Unsupervised Clustering of Landscape Classes for Acoustic Sensor Deployment Planning

A Tutorial

# What are you trying to do?

Say you need to deploy a large collection of acoustic sensors in a scheme that will track variations in local land cover, but you don’t have a clear picture of the relevant habitat classes in which you’d want to deploy. Maybe you want to capture variation in forest structure, but all the data you have so far simply shows areas that are forested with minimal information as to what kind of forest you’re working with. However, you want to make sure that your deployments are capturing an ecological gradient that can be empirically supported. What can you do?

While not every location in the world will have the same access to data or the same particular ecological questions and challenges, this tutorial can provide a conceptual framework for developing these kinds of unsupervised land cover classifications and will describe the processes used to develop the underlying data and evaluate the results.

The case study for this tutorial will be Tippecanoe County in Indiana, in the United States. Here, the project goal was to deploy several dozen acoustic sensors to collect ground-truthed biodiversity data as part of a modeling exercise integrating passive acoustic monitoring with space-based remote sensing on the International Space Station: the GEDI Lidar sensor, the Ecostress temperature sensor, and the DESIS hyperspectral sensor

This tutorial assumes at least a basic level of familiarity with the R programming language, including installing and using extra packages, and with ArcGIS Pro. If you are using alternate GIS software (e.g. QGIS) the principles at play are the same but you may need to adapt the workflow somewhat. An R script file should be included with this tutorial, which contains the code run in the R Analysis section and is highly commented for your reference.

# First Step: Identifying Potential Drivers and Variables

Not every ecosystem or study context will have the same variables available, or even the same variables needed. In the context of our case study, we needed variables that would track land cover class—in our county, generally a mix of forest remnants, urban areas, and open agricultural lands; as well as gradients of human disturbance, which can be quantified in a broad variety of ways. We also knew that we would have access to high-quality LIDAR data from state flights and wanted to integrate these metrics of structural complexity into our analysis.

We looked at two different landscape-level sets of metrics, plus lidar metrics. Our landscape-level metrics were grouped into two broad classes: distance metrics, where we evaluated how far away a location was from the closest instance of some feature that would likely influence the local soundscape (for example, distance to nearest open water); and prevalence metrics, where we looked at how common a particular habitat type was in the area around a point (for example, amount of wetland within a 1km radius). Each of these drivers was evaluated across a range of land cover classes and landscape features, using the US’ National Land Cover Database as a basis for the analysis.  
 We also needed to choose a resolution at which to do our analysis. This will depend on the data being used as well as the sensors in question. With the Song Meter 4s we planned on installing, the sensors should always be deployed at least 500 meters away from each other, so the size of the resulting pixels should be significantly smaller than that. We settled on a 30-meter resolution for our analysis since we knew that many of our data products would be derived from the LANDSAT sensors which provide data at this resolution.

# Second Step: Blank Raster Creation

The analysis will be done using a 30m grid of points, covering the entire area of the county we are considering. In order to generate this grid, do the following:

1. Generate a polygon boundary for the area of interest. In our case, this was taken from a USGS dataset of county boundaries and extracted to the area of Tippecanoe County, but you may want to use less political and more natural boundaries (e.g. some collection of watersheds) to do this analysis depending on your research question.
2. Turn that polygon into a raster using the Polygon to Raster tool. Make note of your projection, as this will need to remain consistent throughout the process so that errors do not occur (Figure 1).

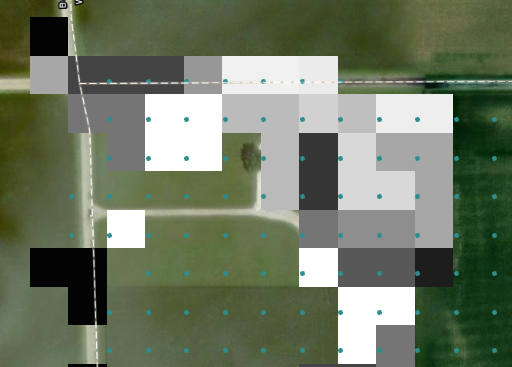


Figure 1: an example error, where a raster and point grid were made with different projections, which can throw off future steps in this analysis. The points should be centered in the raster cells but due to projection issues they are not.

1. Remove areas you don’t want to consider from this blank raster. This may include some urban areas or those that are situated on open water where you can’t deploy a sensor. You will need to determine exactly which areas make sense to use or not as your study necessitates and determine how to most effectively remove those from your blank raster, though “extract by attributes” is likely the best tool to use to accomplish this.
2. Use the Raster to Point tool to turn the now-subset blank raster into a grid of points. Each point will be the center of one of your existing 30x30 pixels, and should therefore be on a 30-meter grid. We will use these point IDs and coordinates as a merge field in later analysis.



Figure : Example point grid for use in analysis. The distance between points is 30 meters.

1. Take your point dataset, and add two new fields. Call them Lat and Lon or similar
2. Use the Calculate Geometry Attributes tool to add latitude and longitude to your point dataset. Make sure it’s formatted in decimal degrees.

# Third Step: Data Collection and preparation

One of the most critical datasets in our analysis is the US National Land Cover Database, developed using data from the LANDSAT program, available at a continental scale at 30x30 meter resolution, and updated every few years (as of writing, the most recent version of this dataset was for 2019, with versions available roughly every three years before that). This dataset can be downloaded from the US’ Multi-Resolution Land Characteristics (MRLC) consortium, with the 2019 NLCD available [here](https://www.mrlc.gov/data/nlcd-2019-land-cover-conus). Download the NLCD raster as a zip file, extract it, and add it to your project in ArcGIS Pro, and clip the raster to an area larger than your blank raster/point grid, but not too much larger—a few kilometers should be about enough of a buffer. Making this raster smaller than its default continental US scale will speed up processing later in the workflow.

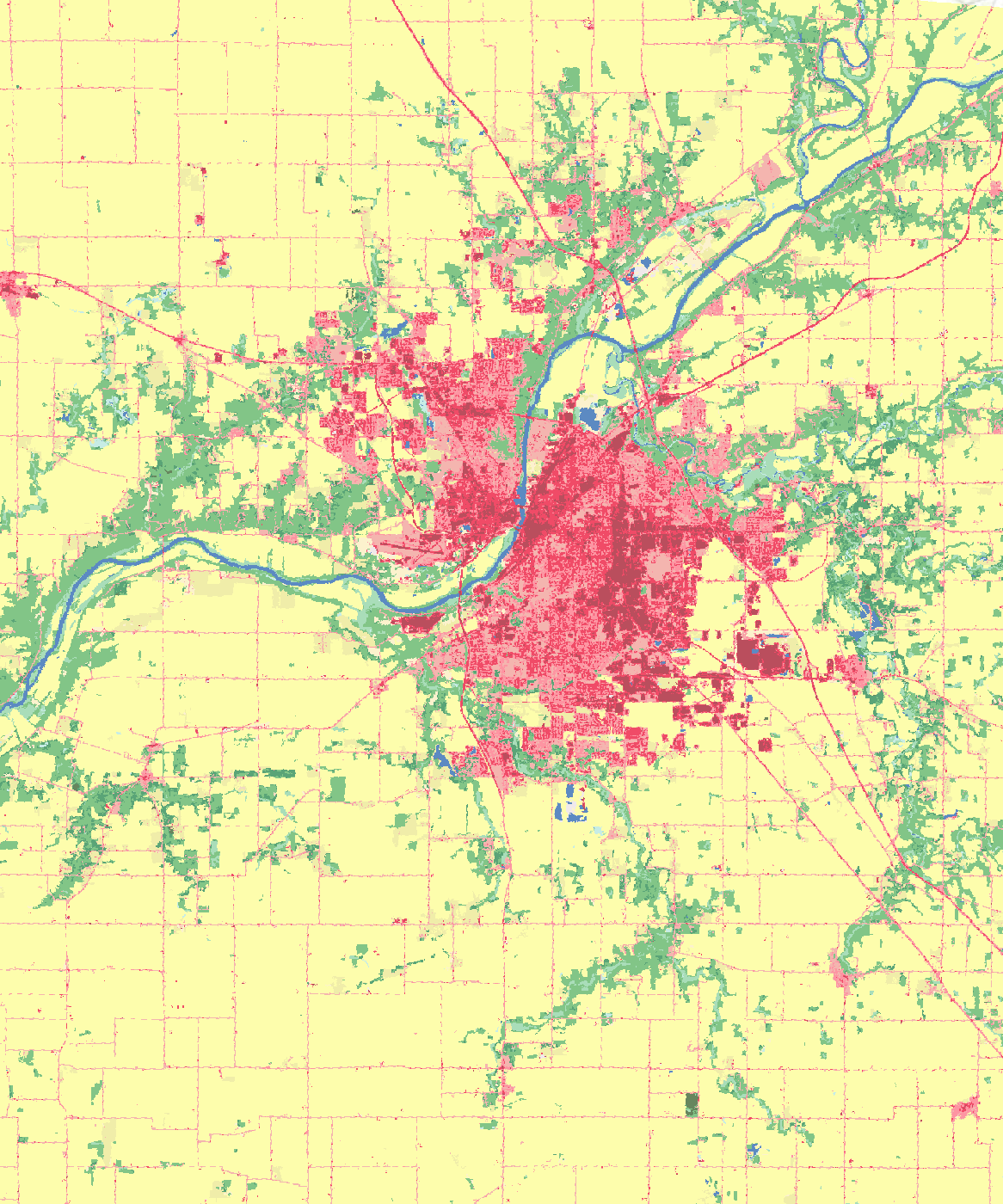


Figure : 2019 NLCD in the study area around Tippecanoe County Indiana

Depending on your local context, you may wish to exclude or merge some classes from NLCD. For example, in our case study, there was only a very small patch of non-deciduous forest, so the three NLCD forest classes (deciduous, coniferous, and mixed) were merged into a single forest class.

You should also download the NLCD Impervious Descriptor dataset, which breaks out some of the areas listed as “urban” in the main NLCD into a more detailed set of classes, which will be useful later. For this dataset, you should create separate rasters for each road class by extracting out primary, secondary, and tertiary roads. These are roads of different sizes and levels of traffic, which will likely influence the local soundscape in different ways and therefore should be treated differently.

You should also extract the non-road non-energy impervious class from this NLCD Impervious Descriptor raster, which will be useful for more carefully delineating urban areas you want to use for your analysis—the NLCD overall shows roads as urban, for example, which is not necessarily desirable in a rural environment where a road may not have a significant impact on the local soundscape.

For each NLCD class that you want to use in your eventual analysis, you should use the Extract by Attributes tool to select the individual classes you’re interested in and turn them into unique rasters, so you might have a forest raster, a pasture raster, an urban raster, a cultivated crops raster, et cetera.

# Fourth Step: Buffer Analysis

This will determine the prevalence of various land-use/land-cover classes within a given distance around each potential point/pixel. The scale of these buffers will depend on how patchy and heterogeneous your landscape is, but using two separate buffers—250m and 1km—is a good starting point. The smaller buffer is roughly the area you might expect some sounds to carry, while the larger buffer is where you might expect wildlife to travel to or from in the local area.

Use the Buffer tool to create your buffers around the points, using your preferred distance. Most of these will be short enough that the curvature of the earth will not matter and you can use a planar distance parameter rather than a geodesic one. However, you should be mindful of the scale of analysis. In most cases planar will also be faster to calculate, though geodesic is more accurate especially at very long distances (100s of kilometers or more).

Then, use zonal statistics to evaluate the area of each class in each buffer, and export the relevant tables as CSVs for use in later analysis. You can get either a percentage of area in that class, a measured total area, or a number of pixels. However, since the buffers are all the same size and the pixels are all the same size, these three metrics should be perfectly correlated and you can use whichever you prefer as long as you are consistent throughout all your repetitions of this process, which you will need to do once for each buffer size you plan on using and once for each class you plan on evaluating.

# Fifth Step: Distance Analysis

This protocol evaluates the distance to the nearest instance of a given class. You will need to think through some things for your particular research question as part of this topic.  
 First, you will need to convert each of your land cover classes to polygons, since the tools that ArcGIS pro uses use only polygon-class datasets as their target.   
 Then, you will need to work to identify where there are polygons you do not want to consider, and determine an effective way to exclude these from the analysis. For example, NLCD datasets will often consider individual homes/buildings in the middle of large areas of agricultural land as urban, as long as they are bigger than one pixel (30x30 meters), but these would not have the soundscape impact that one might expect of a town or city, so it may be advisable to exclude very small urban polygons from analysis (Figure 4). In this case, the exact threshold or criteria that you use will depend on your data and may take some degree of trial and error to find one that ends up with a result that looks roughly correct.

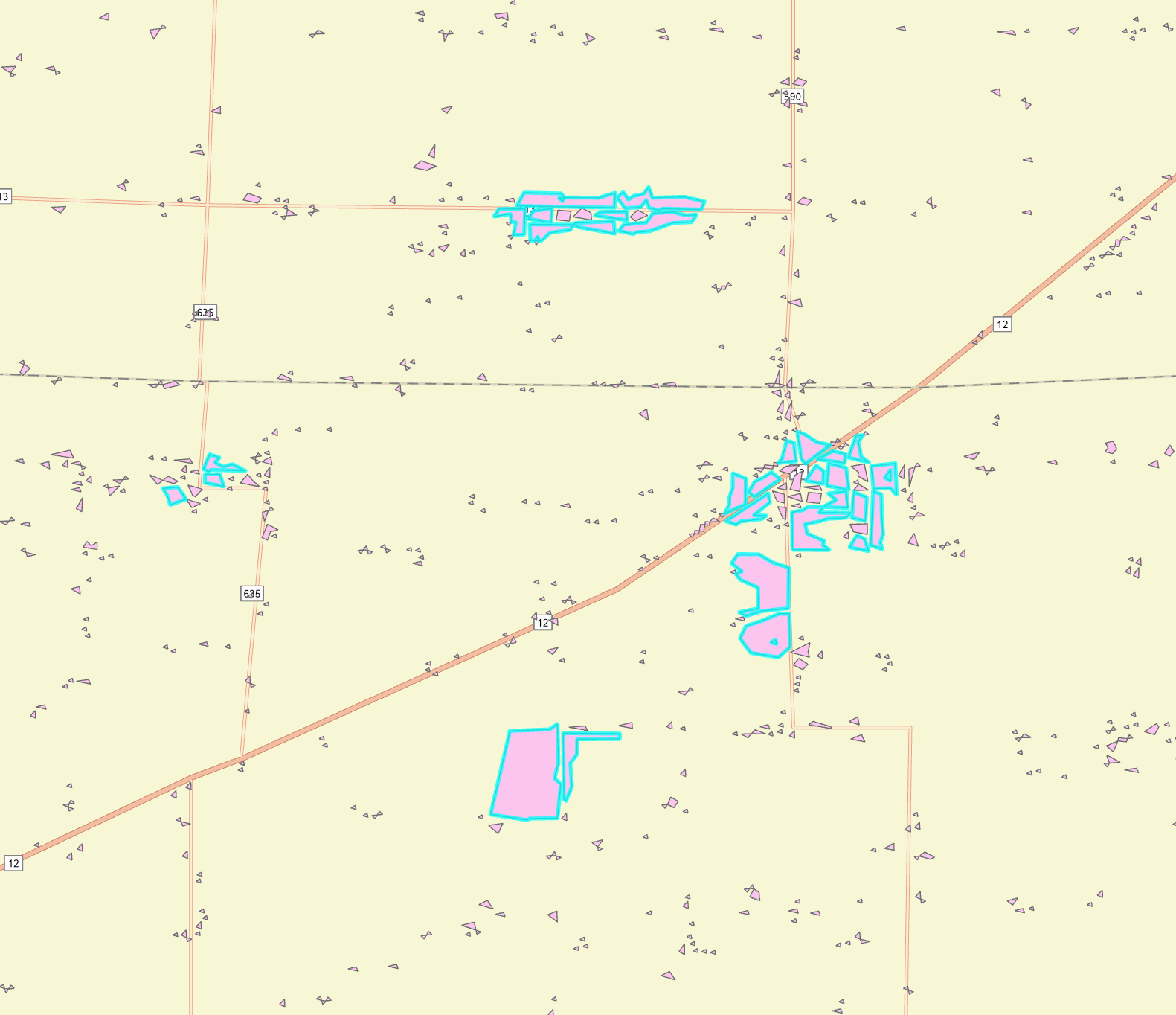


Figure : Example polygons from the NLCD impervious descriptor dataset. Here, we are selecting only those polygons that are >10,000 m2 (those that are outlined in blue) to ignore small individual-building pixels that might otherwise throw off the analysis

Once you have filtered your data to the point you want, you should use the “Near” tool to determine the distance to the closest instance of each class for each point. This tool only gives you a single value. For example, if there are roads 100m north, 125m west, and 300m south of a point, the only value you will get will be the 100m value. You should then export each of these distance calculations to a separate CSV.

# Sixth Step: LIDAR analysis

Using LIDAR can allow a researcher to tease out more complexities in the landscape or ecosystem than simple land-use data by quantifying structural features in the landscape. It does this by measuring laser returns off vegetation in ways that can evaluate 3-dimensional structure at relatively fine resolution.

Important to bear in mind when working with LIDAR data is the time of year that the LIDAR flights, whether by UAV or by fixed-wing aircraft, were done and how this will impact results. In temperate climates, where crops will be planted, grown, and be harvested or where deciduous vegetation will grow and lose its leaves over the course of the year, this is especially important. In the case of our case study, the flights were done in April, before the most common crops in our study area (corn and soybeans) reached more than seedling size, but after trees and shrubs had started growing their leaves.

LIDAR calculations are done based on the cells of the blank raster you generated in step one, averaging the results of calculations performed on hundreds or thousands of returns per cell depending on cell size, and a number of metrics can be calculated from the points returned off of vegetation in that particular cell.

First, we can look at a number of vertical gaps (nVG). This attempts to identify layers in the return that likely correspond to layers of vegetation (canopy, mid-story, understory) working with the assumption that older-growth forest areas will have a greater number of these gaps. This will require identifying a threshold value at which we can define a new layer of vegetation as starting (Fig.7, in which the NVG would be 2). Identifying this value in your research context may take some trial and error and examination of your data. A lower threshold value may give you better NVG values but may also create a more noisy signal (see figures 5 and 6). For example, if you look at the northeastern corner of figures 5 and 6 we can see the algorithm picking up a power transmission line when the threshold is 0.01 and ignoring that line when the threshold is 0.04.

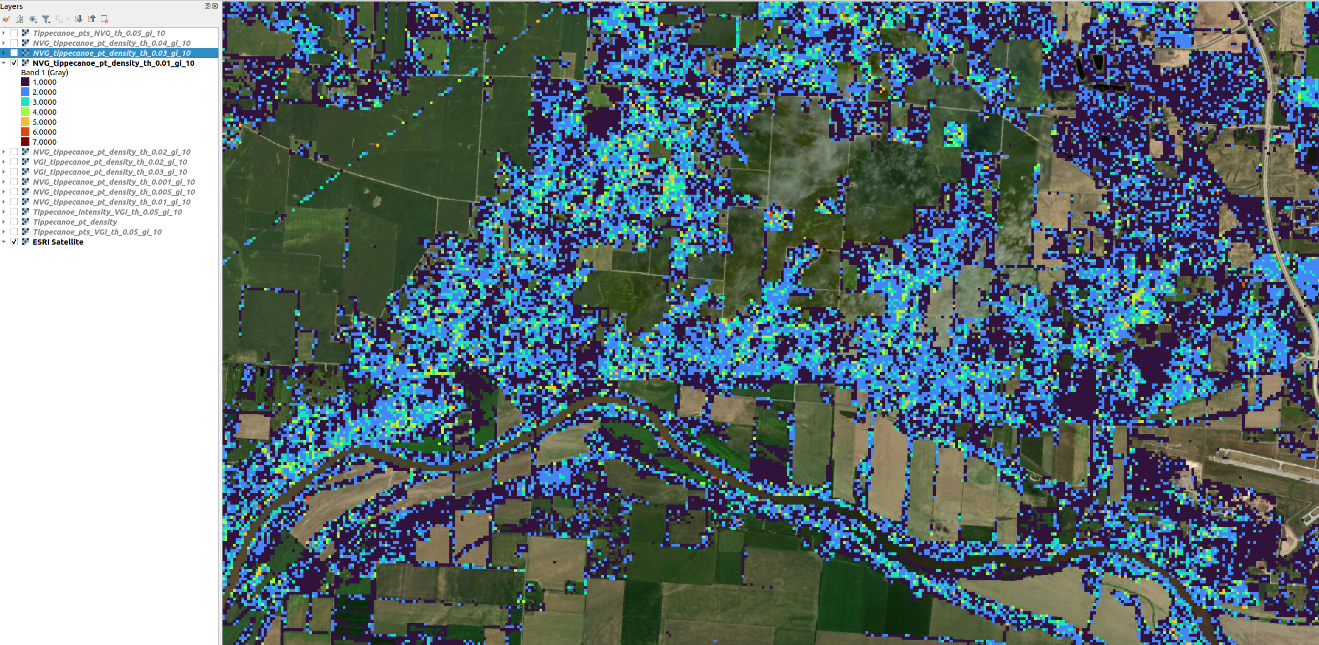


Figure 5: NVG with threshold set to 0.01

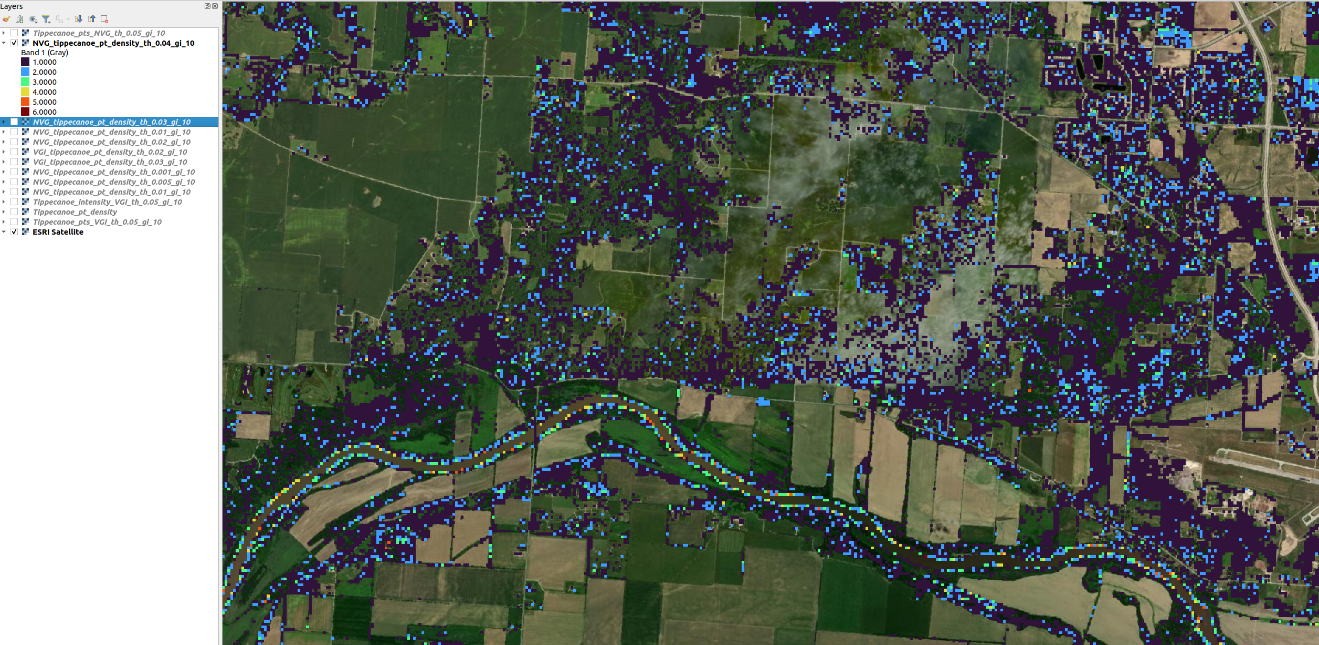


Figure 6: NVG with threshold set to 0.04

Second, we can look at a related metric, the Vertical Gap Index (VGI). This is a ratio of the total height that is below some threshold value (i.e. it is the ratio of canopy/leafy vegetation to open space). In Fig.7 it would be (L1 + L2) / Hmax. As with nVG, it depends significantly on your choice of threshold value which may take some trial and error.

Diagram

Description automatically generated

Figure : Vertical Gap concept, from Jung, Bekin, and Pijanowski 2013

Third, we can look at a percentage of returns above some threshold elevation above ground level. As mentioned before, this will depend on the local conditions and time of year for the lidar flights (e.g. for crops) but can be useful to exclude ground vegetation, and even differentiate orchards from planted crops. In our case, a threshold elevation of 1.5 meters was used, largely to exclude planted crops—while corn will regularly exceed this height, it would not have in April.

It is going to be useful to at least roughly map these LIDAR metrics in ArcGIS to ensure that they match at least roughly with what you would expect to see based on your knowledge of the landscape before moving on to the next step of the process. If the LIDAR metrics are provided in a raster form, you should convert them to point data, ensure that those points correspond exactly to the points you made in step 2, and calculate the latitude and longitude for these points. This is so that you can merge using this information because point IDs do not always match perfectly. You may also want to do a buffer analysis to calculate the mean value of the NVG or VGI for each point as well before proceeding.

# Data you should have at this point, before proceeding

1. Table of points, containing a point id, a latitude, and a longitude. Each point is the center of a raster cell you made in step 2.
2. A table of buffer values you made in step 4. There may in fact be multiple tables of this kind depending on how many different buffer distances you used.
3. A table of distances you made in step 5
4. One or more tables of LIDAR metrics you made in step six

# R Analysis

This section of the tutorial will walk you through the R script used to generate the clustered results, but you should have the code open as well to follow along. Steps 1 through 6 can be useful for a variety of non-kmeans-clustering machine learning approaches to reduce the overall number of predictor variables and speed up eventual processing, though the full tutorial does assume the use of k-means clustering.

1. Read in all your datasets. If you exported tables from ArcGIS Pro in DBF format, you will want to use the package “foreign” which can read these (lines 14-19). If you have already converted them to CSV, you can use the built-in read.csv() function instead.
2. Perform any necessary data cleaning. For example, you will want to make sure that columns are named consistently including capitalization and formatting (lines 22-81), and that these column names are not duplicated (for example, ArcGIS exports raster values as the “grid\_code” field name, which you will need to rename to the relevant data type so that they can later be merged successfully without overlapping
3. Merge all your data into a large table containing point locations, LIDAR metrics, buffer and distance data. You will likely have a few dozen columns and as many rows as you have points, so doing this on a powerful computer with a lot of memory is advisable. You may wish to do some filtering here to remove points you don’t want to cluster (line 85-90).
4. Make a cross-correlation plot using the “lares” package and use the results to remove one of each pair of highly correlated variables (lines 92-109). This plot evaluates how many of your variables are tightly correlated with each other, and in case where variables are tightly correlated they are effectively providing redundant information. In this case, the variable pairs in blue are positively correlated and the variables in red are negatively correlated (Fig.2). You can use this as a first-pass reduction in the variables you want to use for your eventual analysis, since two highly-correlated variables are not adding much in the way of additional information. In our case study, we took any variables where the correlation was greater than 0.65 (vertical line below) and the variable being considered was the same (e.g. Evergreen\_Forest\_300 and Evergreen\_Forest\_1000) and removed the 1000m version of this variable; this allowed us to eliminate 9 variables in this first pass. Your use case may vary, and you should pick a threshold based on what you see. In this case, we can also evaluate the plot conceptually, and understand that for example, it makes sense that Deciduous\_Forest\_1000 and Cultivated\_Crops\_1000 are negatively correlated—densely cropped areas don’t have much forest and vice versa.

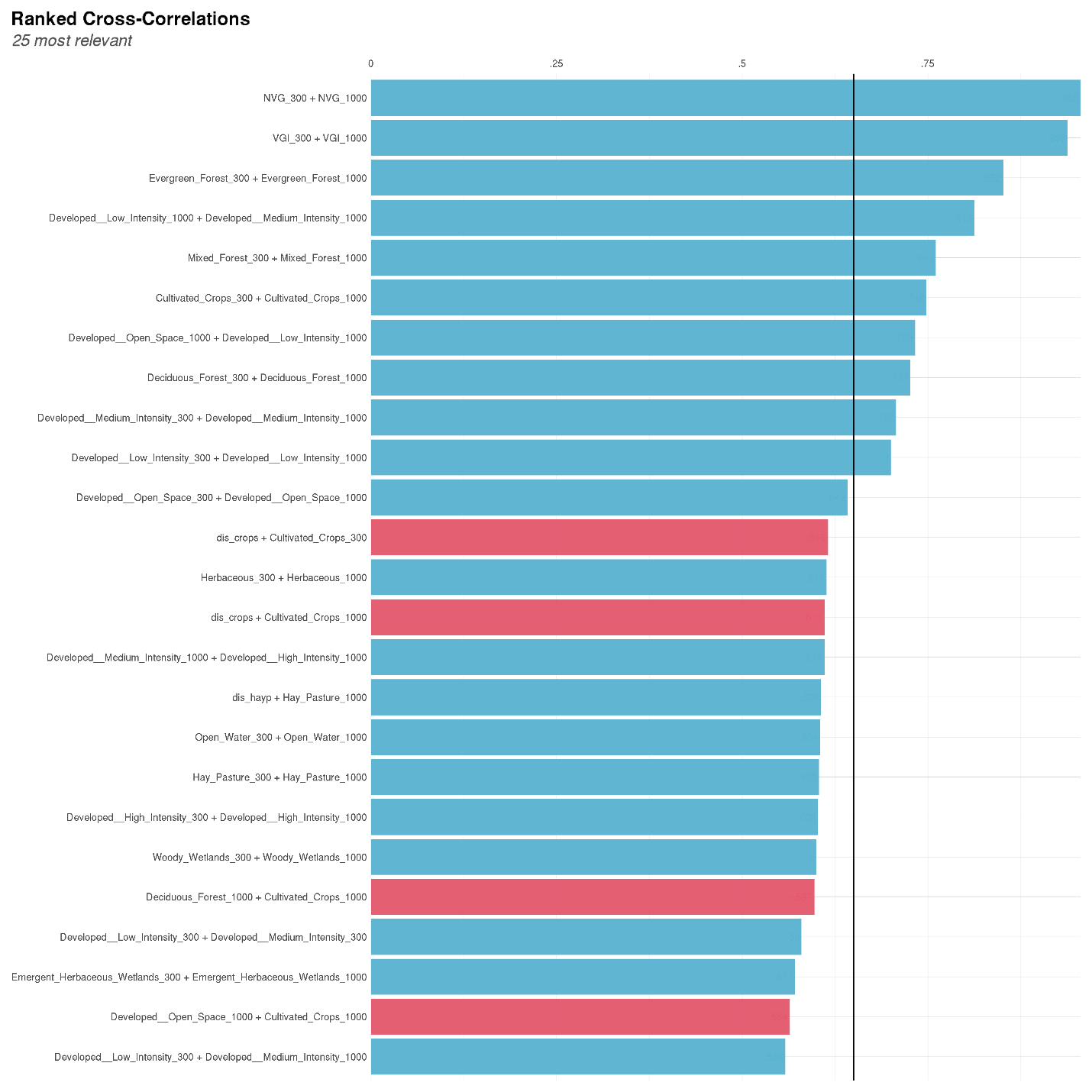


Figure : Example cross-correlation plot

1. Run a Principal Components Analysis (PCA) on your data. This develops new combined indices that will explain as much as possible of the variation in your data with as few variables as possible (line 115)
2. Using the “factoextra” package, extract the values that drive each principal component axis and record the first few most important contributors with their weights. This can help you figure out what these values correspond to in the real world or what kinds of patterns they are picking up on. You should also make a PCA biplot and screeplot to help figure out how many principal components to use moving forward (lines 121-150)
3. Now you’re ready to move forward into a k-means clustering process. The first thing you should do is create scree plots to figure out the number of centers to use in your kmeans (Lines 156-160). There are multiple methods of evaluating the different numbers of clusters that the “factoextra” package provides and you should look at all of them, in hopes that they agree on an optimal number of clusters. You should also subsample your data to improve computational speed here (in our case, we used 10,000 points)
4. Now you can actually run your kmeans and evaluate the results (lines 163-164). You may wish to shuffle the iter.max and nstart parameters, since higher values for these will give you a greater probability of convergence but will also increase the amount of time it takes to run the kmeans
5. Generate a new clusters-by-point table (line 166-178). This will involve merging the results of your kmeans clustering with the point IDs and coordinates, as well as the principal component values
6. Now you should plot your new set of points to see if the kmeans results are actually corresponding to something in the real world. This can be done in R, where very basic map creation code is shown (lines 183-193). However, that is not production-level and should not be used for any kind of report or presentation without extensive modification, but it is useful for a quick evaluation of your results, or you can bring the data into ArcGIS Pro. This may also be a time to add names or descriptors to your classes for any future discussion without confusion. There is no real framework for doing so beyond using your pre-existing knowledge of the ecosystem.

# How to think through your results

Sometimes, running this clustering may end up simply recreating a pre-existing land-use/land-cover map (Figure 9). In this case, while your analysis may have been a useful exercise, simply evaluating an NLCD dataset may be not useful. There are two potential solutions here. First is to simply feed the k-means algorithm a higher number of centers to attempt to cluster the landscape in a more granular way. Second is to ignore classes where you don’t actually think there is much variation to capture, and focus on the areas where you want to really capture variation. For example, Figure 10 shows what it looks like when we apply clustering to only the forest area and break that down into a much more granular set of classes.

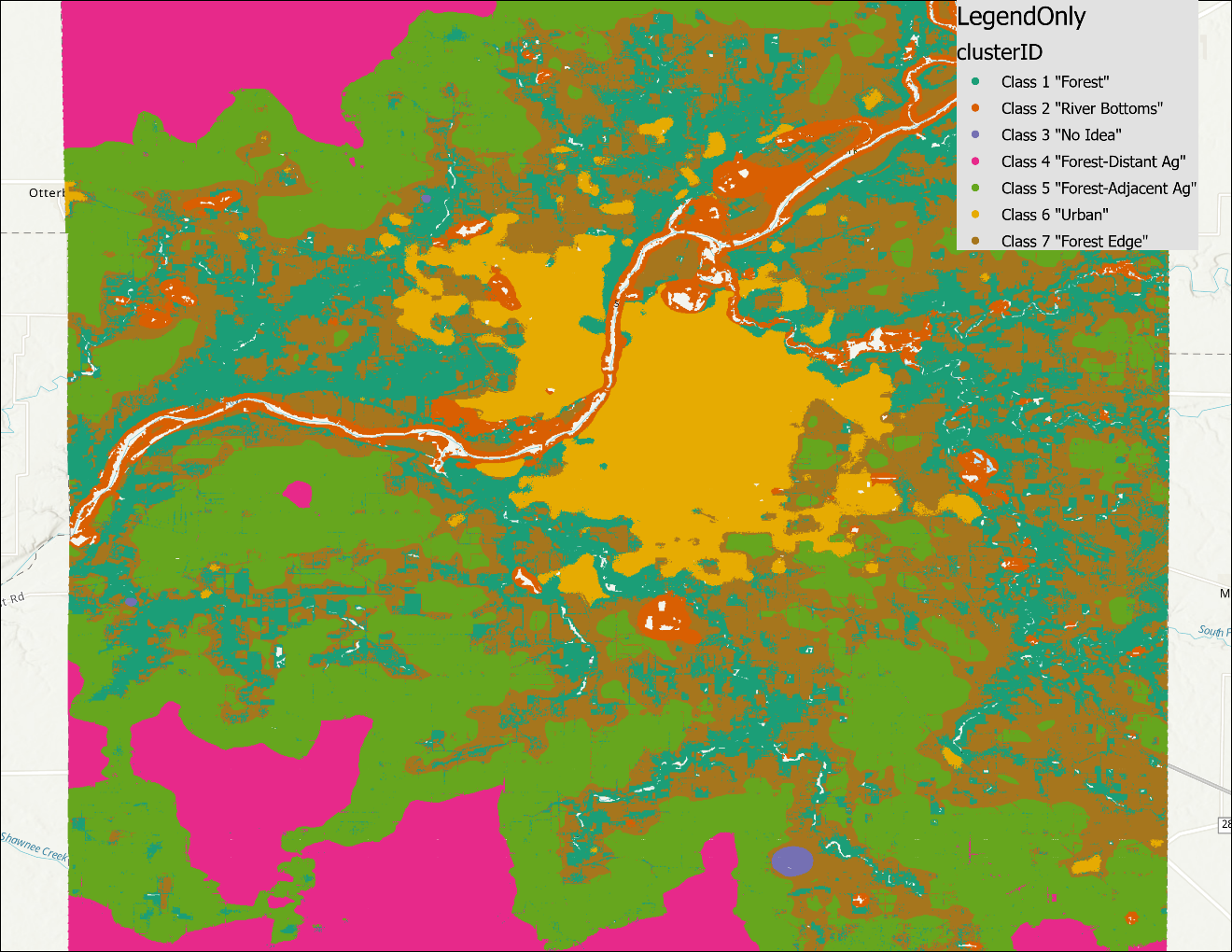


Figure : Clustering results

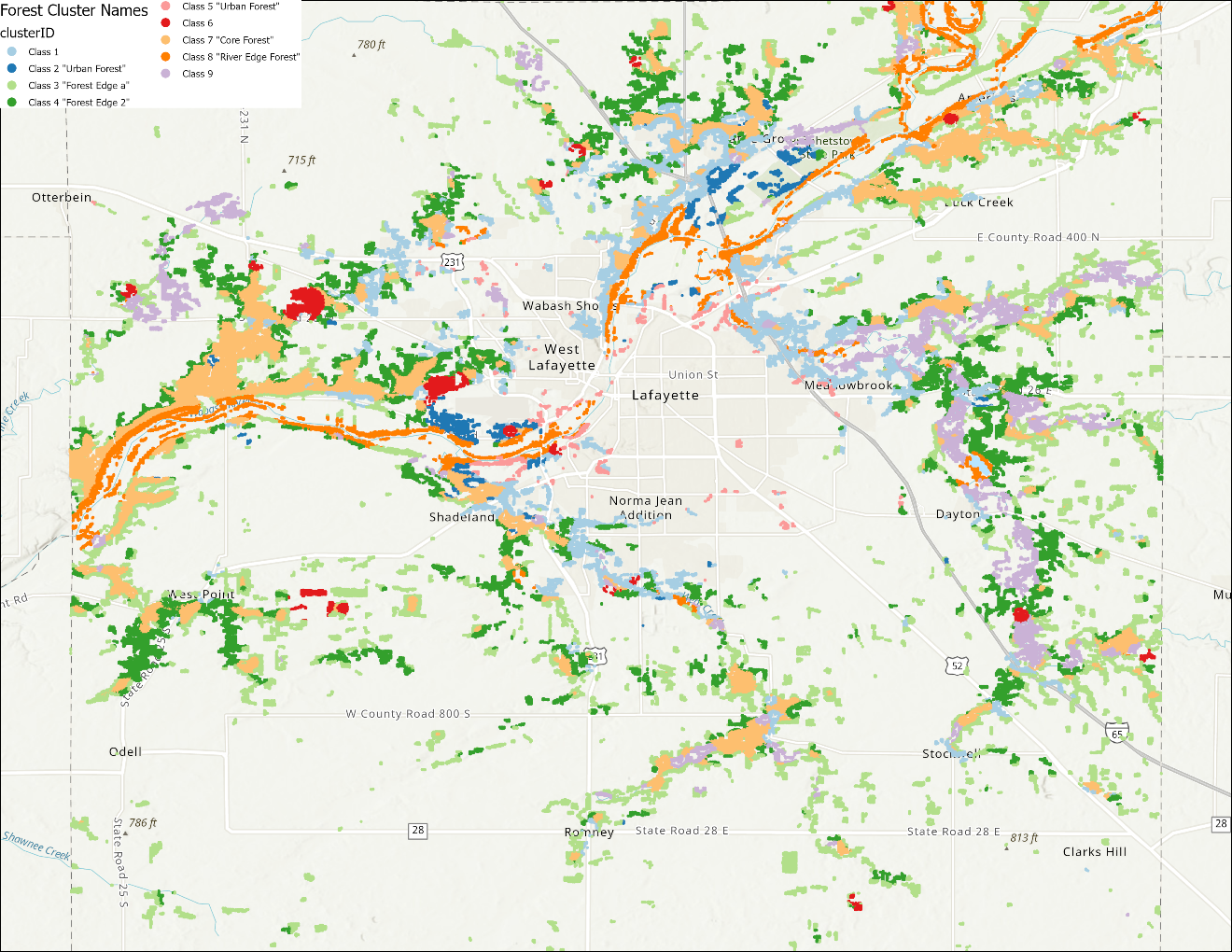


Figure : Cluster results when only the forest area is clustered to provide more granularity.

# Crop Data Layer Analysis

In order to determine the crop rotation patterns for a given area you will first need to know what the likely rotations will be. In our case study, this was primarily corn and soy with intermittent fallow. You will need to think through the likely impacts of different rotation patterns in your area. This tutorial will walk you through the choice of analysis parameters for Tippecanoe County but this part of the protocol contains far more potential sources of variation than some of the other portions.

We will look at sequencing in two ways. The first way is to simply download a frequency layer from the [National Agriculture Statistics Service](https://www.nass.usda.gov/Research_and_Science/Cropland/Release/) that contains the frequency of four crops: corn, soy, cotton, and wheat. In this case, frequency is simply a number of years between 2008 and 2021, so a “1 year” frequency does not provide any information about which year was the 1.

To get a more detailed set of information, and to get information about fallow fields which are likely important for this analysis, you will need to download individual years’ CDLs and combine them to generate sequence information. To do this, you will need to:

1. download each year’s CDL from NASS. Import them into ArcGIS Pro.
2. Clip them to your area of interest boundaries to speed up future processing. This can be done either by downloading a national dataset and then clipping it using ArcGIS Pro’s tools, or by using the NASS dataset pre-clipped to boundaries—it allows for a state or county clip.
3. Reclassify your rasters to only be the classes of agriculture that you’re interested in. In our case, this was corn, soy, other commodity crops, other crops, and fallow area—reducing dozens of classes down to five.
4. Generate sequences for the CDL. The best way to do this, we have found, is to use the raster calculator function to generate a multi-digit sequence. For example, if we wanted to look at a three-year sequence from 2018 to 2020 we would use a raster calculator formula along the lines of: (100 \* CDL\_2018) + (10 \* CDL\_2019) + (CDL\_2020). This will result in a value like 315 where 3 is the class in 2018, 1 is the class in 2019, and 5 is the class in 2020, but the more years you use, and the more potential classes you are interested in for each given year, the more potential sequences you will need to work with. Eventually, you will need to simplify this data.
5. Reclassify your sequences into broad classes. This will depend a lot on local context and your research question and you will need to think through the classes. Shown below is a local example looking at ten-year patterns of corn, soy, and fallow. We’ve been able to reduce hundreds of different sequences of crop rotations over this ten-year period to six meaningful classes that we expect to be different ecologically and therefore be useful for answering our questions

|  |  |  |
| --- | --- | --- |
| **Sequence Type** | **New Value** | **Description** |
| 1111111111 | 1 | All Corn |
| All Corn no more than 3 Soy | 2 | Mostly Corn |
| 3-7 Soy Rest Corn | 3 | Rotated |
| All Soy no more than 3 Corn | 4 | Mostly Soy |
| Up to 2 years fallow | 5 | Light Fallow |
| >2 years fallow | 6 | Heavy Fallow |

1. Using the LIDAR data we generated earlier in this process, you can identify some area that might be considered orchard or other denser-canopy cultivated crops. As mentioned above, this will require some choice of threshold
2. Map your results to confirm that they make at least some level of sense in the context of your real-world knowledge of the systems at play (for example, that you do not have agricultural land plotted in the middle of where you know there to be a town)

