

Dawson Lamb

12-11-25

Dr. Wang

Introduction/Problem Statement:

The current project attempts to solve the problem of importing a JSON dataset containing Well-Known Text (WKT) geometries from the 2018 Land Value records for New Orleans.

Current ArcGIS programs do not allow users to process JSON files containing WKT geometries. Thus, other methods are needed to accomplish this task. Using Python script can allow the user to convert the JSON file, and the included geometric functions, to a shapefile, and visualize the data in ArcGIS Pro.

The primary goal of this project is to develop a Python/ArcPy script tool that converts a JSON file containing WKT geometries and attribute data into an ArcGIS dataset that can be visualized.

Data Sources:

A file titled “no_tax.json” was utilized for this project. The dataset includes parcel boundaries and land value attributes for New Orleans from data.gov.

Before visualization of the data was possible, the JSON data needed to be converted into a shapefile in order to be accessible in ArcGIS.

To begin, the usable data needed to be extracted from the JSON file. The ‘Meta’ and ‘Data’ sections of the file were important for future visualization purposes. 'Meta" includes descriptions of the fields, while "Data" included field values and geometries. Before visualization of the data was possible, WKT strings needed to be converted to ArcPy geometry objects using arcpy.FromWKT(). Additionally, arcpy.da.InsertCursor was utilized to write the geometry and attributes to the shapefile that would then be used in ArcGIS.

Methods and Tools:

A variety of processing software were implemented to process the data. VS Code was the primary coding source used for this project and was used to develop and test the script. Once a shapefile was created, ArcGIS was utilized to visualizes the data and geometrics, but also create a layout and execute script tools.

Inside of these programs, various tools were utilized to read files, extract data, and create a wide array of functions. ArcPy was commonly utilized to construct feature classes, fields, geometries, map frames, and layout elements including a scale, arrow, and more.

GeoPandas was used in this project to validate WKT outside of ArcGIS. The ‘os’ function was commonly used as it allowed the script to interact with the operating system.

Challenges:

Several challenges arose during the project that resulted in delays in completing the tasks. Primarily, the JSON feature is unable to be processed in ArcGIS with preprocessing steps. The JSON file must be parsed manually, and conversion to a shapefile is needed in order to visualize the data and geometrics. Additionally, WKT may include complex polygons or multipolygons that require careful analysis to avoid geometry errors. It can be difficult to create code or scripts that function without errors due to this complexity.

Timeline:

The following timeline represents the period in which this project was completed. In week 1, the JSON structure was analyzed and parsed through. In addition, the script tool was constructed in order to create a shapefile. In week 2, the shapefile was inserted into ArcGIS for visualization. Week 2 also involved constructing a script tool, and adding layout features such as a scale, north arrow, and legend, as well the creation of the final report.

Conclusion/Future Directions:

This project successfully demonstrates how to process non-standard geospatial JSON data containing WKT geometry and convert it into a fully functioning GIS dataset. Various modules and libraries, the script tool parses the JSON file, converts geometry, writes records to a shapefile, and creates a map layout in ArcGIS Pro.

This project proves ArcGIS Pro’s capabilities to handle data files that are not supported by the program’s default formats. As a result, other unsupported formats can potentially be converted into formats or files that can be implemented into ArcGIS for various purposes including visualization.