical manner (higher in the summer than fall/winter). The time cube is too large to easily drag through time with the slider, so this chart will have to be accurate. The animation also confirms changes over time.

Figure 6: Chart of precipitation by month



Discussion and Conclusion

The LAS ETL was straight forward, as the request format was just a basic GET request. There were no CKAN or FTP requirements. The need to manually create a layout was a snagging point; the documentation for arcpy does not make this clear. I learned how to better navigate GIS object maps, layers, and layout in this step. The TIN and DEM outputs were compared and contrasted in class as different ways to display the terrain data contained in another file.

The 3D capabilities of ArcPro are new to me. I have seen the 3D options before but was uncertain how to utilize them for my own projects. Seeing the actual surface in space makes the visualization far easier for less spatially inclined team members. There are several unique tool options for LAS datasets, making LAS visualizations superior to DEM or TIN files in terms of visualization options. LAS datasets can be utilized with conversion, as shown in this test. The ability to see derivatives, slope in particular, is not (as far as I know) available elsewhere in ArcGIS, and is very easy to use.

The space-time tools are simple, linear to use, and easy to automate. The time that the Time Cube takes to create, modify, and appear makes it nearly unusable even with a powerful desktop computer. The advantage is not obvious to me unless there were an enormous number of datasets to visualize. It seems it would make more sense to just make map layers of every month instead, and stack them in an animation. This would however, be tedious if there were more than a dozen datasets. The purpose of this part of the lab is to show a cube can be derived from a set of rasters, as these cubes show attributes through time.

Model Builder was a new and extremely helpful tool for writing python code. By using the GUI tools and being able to convert them to code directly, errors in parameters, inputs, outputs are easily avoided. It provides the text scripting signature to code, but the ease of use of the regular ArcGIS GUI, along with a very nice workflow visualization.

References

- Buie, L. (2020, May 18). Explore your raster data with space Time Pattern Mining. Retrieved March 02, 2021, from <https://www.esri.com/arcgis-blog/products/arcgis-pro/analytics/explore-your-raster-data-with-space-time-pattern-mining/>
- ESRI. (2021). Arcgis pro geoprocessing tool reference. Retrieved March 02, 2021, from <https://pro.arcgis.com/en/pro-app/latest/tool-reference/main/arcgis-pro-tool-reference.htm>
- ESRI. (2021). Use LIDAR in ArcGIS Pro. Retrieved March 02, 2021, from <https://pro.arcgis.com/en/pro-app/latest/help/data/las-dataset/use-lidar-in-arcgis-pro.htm>
- MN DNR. (2021). Index of /pub/data/elevation/lidar. Retrieved March 02, 2021, from <https://resources.gisdata.mn.gov/pub/data/elevation/lidar/>
- PRISM Climate Group. (2021). 30 year Normals. Retrieved March 02, 2021, from <https://prism.oregonstate.edu/normals/>
- Python Software Foundation. (2021). Zipfile - work with zip archives. Retrieved March 02, 2021, from <https://docs.python.org/3/library/zipfile.html>

Self-score

Fill out this rubric for yourself and include it in your lab report. The same rubric will be used to generate a grade in proportion to the points assigned in the syllabus to the assignment.

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	22 (did not have time for animation export)
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	91