**Lab Report**

Title: Data Analysis and Spacetime Cubes with MNDNR. las and .bil Data

Notice: Dr. Bryan Runck

Author: Jake Ford

Date: 10-12-2022

**Project Repository:** [**https://github.com/ThisFord/GIS5571-arc1.git**](https://github.com/ThisFord/GIS5571-arc1.git)

**Google Drive Link:**

**Time Spent:** *40*

**Abstract**

ETLs are important and powerful tools for retrieving, analyzing and displaying geospatial data. This lab builds on skills developed in previous labs to deepen understanding and demonstrate the execution of more common geospatial analysis tasks. Using lidar data from the Minnesota DNR and precipitation data from PRISM the lab strengthens data wrangling techniques with two different types of API and builds on visualization and analysis through the creation of multiple static and dynamic visual outputs in two PDFs and a GIF of a space-time cube animation. The lab uses jupyter notebooks in ArcGIS Pro to program the geoprocessing, data wrangling, and output procedures using arcpy and standard python methods. Model Builder in ArcGIS pro is used to generate a flexible and editable model of the complex space time cube creation, which is exported and executed as a python script. The resulting output demonstrates the power and flexibility of ETLs and the advantages of programmatic automation in deterministic spatial analysis. The outputs are checked for accuracy against a set of peer generated papers and programs. Ultimately the lab highlights skills needed to improve efficiency and succeed in deep spatial analysis and provides scaffolding on which to build in future labs.

**Problem Statement**

Working with data in ETL pipelines is essential for proficiency in GIS. This project builds on previous working knowledge of ETL pipelines by incorporating new datasets, implementing exploratory data analysis, and producing a simple geovisualization in the form of a PDF export through ArcPro based Jupyter Notebooks. The project then compares 2D and 3D visualizations of the data side by side and creates a spacetime cube from a dataset spanning 30 years of average precipitation values.

*Table 1. Requirements*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Requirement** | **Defined As** | **Prep** |
| 1 | ArcGIS Pro | Software for geospatial processing |  |
| 2 | Jupyter Notebook in ArcPro | Python programming interface in esri’s ArcPro software |  |
| 3 | Lucid Chart | Model sketching online software | Sketch out workflows |
| 4 | Model Builder in Arc Pro | For building Geoprocessing workflows in Arc Pro and visualizing the process | Create and execute workflows |

**Input Data**

The input data is comprised of a sample set of lidar data provided through the Minnesota DNR and 30-year precipitation annual normals data from PRISM. The lidar data is stored in the .las format, which represents a point cloud of lidar returns with x y and z coordinates. The data is detailed and precise and each point represents a reflected laser pulse in 3D space, it can be used to create digital elevation models and normalized DEMs which can be used to calculate the height of objects in the scene. The 30-year Normals precipitation data is a raster of total rainfall amounts for the contiguous United States compiled annually. The time series nature of the data allows us to display change over time and look for shifting patterns of precipitation by combining many years’ worth of data into a multilayer raster data set and space-time cubes.

*Table 2. data used in the project*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Title** | **Purpose in Analysis** | **Link to Source** |
| 1 | MNDNR example las file | Input data for TIN and DEM creation | [MNDNR API](https://resources.gisdata.mn.gov/pub/data/elevation/lidar/examples/lidar_sample/las/) |
| 2 | PRISM 30yr Normals | Change over time, patter analysis and space time cube generation | [PRISM](https://www.prism.oregonstate.edu/) |

**Methods**

Using an Extract Transform and Load script in combination with geoprocessing tools from the arcpy module forms the core of the methodology for this lab. The first step was to break down the lab into 3 parts which are described in detail below.

**Diagram, schematic

Description automatically generated**

*High level overview of the file saving process and export to pdf function.*

Part one involved creating the ETL pipeline from the MNDNR example las data set data repository through the API using the requests module in python. The figure above describes the processes necessary to download las files from the MN DNR database, transform them to DEM and TIN files, save them to the disk and export the final images as pdfs. The pdfs are included in the figures in the results section.

**Diagram

Description automatically generated**

*High level overview of a process to programmatically set the linked split display in arc GIS.*

Part 2, described above, involved visualizing the las files side by side in 2d and 3d scenes. In ArcGIS pro you can create features and objects with z values in all the abstractions (point line polygon 3d object) that you can use in the regular mapping process. Scenes allow you to rotate the map frame on a x-y-z axis to add and edit features in a dynamic way. 2D features that don’t have a z value can be given them through several interpolative tools, through extrusion or in combination with elevation surfaces (like the DEM we’ve created.) 3D surfaces which have z values will automatically generate objects in a 3D scene. Side by side displays can be linked to display the same extent and zoom level, making comparison easy. The las data sets have a lot of functionality for analysis, you can select display of points based on category of return, elevation, and even class. If classes are unassigned there is even a classification process available. The scenes can be set programmatically using the arcpy. camera method and changing the display value to a GLOBAL or LOCAL scene when referencing the source map and layer elements including but not limited to las files.

**Diagram

Description automatically generated**

*Figure showing the process involved in creating a stand-alone animation of the 30-year normal files downloaded through the ETL.*

Part three involved animating the timeseries and exporting the animation as a .gif file. This process was very involved and quite difficult. I was able to animate the series, but unable to create an animation in ArcPro. Using the tutorial on Esri’s website I created a model (shown in the figure below) to build the multidimensional raster layers from the PRISM data, which were then animated in a loop. Each dimension of the raster layer represents one year of average precipitation data from the PRISM API. The accompanying code downloads the data, creates a mosaic dataset, rasterizes the dataset and fills it with the corresponding values by year then uses the temporal data to sequence and animate the layers and produce a space-time cube.

*Diagram

Description automatically generated*

*Model from ArcPro Model builder for the processing of 30-year precip Normals into a space-time cube.*

**Results:**

I was able to produce the required products for parts one through three, which are displayed below.

Graphical user interface

Description automatically generatedMap

Description automatically generated

*Part 1: DEM and TINs derived from the .las files downloaded from the MNDNR API.*

*Graphical user interface, application

Description automatically generated*

*Part 2: Figure showing the 2D scene of the las point cloud symbolized by elevation, linked to the 3D scene symbolized by class, the pane to the right shows some symbology options for the 3D scene.*

*Table

Description automatically generated*

*Part 3: Figure shows the completed model for spacetime cube creation.*

*Graphical user interface, application

Description automatically generated*

*Part 3: Map product with spacetime cube, and time animation with code snippet.*

**Results Verification**

The results were verified by direct assessment, that the ETL scripts are functional and perform the required tasks in a programmatic way demonstrate their efficacy. Additionally, I used comparisons to peer projects and the collaborative class process to confirm that the scripts were in order and executed properly.

**Discussion and Conclusion**

The ETL pipelining process is a powerful tool for automating data wrangling and data processing. I’ve found that scripting the processes for transforming the data to DEMs and TINS and exporting the results in a new sharable format was relatively straightforward. The advantages to creating a script for a process like this are clear. I was even able to grab the .py file that I had stored on github and run it in a new project when I was unable to access the original ArcGIS project. It worked great and produced identical results in a great example of the deterministic processes at work.

The side-by-side visualization was more difficult to achieve programmatically, but the tools for displaying 3D scenes from a dataset’s z values in the GUI are quite powerful. They include built in classification and visualization schemes that use elevation data to delineate objects and cover type, slope and elevation.

The Space-Time Cube creation portion demonstrated some of the capabilities of the ArcPro Model Builder in creating a workflow. Using the model builder to create a process and then exporting that as a .py file was illuminating. Using this method simplified the complex arcpy geoprocessing tools and clarified the input parameters. I was able to move the different tools around to optimize the order and organize the outputs. Model builder presents a flexible way to take the template models from the planning phase into the execution phase, a step I was missing until now.

**References**

*2D and 3D features—ArcGIS Pro | Documentation*. (n.d.). Retrieved October 28, 2022, from <https://pro.arcgis.com/en/pro-app/2.9/help/editing/introduction-to-creating-2d-and-3d-features.htm>

*BIL, BIP, and BSQ raster files—Help | ArcGIS for Desktop*. (n.d.). Retrieved October 30, 2022, from <https://desktop.arcgis.com/en/arcmap/10.3/manage-data/raster-and-images/bil-bip-and-bsq-raster-files.htm>

Buie, L. (n.d.). Explore your raster data with Space Time Pattern Mining. *ArcGIS Blog*. Retrieved October 29, 2022, from <https://www.esri.com/arcgis-blog/products/arcgis-pro/analytics/explore-your-raster-data-with-space-time-pattern-mining/>

*Creating a cost surface raster—ArcGIS Pro | Documentation*. (n.d.). Retrieved November 6, 2022, from <https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/creating-a-cost-surface-raster.htm>

*Introduction to arcpy.mp—ArcGIS Pro | Documentation*. (n.d.). Retrieved October 28, 2022, from <https://pro.arcgis.com/en/pro-app/latest/arcpy/mapping/introduction-to-arcpy-mp.htm>

*Tutorial: Getting started with arcpy.mp—ArcGIS Pro | Documentation*. (n.d.). Retrieved October 28, 2022, from <https://pro.arcgis.com/en/pro-app/latest/arcpy/mapping/tutorial-getting-started-with-arcpy-mp.htm>

*Using weighted overlay analysis to identify areas that are natural and accessible*. (n.d.). ArcGIS API for Python. Retrieved November 6, 2022, from <https://developers.arcgis.com/python/samples/calculating-cost-surfaces-using-weighted-overlay-analysis/>

**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 for 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 | **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** |