

Unit 42 - Using Map Algebra

Contributed by: Chris Allen, The Clark Labs, Clark University

Context

Map Algebra is one kind of derivative mapping; new data layers are derived from existing data layers through combination and transformation. A GIS typically provides tools for combining and/or transforming map layers mathematically. We might find that this is necessary because an analysis we are conducting requires that data values for one or more data layers be numerically processed and/or combined according to some formula or equation. For example, if we have an equation that predicts temperature as a function of altitude, we may wish to derive a temperature map from a digital elevation model. Or, as another example, consider the possibility of creating a soil erosion potential map given data layers of soil erodability, slope, and rainfall intensity and an equation relating soil erosion to these three factors. Clearly we need the ability to modify data values in our map layers by various mathematical operations and transformations and to combine factors or data layers mathematically to produce a final result.

More Discussion on [Map Algebra](#) (including graphics)

Learning Outcomes

Awareness

Students will understand:

- that Map Algebra allows one to use GIS data layers as variables in algebraic/mathematical expressions
- why there is a need for transformation of data layers to provide proper input to a model (e.g., by scale or range-of-data-values conversion) or for applying a model to analyze data.
- that combination of data layers is a powerful tool for derivation of new data layers.
- the difference between applying a scalar, a function, or another data layer to a data layer to create a new output layer.

Competency

Students will understand:

- how to perform each of the three types of data layer transformation: data layer with scalar, data layer with function, and data layer with data layer.
- how a mathematical equation can be implemented using a sequence of Map Algebra steps to derive an analytical result.

Mastery

Students will be capable of:

- working back from a desired outcome or given equation to determine the sequence of data layer manipulation steps required to derive a desired data layer result.

Example Application

Imagine that we have a simple model for estimating runoff from an area and we wish to apply it to a study area to estimate the total runoff in a precipitation event. The equation for the model is:

$$(S * C * P)/160 = R$$

where:

* indicates multiplication, / indicates division,

S = the surface slope, categorized into values of 1 (0 to 3 degrees), 2 (3 to 6 degrees), 3 (6 to 9 degrees), or 4 (greater than 9 degrees),

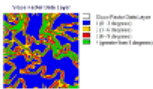
C = ground cover coefficient, a value of 10 for dense, broad leaf cover, 20 for grass or mixed coniferous forest, 30 for sparse canopy forest, and 40 for bare ground,

P = Precipitation in millimeters, and

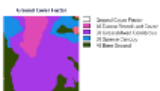
R = Runoff volume of water, in liters per square meter

We start with the following data layers, each has 201 rows by 201 columns; each cell covers a 20-meter by 20-meter portion of the study area:

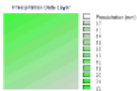
Categorical slope image, containing values from 1 to 4



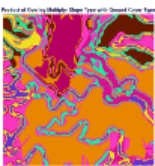
Ground Cover for our study area, containing values of 10, 20, 30, and 40:



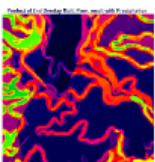
Precipitation for our study area for a particular rainfall event, values range from 12 mm to 22 mm:



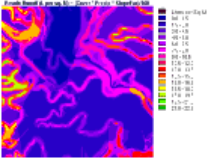
In order to perform our analysis, we first overlay-multiply our slope category image times our ground cover category image. This overlay-multiplication step creates a new image in which each cell contains an attribute value equal to the multiplication product of the two corresponding cells in the input images:



We take this image, which resulted from the first overlay step and overlay-multiply by our precipitation image to produce this image:



Next we perform a scalar operation. We divide this image by 160 to produce this result:



For each cell in our raster data layer, we have estimated the amount of runoff in liters per square meter. Each cell covers an area of 400 square meters. We can transform this data layer from representing liters per square meter to liters per cell by performing a simple scalar multiplication of this data layer times 400.

For example, if a particular cell in our resulting image above has an attribute value of 15.2 liters per square meter, then the total runoff for that cell is:

$$15.2 \text{ liters per square meter} \times 400 \text{ square meters} = 6080 \text{ liters.}$$

An image is not provided to show the result of the scalar multiplication of the raster result above (Runoff in liters per square meter) times 400 because the image looks identical. The only difference is that the attribute value in each cell in the new image (Runoff in liters per cell) is 400 times larger than the value in the image above.

If we wish to compute the total runoff in our study area, we need to sum the attribute values in all of the cells in our raster (Runoff in liters per cell). This is a fundamental type of database query operation, and the capacity to perform this analysis is likely to be found in any robust GIS. In Idrisi, an easy way to sum up all of the attribute values in an image is to create a "blank" raster - an image containing one value, zero, for instance. This blank raster has the same number of rows and columns, spatial extent, and reference system as the rasters we have been working with. Next we use the EXTRACT module specifying that the "blank" raster is our Feature Definition Image, and the image we wish to sum is our Image To Be Processed. Specify the SUM option of EXTRACT to add up the attribute values for all of the cells in the raster. The EXTRACT module sums (if we choose the SUM option) the values in all of the cells which underlie each value category in the Feature Definition Image when it is placed over the Image To Be Processed. In this case we have only one value category (zero) throughout our blank raster, the Feature Definition Image. Therefore the attribute values for the entire image are summed to produce a value of 81,378,400 liters for the entire study area for this precipitation event.

Preparatory Knowledge/Skills

Recommended:

1. Background Materials: What is GIS?
2. Unit 22: Merging Tabular Data with Spatial Data
3. Unit 34: Types of Overlay Operators
4. Unit 41: Using Boolean Search Techniques

Complementary:

1. Unit 19: Planning a Tabular Database
2. Unit 28: Editing Raster Data
3. Unit 43: Using Derivative Surface Operators
4. Unit 44: Using Hydrologic Models

Awareness

Learning Objectives:

1. Student can define basic vocabulary relating to Map Algebra.
2. Student can explain the differences between mathematical application of a scalar, a function, and a data layer.
3. Student can describe why Map Algebra operations are performed on the actual data values in a raster or in a database for a vector GIS.
4. If software will be used, students can describe the software steps used to conduct Map Algebra using a scalar, a data layer and a function.
5. Student can give an example of each of the three types of operation.
6. Student understands the special case when two data layers are multiplied and the second contains only zeroes and ones. This has the effect of "masking out", i.e., converting to zero, the areas in the first data layer which correspond to zero areas in the second, while leaving attributes in the areas corresponding to ones unchanged.

Vocabulary

Map Algebra

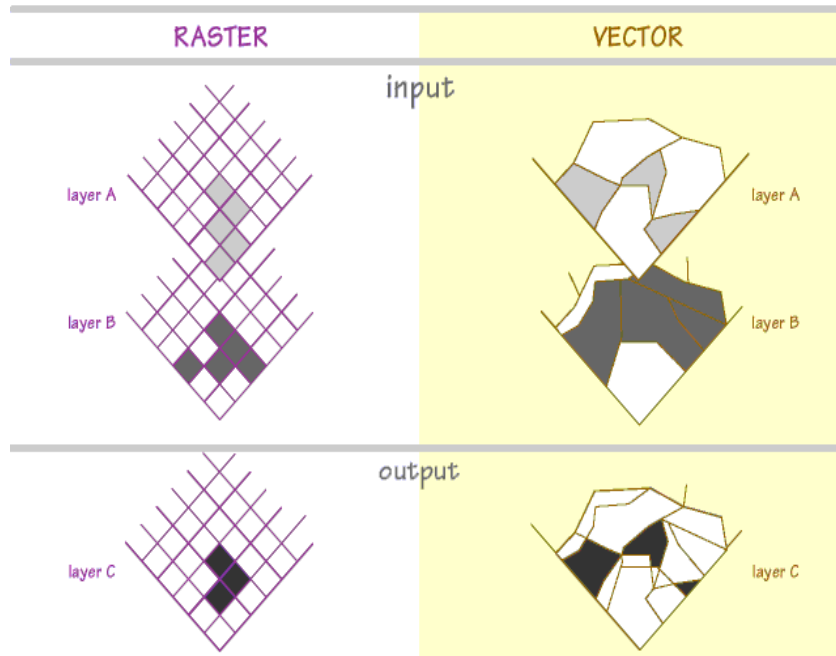
Scalar

Function

Overlay

Awareness Topics

- Map Algebra is used to transform or derive new data values from one or more existing data sets.
- The spatial extent of the data layers is not altered in Map Algebra.
- In a vector system, in which attributes are stored in a database, Map Algebra using a scalar or a function involves transformation of the attribute values in the database. These are commonly written to a new field in the database.
- In a vector system, Map Algebra in which a data layer is combined with another data layer can produce a new data layer in which derived features may have different sizes and or shapes than in either input layer. For example, a data layer in which polygons represent areas suitable for development, with attribute values ranging from 0 (unsuitable) to 1 (most suitable) could be multiplied by a data layer in which preference for development was also depicted as polygons with attribute values between 0 and 1. By multiplying these two data layers together, we can produce some composite probability of development image. It is important to note that we might find that our output polygons differed in size and shape from the polygons in either of our input images (Figure 5).

Figure 5: Combination of data layers produces new feature geometry


- In a raster system, sizes and extents of individual cells are not altered by application of a scalar, a function, or another data layer, but the sizes and shapes of features (collections of cells, generally with the same single value) depicted in the result may differ from the sizes and shapes of features in the input. For example, one could have a raster image in which each pixel value is the average slope for that pixel, and one could have another image where the attribute value of each pixel is the soil erodability of that pixel. If the two images are multiplied together one can derive a simple erosion risk map. In this resulting image, the features, i.e., the areas of high erosion risk, intermediate levels of risk, relatively lower risk, or whatever way the output values are categorized, will have shapes and sizes which differ from features (areas of constant erodability or areas of constant slope) in the input images.

- If software is being used, the student should know what Map Algebra operations are available, and how they can be invoked.

Competency

Learning Objectives:

1. Student will be able to invoke the necessary software options to transform data attributes by combination with a scalar, a function, and another data layer.
2. Student will demonstrate the capability to perform a sequence of Map Algebra steps to conduct an analysis described by a mathematical equation.
3. Students will be able to explain why combination of data layers is likely to produce a resulting data layer in which the sizes and shapes of features differ from the input layers, whereas when a scalar or function is applied to a data layer, the resulting features have the same geometry as the input layer.

Generic List of Tasks

1. Perform a scalar operation for unit conversion:

In a raster system, multiply the elevation values in feet in the input layer by 0.3048 to produce a new data layer containing elevation values in meters (or transform from meters to feet by multiplying by 3.28).

In a vector system, transform the line attributes for an elevation contour layer, by multiplying by 0.3048 to transform the line attributes from elevation units of feet to units of meters.

2. Apply a simple mathematical function to a data layer, e.g., square the attribute values in a raster or vector.

3. Combine two data layers to produce a result:

Raster: multiply urbanization factor map by travel time per cell

Vector: multiply property values by appreciation coefficient map; public properties have one value of appreciation, private properties have a different value.

4. Perform a sequence of Map Algebra steps to interpret an equation, e.g., raster: estimate precipitation given a temperature map (degrees f) and an elevation map (m) using an equation such as $\text{precip(mm)} = (\ln(\text{temp(deg K)}) * (\text{elev(m)}/250))$



Example Implementation: Estimating the Number of Pets in a Region to Evaluate Suitability for an Animal Hospital

Mastery

Learning Objectives:

1. Student will understand how to start with an algebraic expression, determine what data layers are needed, transform the data layers to proper units if necessary, and combine the data layers to conduct an analysis which interprets the algebraic expression.
2. Student will understand how layer multiplication by second layer containing only attributes of zero and one can be used to discriminate features of interest or to mask features to be omitted from further analysis.
3. If vector software is used, student will understand how combination of two data layers containing features whose geometry differs between the two layers can produce a new data layer whose features differ in shape from features contained in both input layers.

General Tasks (see Example Implementation)



Example Implementation: Determining Potential Wind Power

References:

Nelson, Vaughn, Earl H. Gilmore, Kenneth Starcher, 1994, Introduction to Wind Energy, Alternative Energy Institute, West Texas A & M University, Canyon, Texas

Rohatgi, Janardan S., Vaughn Nelson, 1994, Wind Characteristics: An Analysis for the Generation of Wind Power, Alternative Energy Institute, West Texas A & M University, Canyon, Texas

Follow-up Units

1. Unit 21: Using Spreadsheets
2. Unit 22: Merging Tabular Data with Spatial Data
3. Unit 31: Managing Database Files

4. Unit 33: Using Buffering Operators

5. Unit 34: Using Overlay Operators; Pre/Post-Overlay Tasks

Resources:

Idrisi Resource Center Salzburg: Idrisi Tutorial *[outdated link removed]*

Acknowledgements

The author wishes to thank Michele Fulk of Clark Labs and the Clark University Cartography and Information Graphics Service for the invaluable assistance they provided in the preparation of this unit. Michele Fulk offered expert advice and editorial assistance. The Clark University Cartography and Information Graphics Service created the concept graphics for the unit.

Created: May 14, 1997. Last updated: October 5, 1998.

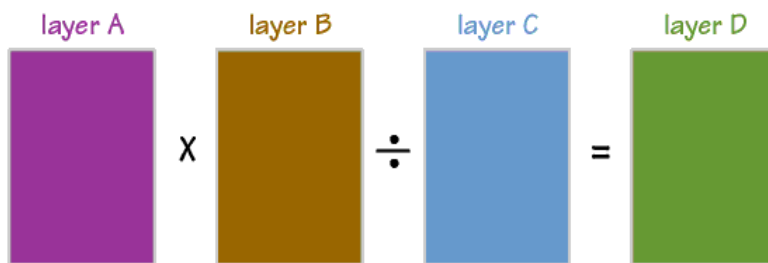
The NCGIA GIS Core Curriculum for Technical Programs

Map Algebra (from [Unit 42](#))

When data layers in a GIS are used as variables in equations, the term "Map Algebra" is applied (Figure 1). Modeling requires the ability to combine layers according to various mathematical equations. As another example, one might model the water runoff from an area based on data layers representing ground cover, precipitation, and slope. Each of these data layers would be used as "variables" in an equation relating these factors to the runoff from an area. For each corresponding pixel (in a raster system) or feature attribute record (in a vector system) the equation which relates the factors would be evaluated.

Figure 1: Map algebra

example: Multiply A by B and divide by C
($A \times B \div C = D$)



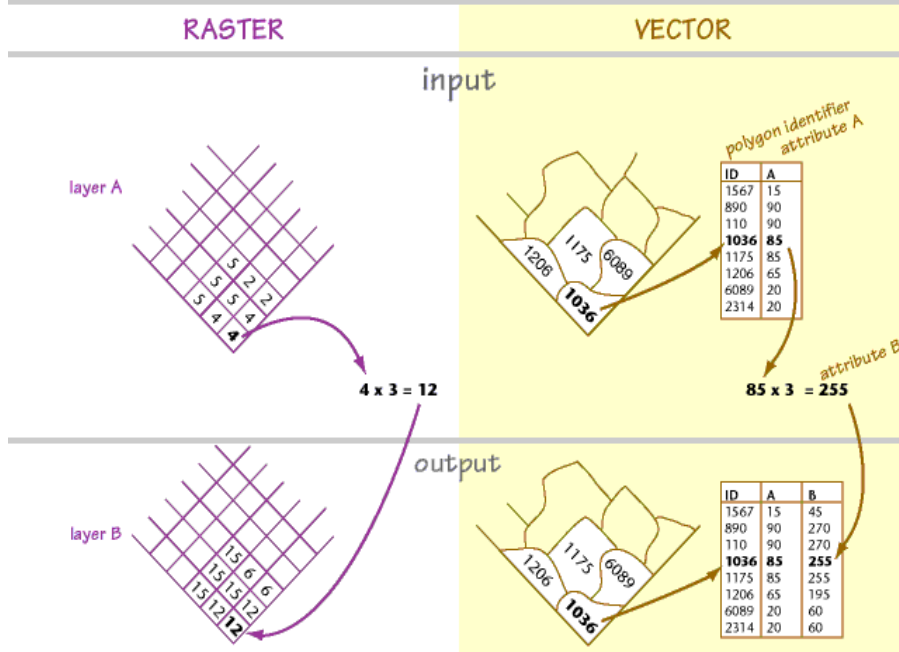
Map algebra can also be used to convert the units or range of units in which data is represented (e.g., changing an elevation data layer's units from feet to meters by multiplying the elevation value in each pixel or attribute record by 0.3048).

The Map Algebra tools will typically provide three different kinds of operations:

1. the ability to arithmetically modify attribute data values by a constant (i.e., scalar arithmetic, Figure 2) such as when we divide all of the distance values in a raster or attribute table by 1000 to convert distance attributes in meters to new distance values in units of kilometers;

Figure 2: Data layer with scalar

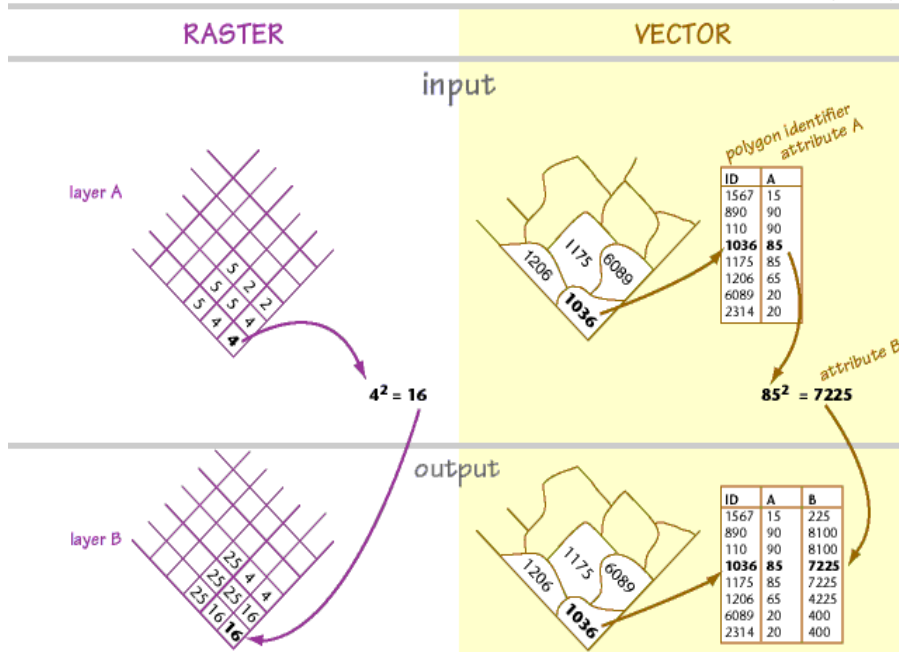
example: value x 3
(A x 3 = B)



2. the ability to mathematically transform attribute data values by a standard operation (e.g., the trigonometric functions, log transformations and so on, Figure 3) such as when we create a new data layer or attribute field by taking the cosine of all of the slope value attributes because this is required by our particular analysis;

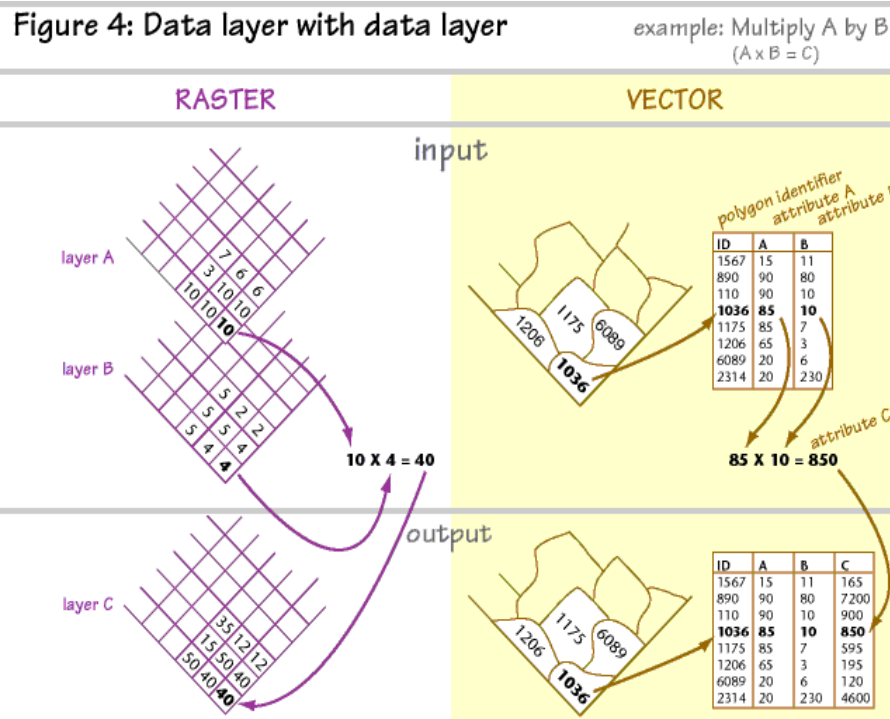
Figure 3: Data layer with function

example: value²
(A² = B)



3. the ability to mathematically combine (such as add, subtract, multiply, divide) different data layers to produce a result (Figure 4). Perhaps we have an equation that is the result of a regression analysis. To apply that equation to a set of variable layers, we would use map algebra. If we have an equation which relates the amount of rainfall and the elevation of an area to biomass productivity, we could use a digital elevation model and a rainfall map as variables in a map algebra equation to

derive an estimated biomass productivity data layer.



This third operation is simply another form of overlay - mathematical overlay, as opposed to the logical (Boolean) overlay of database query.

In a raster GIS, the input and output layers all have the same geometry, (e.g., number of rows/columns, spatial extent represented, pixel size/resolution) but the attribute values for some or all of the cells in the output raster are altered by the transformation applied.

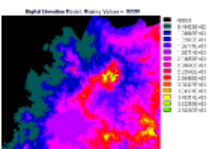
In a Vector GIS, transformation of data values by a scalar or function is typically performed directly in the attribute database or wherever the feature attributes are stored. Data layer combination generally involves creation of a new data layer in which the number and shapes of features differ from those in the input data layers. This is often accompanied by the addition or removal of features in the attribute data storage location.

Although it may be obvious, we might note for completeness that application of Map Algebra between images requires that the images used coincide with respect to projection, spatial extent, and, in addition for raster layers, pixel size.

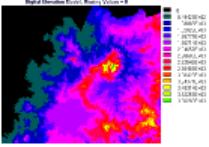
Three operations which use Map Algebra techniques, but which are discussed more fully in other units are:

1) recoding of data values. For example, if Missing Data is coded with a pixel value (raster system) or an attribute value (vector system) of -99999 and, for the purpose of an analysis, attributes for Missing Data should be recoded to zero (or some other value), then every attribute value of -99999 in the raster or attribute table can be replaced by zero (or the appropriate value). (Unit 40 - Using Reclassification Operators). This process is related to Map Algebra because attribute values in the output image are determined by applying a transformation to each input image value. In this case the transformation is to replace the input feature attribute value with another in the output layer whenever the input value satisfies a particular mathematical condition. In this case, when the input feature attribute value equals -99999, it is replaced by zero.

In this raster data layer, areas for which data is missing, i.e., the elevation is unknown, have an attribute value of -99999.



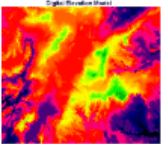
In this raster data layer, everywhere the attribute value was -99999 in the previous image, the attribute value is now zero.



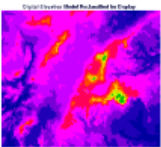
2) scaling of data values for output or display. For example, if one has a digital elevation data layer, rather than to designate a different color to each discrete elevation value, a data set might be easier to visually interpret if the range of elevation values is divided into equal portions and each portion assigned a different numeric value (or color for display). If elevations range from 100m to 300m, then all features or pixels in the elevation range from 100m to 140m could be designated with a new attribute value of 1 (which we will designate as dark blue for display), 140m to 180m as new attribute value of 2 (medium dark blue), 180m to 220m as value 3 (medium blue), and so forth. Sometimes this scaling process is performed by creating a new data layer containing the new values (raster system) or a new attribute field in the database (vector system). Alternatively, such value scaling can sometimes be performed in computer memory only for display or printing without altering or creating new attribute data. (Unit 48 - Designing Products for Printing).

The scaling of data values, or reclassification, as described here is a form of map algebra because a transformation of each feature's attribute value is based on the attribute value. In the example described above, if a feature in the input data layer has an elevation attribute value of 184 meters, then that feature will be reassigned a new attribute value of 3 in the output data layer.

In this digital elevation model, the raster cells have attribute values which range from 12 (meters above sea level) to 972 (meters).

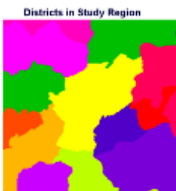


In this raster data layer, the elevation attribute values in the digital elevation model above have been reclassified into 10 elevation categories. All cells in the digital elevation model which had an attribute value of 12 (meters, the minimum value present) to 100 (meters) now have an attribute value of one. All cells with a value of 101 to 200 in the digital elevation model now have a value of two, and so forth.



3) masking out areas in a data set which should be excluded from analysis. In a GIS, if a first data layer is multiplied by a second data layer which contains only zeroes and ones, the result is a data layer in which attributes are unchanged in areas where the second layer attribute is one, and zero in areas where the second layer attribute is zero. This is a way to "mask" data, thereby limiting analysis to areas of interest. (Unit 41 - Using Boolean Search Techniques). Masking can be used to isolate areas based on attribute characteristics (e.g., we mask out all features except those with an attribute value of 17, 18, or 25) or based on spatial characteristics (e.g., we mask out all features outside a country boundary polygon to restrict our analysis to that country's features).

One use of masking is to isolate a feature of interest by changing pixels belonging to all other features to zero. In this example we are interested in the district which is colored dark blue near the center of the image.



We have reclassified the pixels in the feature of interest, in this case a municipal district, to have a value of one; all other pixels

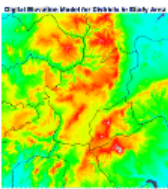
have been reclassified to have a value of zero. This is a mask image.



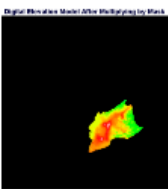
We can overlay-multiply our mask image by our original image to produce an image in which pixels in our feature of interest retain their original attribute values and all other pixels are changed to background (zero). Note that this image is displayed using a different palette; background values of zero are displayed using a gray color instead of black.



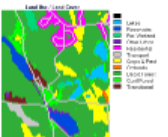
We can use the mask we just created to mask out the district of interest from a digital elevation model of the same area (assuming the image parameters are the same, e.g., number of rows/columns, projection, resolution, etc.) Here is a digital elevation model of the study area. For reference, a vector file showing the boundaries of the districts has been superimposed:



By multiplying our digital elevation model by the mask image, we create an image in which only the topography in the district of concern is shown.

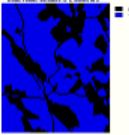


Masking of features can be applied, as we just saw, to isolate a single feature of interest. This technique can also be applied to isolate sets of features. This raster data layer contains eleven land use/land cover classes.

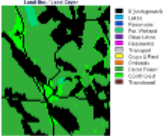


Suppose that for our analysis we only wish to be working with those pixels which represent forest areas. We can change all non-forest pixels to a background value of zero in two steps.

Step 1: Reclassify all pixels of Forested Wetland, Deciduous Forest, and Coniferous Forest to a new attribute value of one. At the same time, reclassify all other pixels to a new attribute value of zero. This procedure produces this image which contains only ones and zeroes:

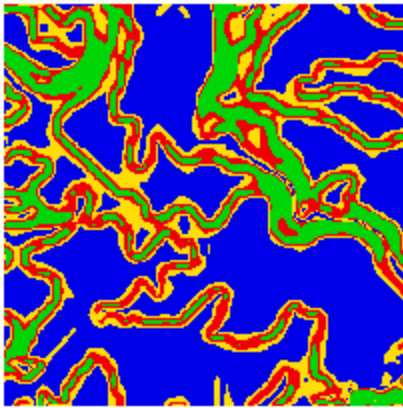


Step 2: Perform an overlay-multiply operation between the mask image created in Step 1 and the original land use/land cover data layer. For each pixel representing a forested area, the value of one in the mask image will be multiplied to leave the land use/land cover attribute unchanged. Wherever a non-forest pixel is present, it will be multiplied by a zero-attribute-value pixel in the mask image to produce an attribute value of zero in the corresponding cell in the raster layer created by this overlay step. As you can see, this produces a raster data layer in which the three forest classifications remain unchanged and all other land use categories have been converted to background (zero).

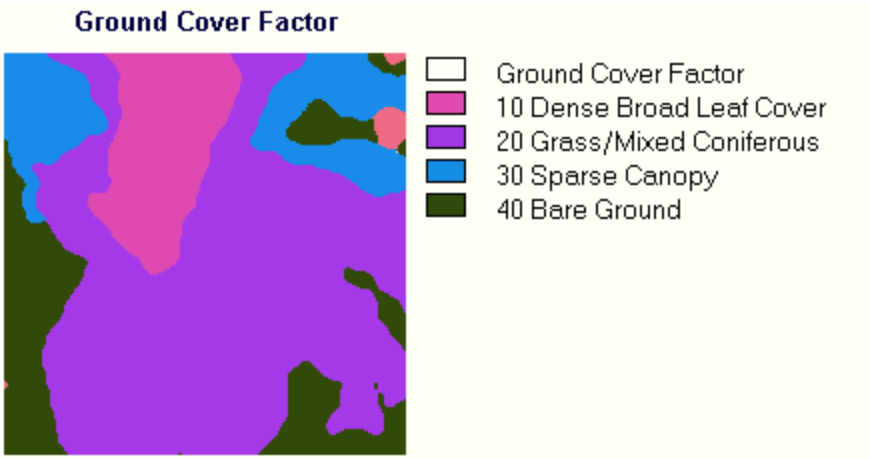


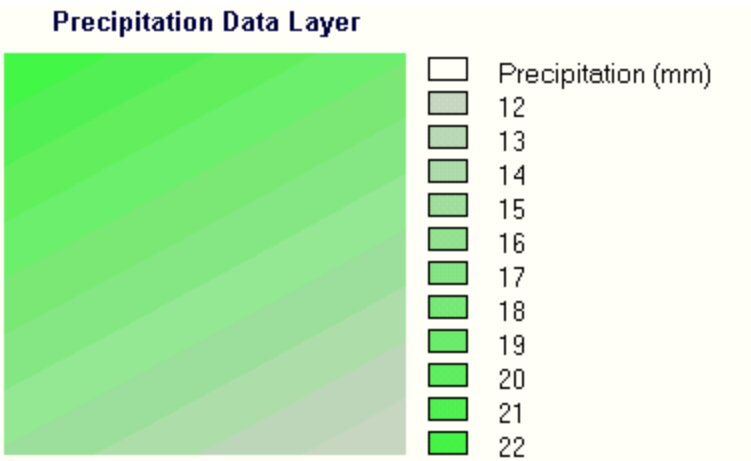
Created: May 14, 1997. Last updated: October 5, 1998.

Slope Factor Data Layer

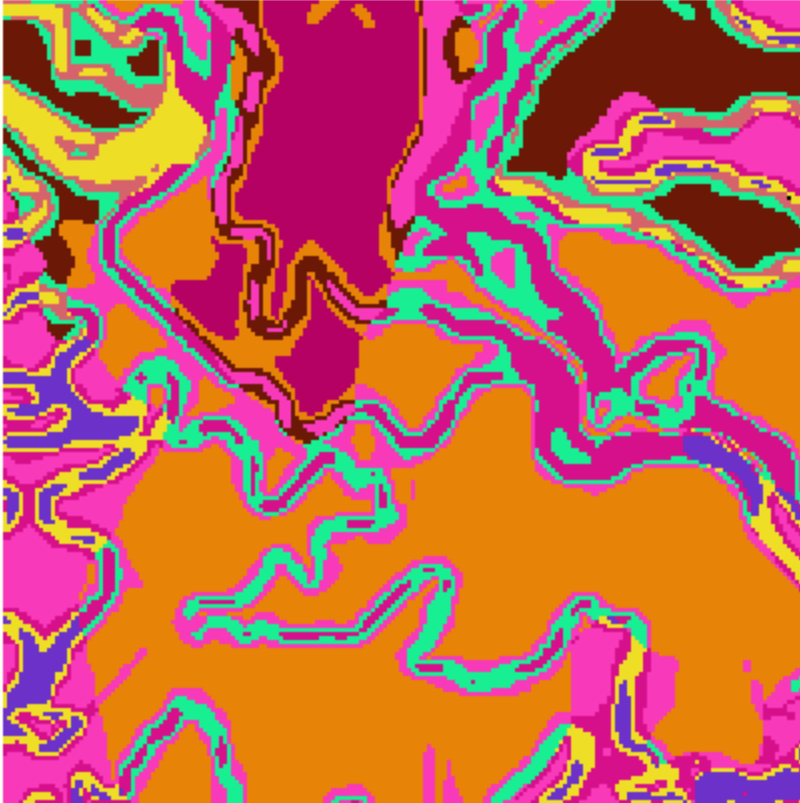


- Slope Factor Data Layer
- 1 (0 - 3 degrees)
- 2 (3 - 6 degrees)
- 3 (6 - 9 degrees)
- 4 (greater than 9 degrees)

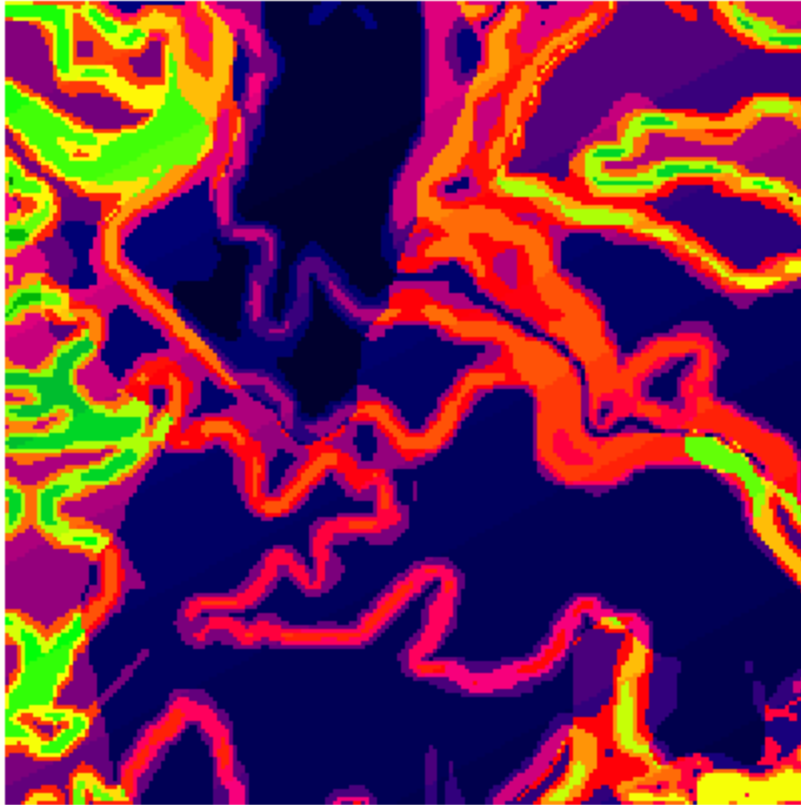




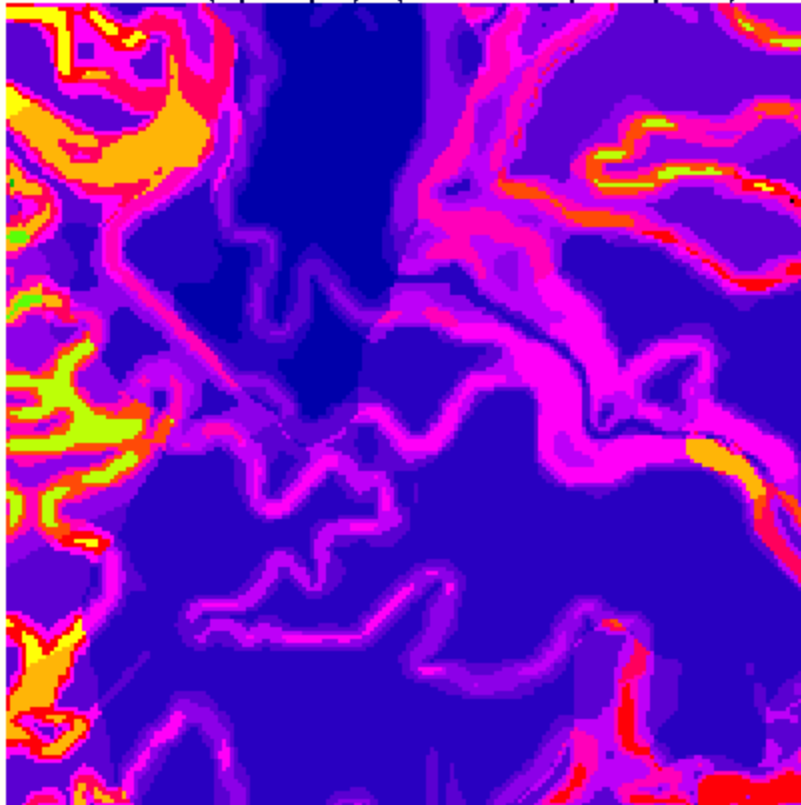
Product of Overlay-Multiply: Slope Type with Ground Cover Type



Product of 2nd Overlay-Mult: Prev. result with Precipitation



Result: Runoff (L per sq. M) = (Cover * Precip * SlopeFac)/160

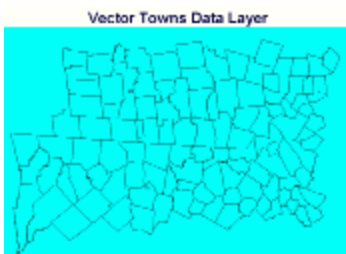


- Liters per Sq. M
- 0.0 - 1.5
- 1.5 - 3.0
- 3.0 - 4.5
- 4.5 - 6.0
- 6.0 - 7.5
- 7.5 - 9.0
- 9.0 - 10.5
- 10.5 - 12.0
- 12.0 - 13.5
- 13.5 - 15.0
- 15.0 - 16.5
- 16.5 - 18.0
- 18.0 - 19.5
- 19.5 - 21.0
- 21.0 - 22.5

Estimating the Number of Pets in a Region to Evaluate Suitability for an Animal Hospital

We have a proposed site for a new animal hospital and we wish to evaluate whether or not there is an adequate number of house pets (dogs, cats, gerbils, rabbits, etc.) to support an animal hospital at this location. Our research indicates that an area which contains 3,000 pets is likely to provide enough veterinary work to sustain an animal hospital. We have the following data to work with:

- 1) A vector polygon data layer representing the towns in the region.



- 2) The associated towns database contains attribute fields for total population (people) and area of each town (square kilometers).

TOWNS ID	TOWN POP	TOWN AREA
1	1474	75.38544
2	5094	103.7560
3	5203	103.5642
4	9055	90.18682
5	2410	52.15469
6	1519	67.40694
7	2221	17.79075
8	5705	53.71561
9	19550	75.35339
10	5515	40.02531
11	1294	80.4007
12	99567	90.42961
13	1213	62.07563
14	73105	54.85965

- 3) We have the coordinates of the nearest existing animal hospitals and of our proposed site. We obtained these coordinates by digitizing from a map.
- 4) We know that, on average for this region, there is one pet for every 17.25 people.
- 5) For the purpose of our analysis, we will assume that people will take their pets to the closest animal hospital when emergency treatment or routine veterinary activities, such as vaccinations, are necessary.

We will perform our analysis using Idrisi, a raster GIS. We start by creating a blank raster layer. All of the raster steps in the analysis will use raster layers which are identical to this one with respect to spatial extent, pixel size, number of rows and columns, projection, etc. We will use a Universal Transverse Mercator (UTM) grid referencing system and ground units of meters. Based on the extent of the study area 47.4 KM East/West and 40.2 KM North/South we decide on a raster cell size of 50 meters, which will provide sufficient detail for our analysis while maintaining manageable sizes of the data files as we conduct our analysis.

Next we rasterize the towns vector layer into a blank raster. Each pixel within a town takes on

an attribute value equal to the value of USER_ID for that town in our database.



This image shows the towns layer with vector town outlines superimposed as well as the locations of the existing and proposed animal hospital facilities. The attribute values in this raster layer range from 1 to 46. Every pixel in the town at the upper right corner of this image has a value of 10, which is the USER_ID of this town in our attribute table above. As can be seen in the table, this town, which extends beyond the edge of our study area, has a population of 5516 people and an area of 40.01531 square kilometers.

We can break down our analysis into two sub-questions:

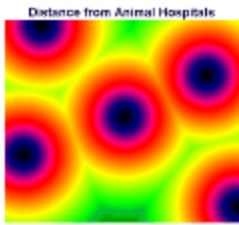
- 1) What portion of the study area satisfies the condition that the proposed animal hospital is the closest available hospital?
 - 2) How many pets are located in that portion of the study area? We know that in this area there is one pet for every 17.25 people, so we could estimate the number of pets if we could estimate the number of people who live in this area.
- 1) We will solve the first question first. While this part of the analysis demonstrates distance and allocation techniques more than map algebra techniques, it is included to clarify the context and interrelation between GIS procedures necessary for a "typical" GIS analysis.

We know the locations from which distance needs to be considered. They are the locations of our existing and proposed animal hospitals (target points). We create a vector point file using our animal hospital coordinates and giving each of the five location points a unique numeric ID. We will use ID's from one to five. (It would do just as well to use 6,7,8,9, and 10 as IDs; it doesn't matter as long as we keep track of which site has what ID.)

We create a blank raster which matches the characteristics of the raster layer for our study area. The blank raster (created in Idrisi using the INITIAL module) contains an attribute value of zero for every pixel. We now rasterize our five animal hospital locations (four existing and one proposed) into the blank raster. Now every pixel in the raster has an attribute value of zero EXCEPT for the five pixels at the locations of our animal hospitals (which are called Happy Pets, Fido's Friend, Proposed Animal Hospital Site, Kitten's Comfort, and Animal Haven, respectively). Those five pixels each have a value of 1, 2, 3, 4, or 5, corresponding to the IDs we chose for our animal hospital sites. The pixel at the Happy Pets location has an attribute value of 1. The pixel at the location of Fido's Friend has a pixel value of 2 and so forth.

We first create a distance image in which each pixel in the raster is given an attribute value which is the pixel's distance from the closest animal hospital. In Idrisi this is accomplished

using the DISTANCE module. By running DISTANCE on a raster containing the five animal hospital locations, the following raster data layer is produced:



Given a distance image such as this one as input, the Idrisi module ALLOCATE will allocate each pixel in the input to the nearest target point. ALLOCATE uses the blank raster containing only our animal hospital locations (the target points) with unique IDs to determine what features to allocate to. ALLOCATE produces an image like this (the target points and labels have been superimposed for reference):

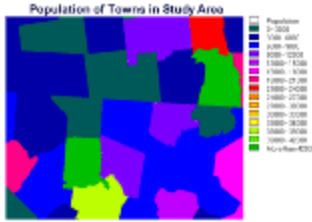


At this point in our procedure we have the answer to our first question. The blue area in the allocation data layer above is the area for which our proposed facility will be the closest animal hospital.

Now we can proceed to answer the second question. How many people live in the area which is colored blue in the allocation image? Now we undertake some true map algebra procedures.

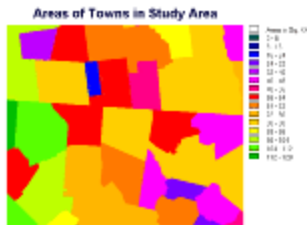
Our town database has attribute fields for population and area of each town. With this information to work with, we can determine the average population densities by town across the study area. From this we can determine the total number of people, and, by our known people-to-pets ratio of 17.25, the number of pets who live in the area which will supply customers for our proposed animal hospital (this is the blue area on the allocation image above).

We can proceed with the analysis this way. Our database, or attribute table is linked to our data layer based on the USER_ID field. We can therefore ASSIGN the population values (TOWN_POP) from the database to the towns raster layer. This produces an image in which each pixel has value equal to the whole town's population; every pixel in the town has that value. The resulting data layer looks like this:

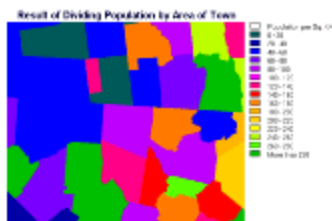


In this population data layer, for example, every pixel in the dark blue area at the upper right corner of the image has the value 5516, which is the total number of people that live in that town. It is not important that only a portion of the town is represented; we still assign to every pixel in the portion of the town within our study area the population of the entire town. We will account for this fact in an upcoming step.

Next, we ASSIGN the values from the TOWN_AREA field of the attribute table to create a new raster data layer in which the attribute value of each pixel is equal to the area of the town. For example, every pixel in the (pinkish purple) town in the upper right corner of the raster has an attribute value of 40.01531, which is the area of that town expressed in square kilometers. Again, each pixel is assigned the value of the entire area of the town, even though the whole town is not contained within our study area.



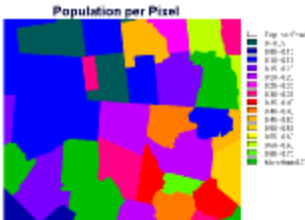
We next create a raster layer which shows the number of people per square kilometer for each town. This is done by performing an OVERLAY-DIVIDE operation. We divide the population per town data layer described above by the area per town layer we just created. This produces the following data layer.



We divided people/town by square kilometers/town to produce a data layer with attribute values of people per square kilometer. In this data layer, every pixel in the rose-colored town at the upper right corner has an attribute value of 137.8472 people per square kilometer (which is also the result of dividing the population and area attributes directly in our attribute table. Recall that this is the town with USER_ID = 10 in the attribute table above.)

From this we just produced a population density data layer which has attribute values in units

of people per square kilometer. For the next step in our analysis, we want to convert that to people per pixel. We have 50M by 50M cells and there are 400 such cells in a square kilometer. If we divide (the attribute values in) the population density data layer in units of people/square kilometer by 400 cells/square kilometer, we will produce a raster layer in units of people per pixel:



In this case every pixel in the rose-colored town at the upper right corner of the image has a value of 0.3446, representing the number of people living in that pixel.

The two steps we just completed, producing a population per area by town map and producing a population per pixel map were based on whole-town areas and populations. The attribute values within each town were uniform. If we were to enlarge our study area to include the entire area of a town that is only partially included. We would find that the pixels in the entire town would be the same as those pixel values in the portion of the image in our present analysis. It is in the next step where we take into account that we are only considering the portion of each town that is contained within the study area.

Our analysis is almost completed. All we have to do now is add up the number of people per pixel (depicted in the data layer we just created) for all of the pixels in the customer supply area of our proposed animal hospital. The allocation raster we created above indicates exactly which pixels those are. We use the EXTRACT module in Idrisi to add the pixel values in the people-per-pixel image for each of our animal hospital supply areas. We tell EXTRACT to use the allocation image prepared above to define the areas over which our mathematical analysis, in this case summation, will take place. This is called the Feature Definition Image and it shows 5 areas; our study area is divided into the 5 regions which supply the respective existing/proposed animal hospitals. The Image To Be Processed is the population-per-pixel image. We wish to add up all of the population-per-pixel values within each of our five areas to estimate the total number of people in the supply area for each animal hospital. If we run EXTRACT and specify that we wish to have a table of the results, this is what is produced:

Category	Total	Legend
1	10260.00242	Happy Pets
2	32768.33828	Fido's Friends
3	83797.864328	Properpet's Place
4	74526.562922	Kitties' n' Cuddles
5	42550.494517	Happy Paws

Based on the assumptions we made (e.g., population is distributed uniformly within the area of each town), we estimate that there are 81,797 people for whom the proposed animal hospital site will be the closest animal hospital facility. In fact, the analysis suggests that there are probably more people, because the blue area allocated to our proposed site appears to extend beyond the southern limit of our study area (assuming north is toward the top in these examples). Based on our demographic statistic for our study area that there tends to be one pet for every 17.25 people, we estimate that the total number of pets for which the proposed

facility will be the closest animal hospital is $81797/17.25$, which equals 4,741 pets. This exceeds the 3,000-pet minimum that we needed to justify a new animal hospital at the proposed site.

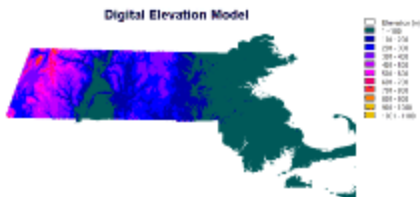
There are many ways we could improve the accuracy of this analysis. For example, if we had information about how population is distributed within town boundaries, perhaps we could make a more accurate population density map. We assumed that people will take their pet to the animal hospital that is physically the closest facility. In fact, people may take their pet to the facility that they can reach most quickly. If we had a streets and roads data layer we could use cost-distance modelling or network analysis to create a travel time map. We could then allocate the population within our study region based on the shortest travel time rather than the shortest distance.

Determining Potential Wind Power

This example demonstrates the use of GIS to assess the potential use of wind power. The example is adapted from a portion of a Clark University Master's Degree project completed by Margaret Robertson of the Idrisi Project/Clark Labs. The use of her materials for this example is gratefully acknowledged.

The use of wind-powered generators to produce electricity is a matter of great interest as people become more sensitive to potential environmental consequences of nuclear power and the use of fossil fuels. If wind energy is to be considered as a potential source of electricity in a region, it is necessary to evaluate whether and where there is sufficient wind to justify electricity generation from wind. It is important to estimate the amount of power in the wind at a potential site. If we know (or can estimate) the wind speed at the height where the blades of a windmill are located and the air density at that height, we can estimate the amount of available power in the wind.

The study area consists of most of the state of Massachusetts in the U.S. We start by estimating the air density at the height of the windmill blades (assumed to be 50 meters). We will derive the air density by using a digital elevation model of the study area.

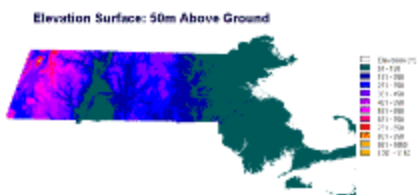


Air density is given by the following equation:

$$D = 01.226 - (1.194 * 10^{-4})z$$

where D is the air density in kg/cubic meter, z equals the elevation in meters and 1.226 kg/m³ corresponds to the standard air density at sea level (z = 0) at a temperature of 15 degrees C (Nelson, et al, 1994).

As a scalar map algebra step, we add 50 meters to the elevations in our digital elevation model. This produces the following image:



This elevation surface is used as "z" in the air density equation above. The equation can be evaluated in one step using Idrisi's Image Calculator, which allows us to type in the equation directly. Or if necessary or desired, the equation can be evaluated in two steps:

First step:

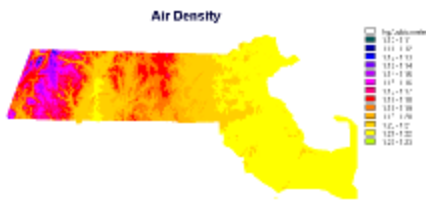
$$0.0001194 * (\text{50-meter surface}) = (\text{intermediate result})$$

where * indicates multiplication and parentheses indicate that data layers are used as variables.

Second step:

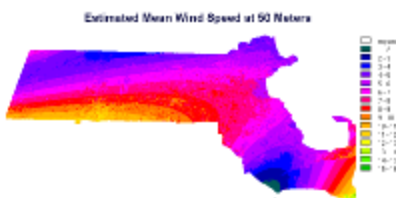
$$(\text{density at 50 meters}) = 1.226 - (\text{intermediate result})$$

This density image is the result:



We also have a wind speed surface for a 50-meter height above ground for the study area. It is beyond the scope of this example to discuss the derivation of the wind surface in detail.

Essentially, long-term wind measurements at many known locations were analyzed and interpolated to derive a wind surface near the ground, where the measurements were taken. Empirically calibrated formulae allow the extrapolation of near-surface winds to elevated locations. Further information can be found in the materials referenced below. This is the 50-meter wind surface, which depicts the average wind speed at an elevation of 50 meters:



Now that we know the density of air at 50 meters above ground level and the average wind speed at 50 meters above ground level, we can use the following equation (from Nelson, et al, 1994) to estimate the power per cross-sectional area in the wind:

$$P/A = 0.5D(v \text{ exp } 3)$$

where

P is power in watts

A is cross sectional area (through which wind is passing) in square meters

D = density of air in kilograms/cubic meter

v = wind speed in meters/second

$v \text{ exp } 3$ = the cube of v

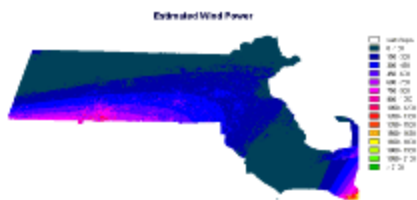
This can be evaluated using map algebra techniques as:

(answer, in watts per square meter) = $(0.5 * (\text{density50m}) * ((\text{wind50}) \text{ exp } 3))$

or:

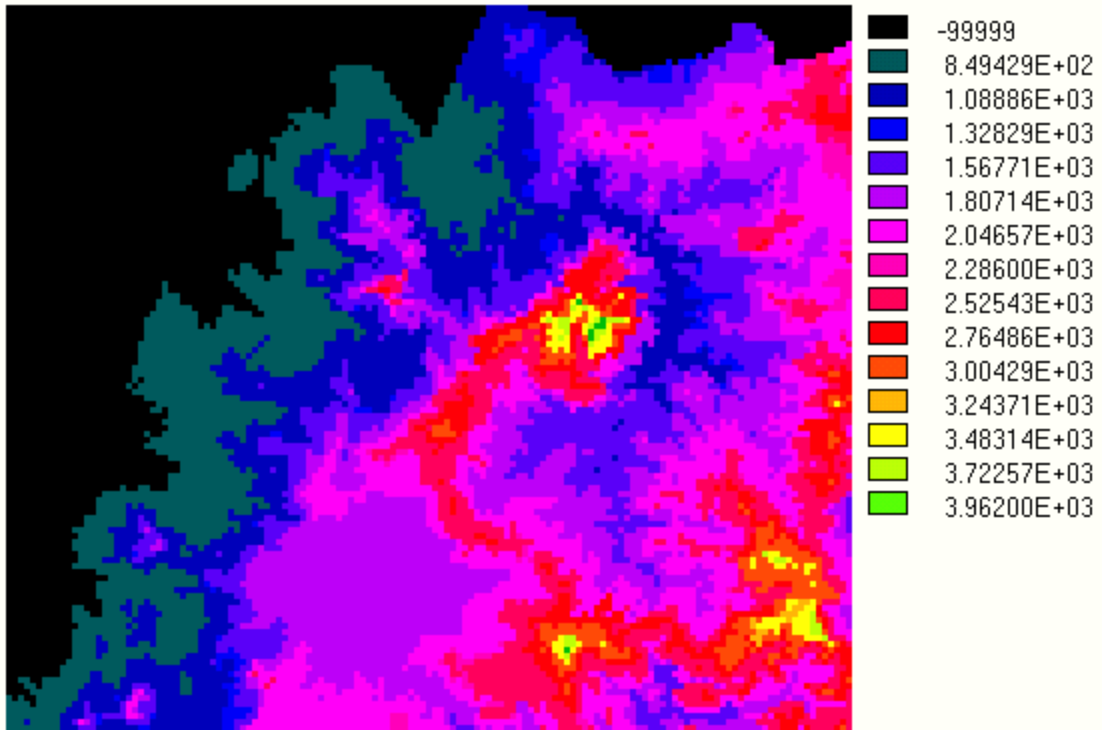
(answer, in watts per square meter) = $(0.5 * (\text{density50m}) * (\text{wind50}) * (\text{wind50}) * (\text{wind50}))$

In Idrisi this can be performed in one step using Image Calculator. Depending on which GIS software is used, it might be necessary to evaluate the equation as a series of substeps. In any case the result we obtain is this:

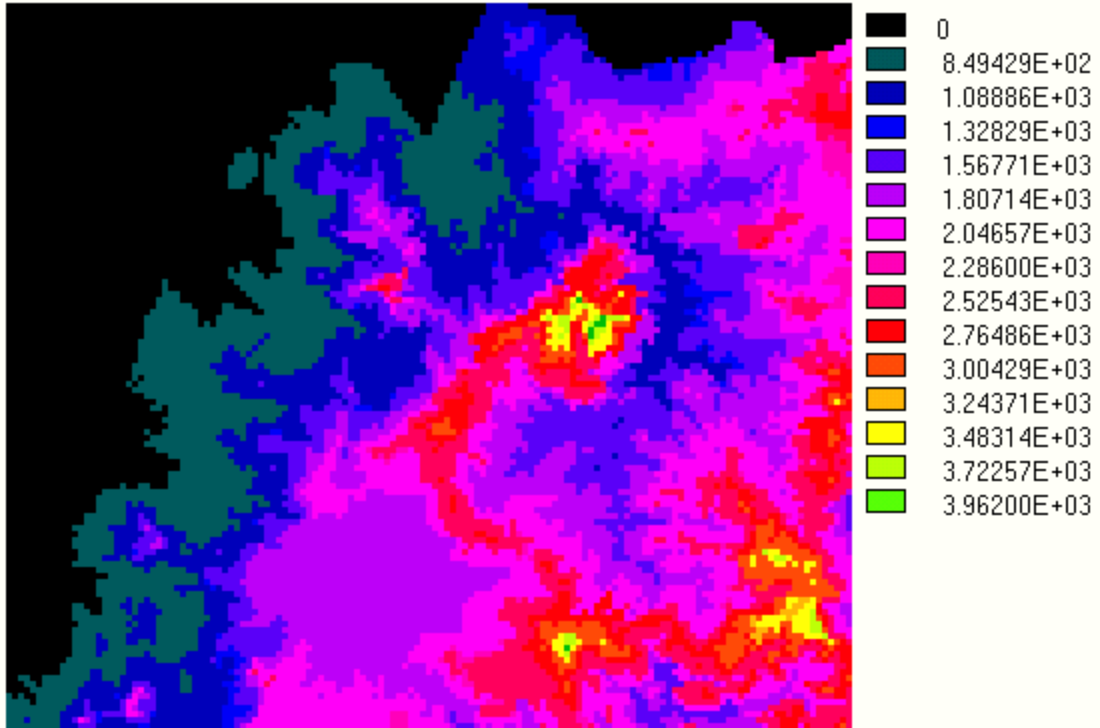


Rohatgi and Nelson (1994) suggest that generation of electrical power may be viable in areas where 50-meter wind power values exceed approximately 300 watts/sq m. Based on our analysis, it appears that large areas of the State of Massachusetts could be considered for wind power development. The wind power map above suggests parts of the state (e.g., the southeastern corner of the state and along the southern border in the western half of the state) where pilot tests might be conducted.

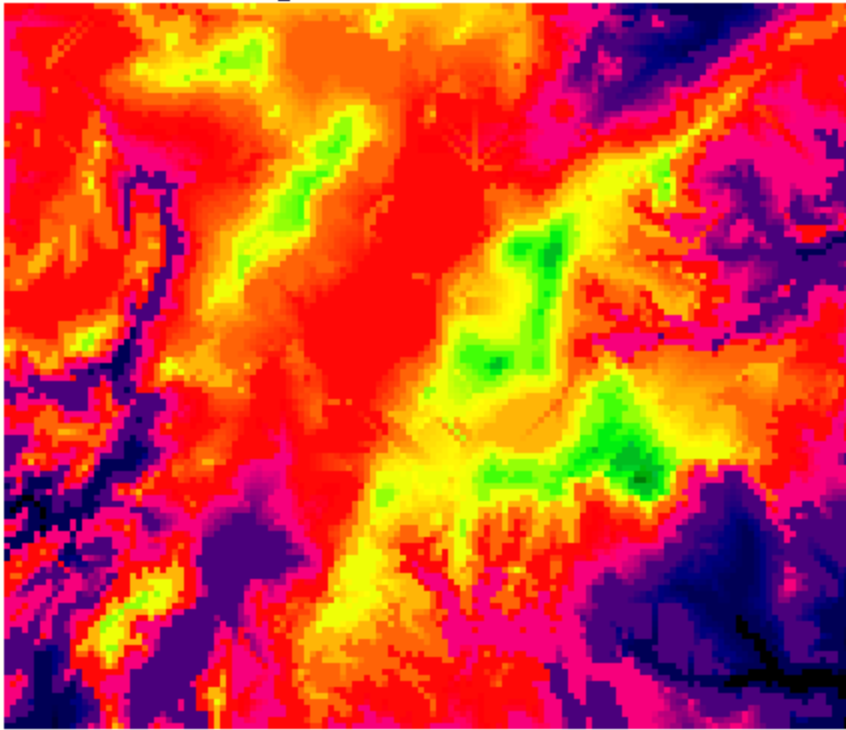
Digital Elevation Model, Missing Values = -99999



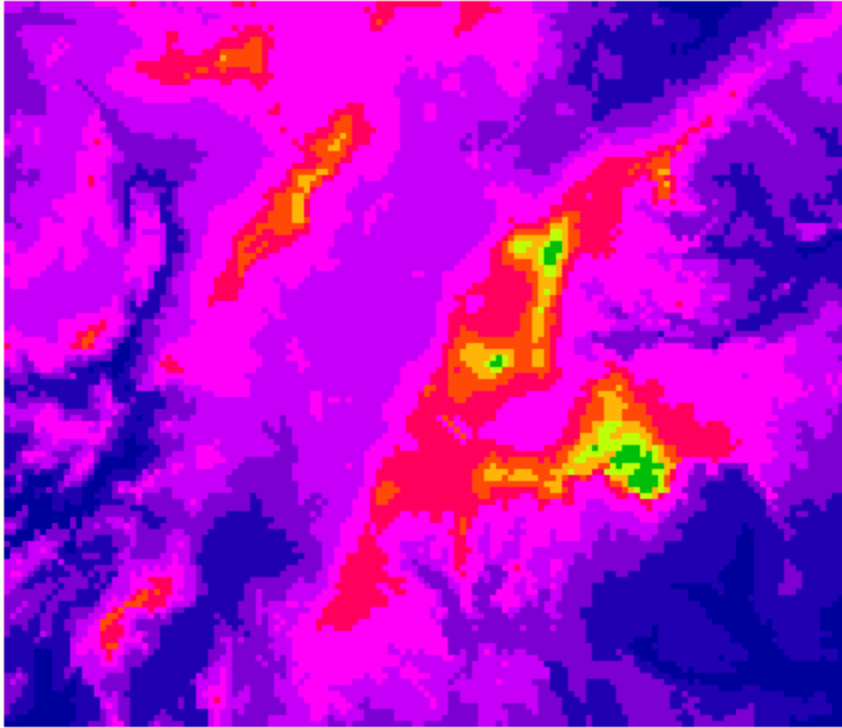
Digital Elevation Model, Missing Values = 0



Digital Elevation Model



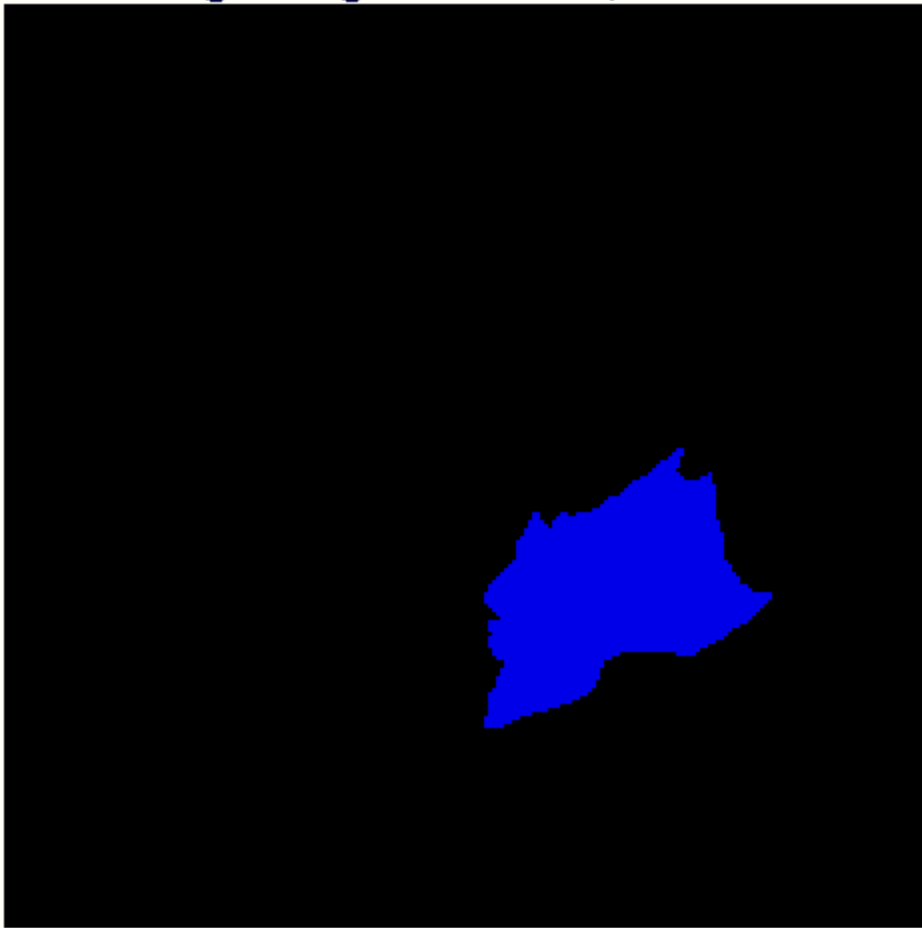
Digital Elevation Model Reclassified for Display



Districts in Study Region

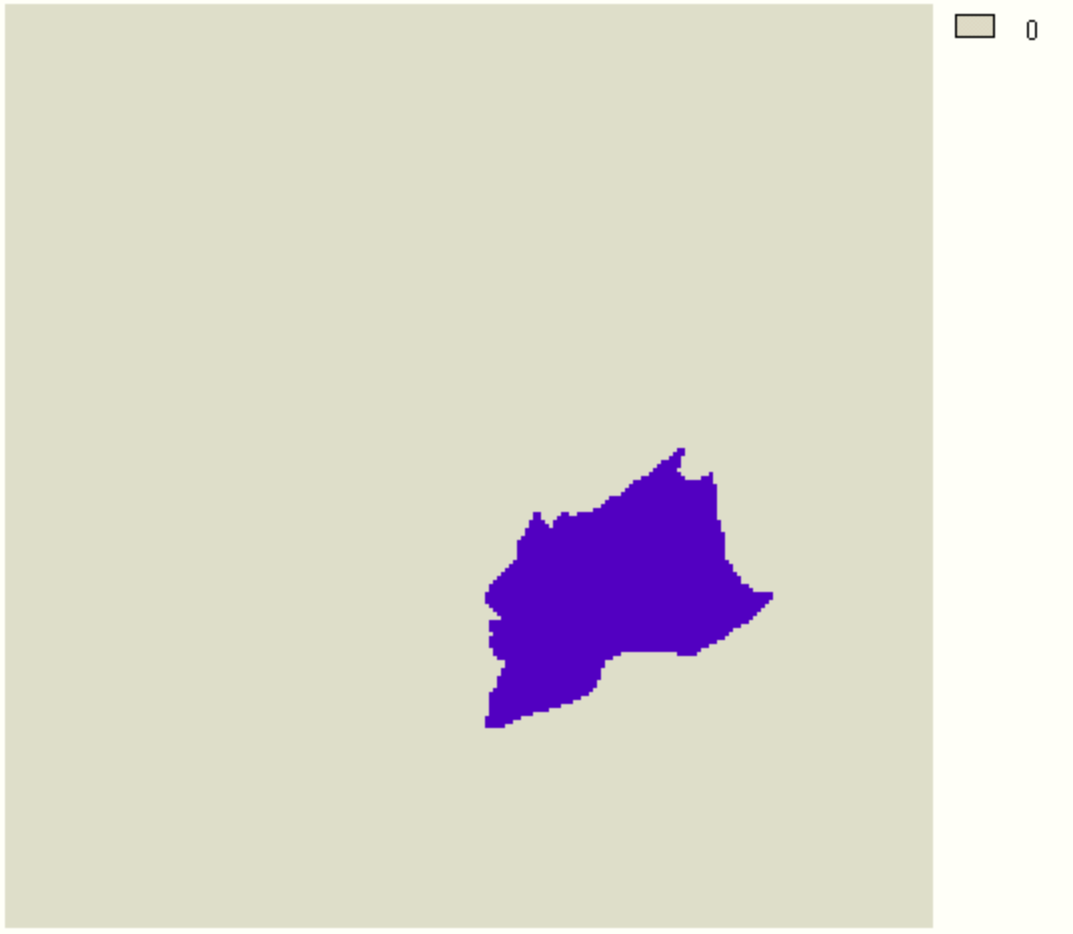


Mask Image: Target Pixels = 1; All Others = 0

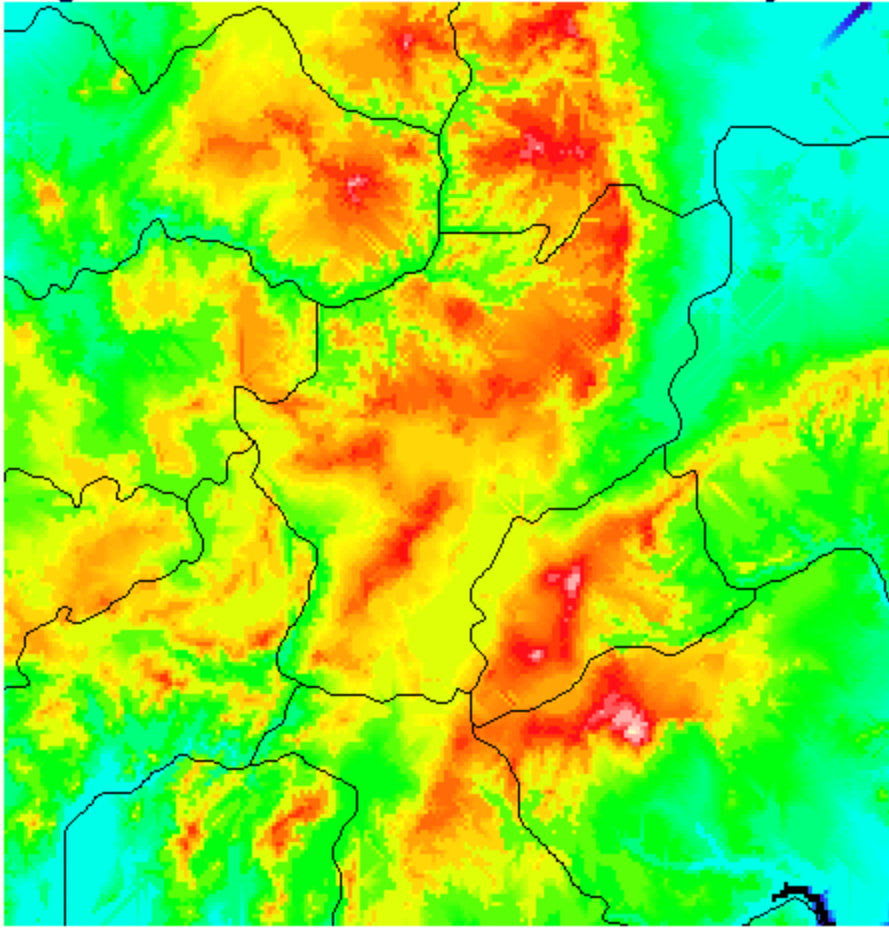


0
1

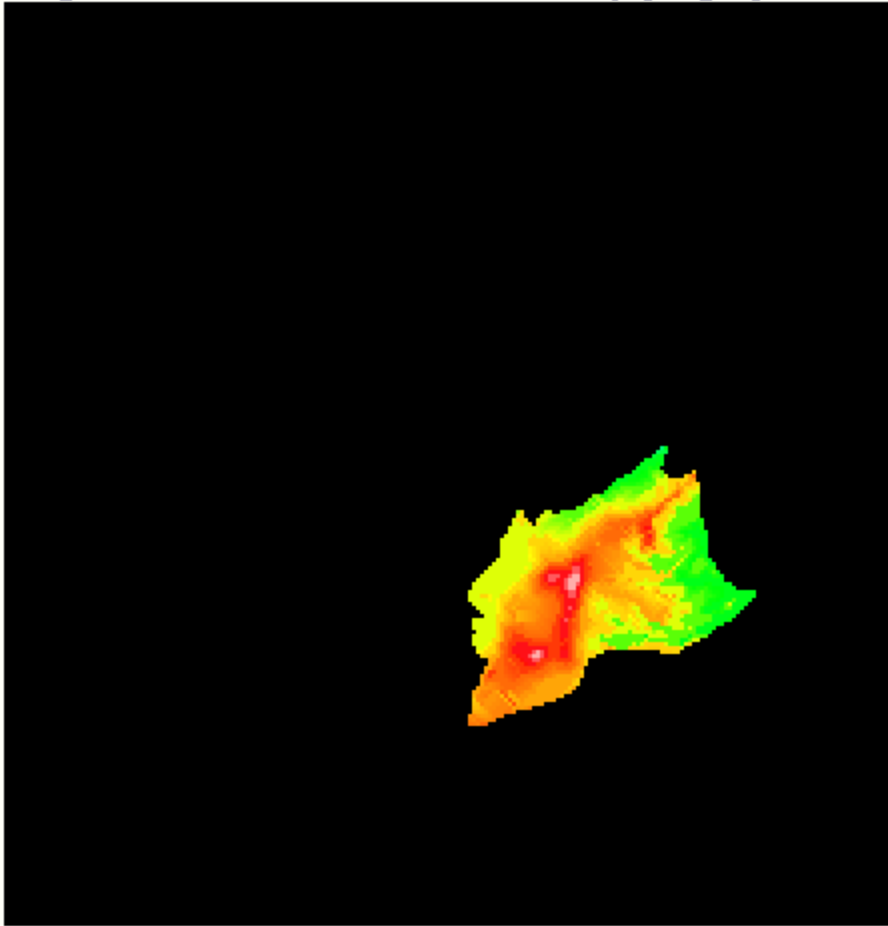
Result of Multiplying Original Image by Mask

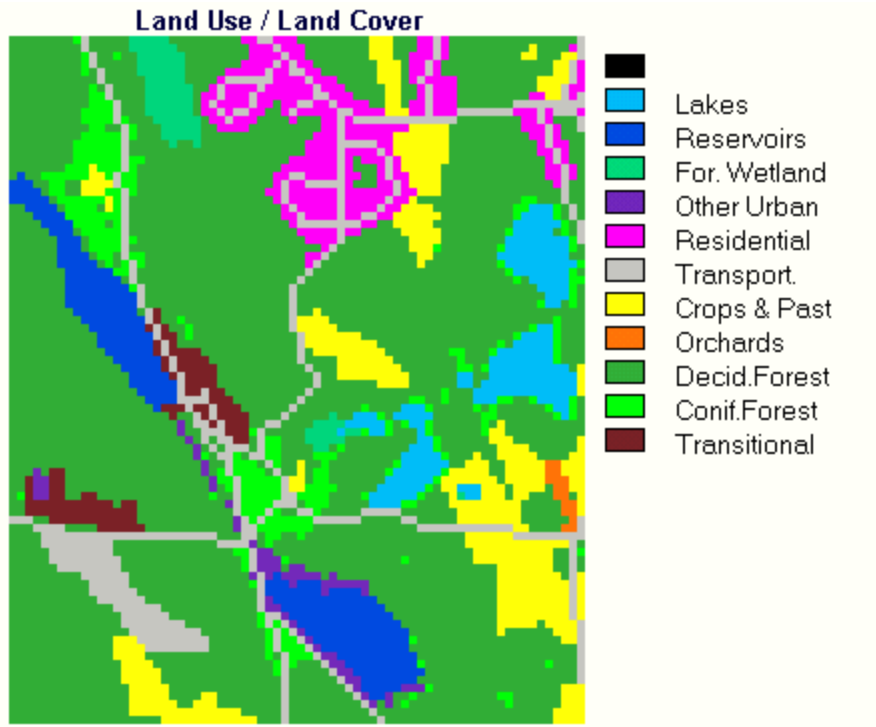


Digital Elevation Model for Districts in Study Area

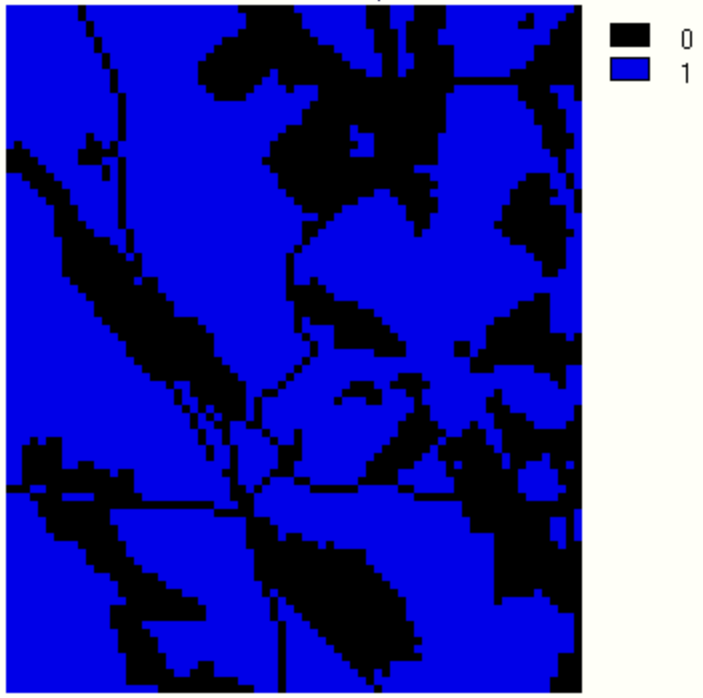


Digital Elevation Model After Multiplying by Mask

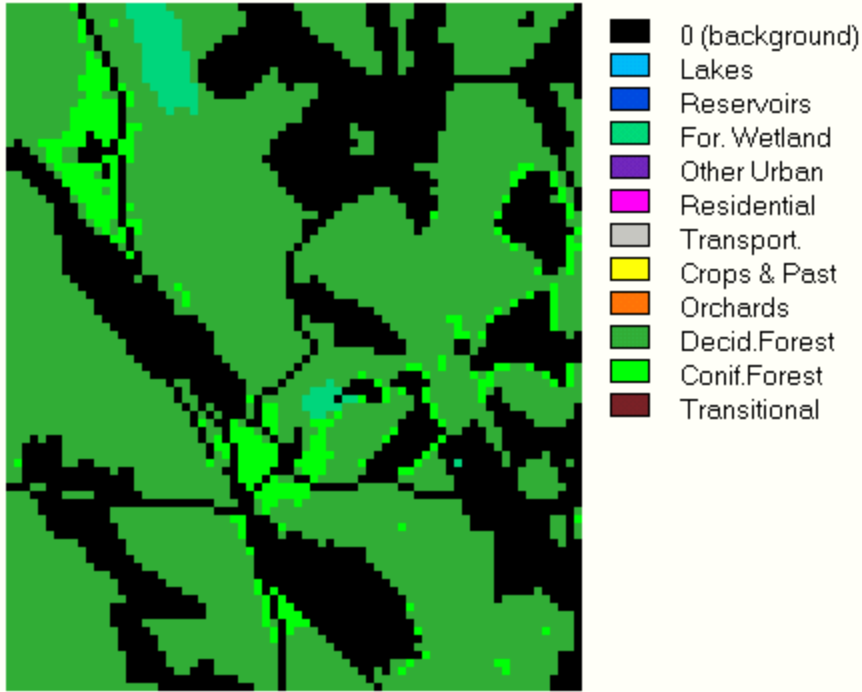




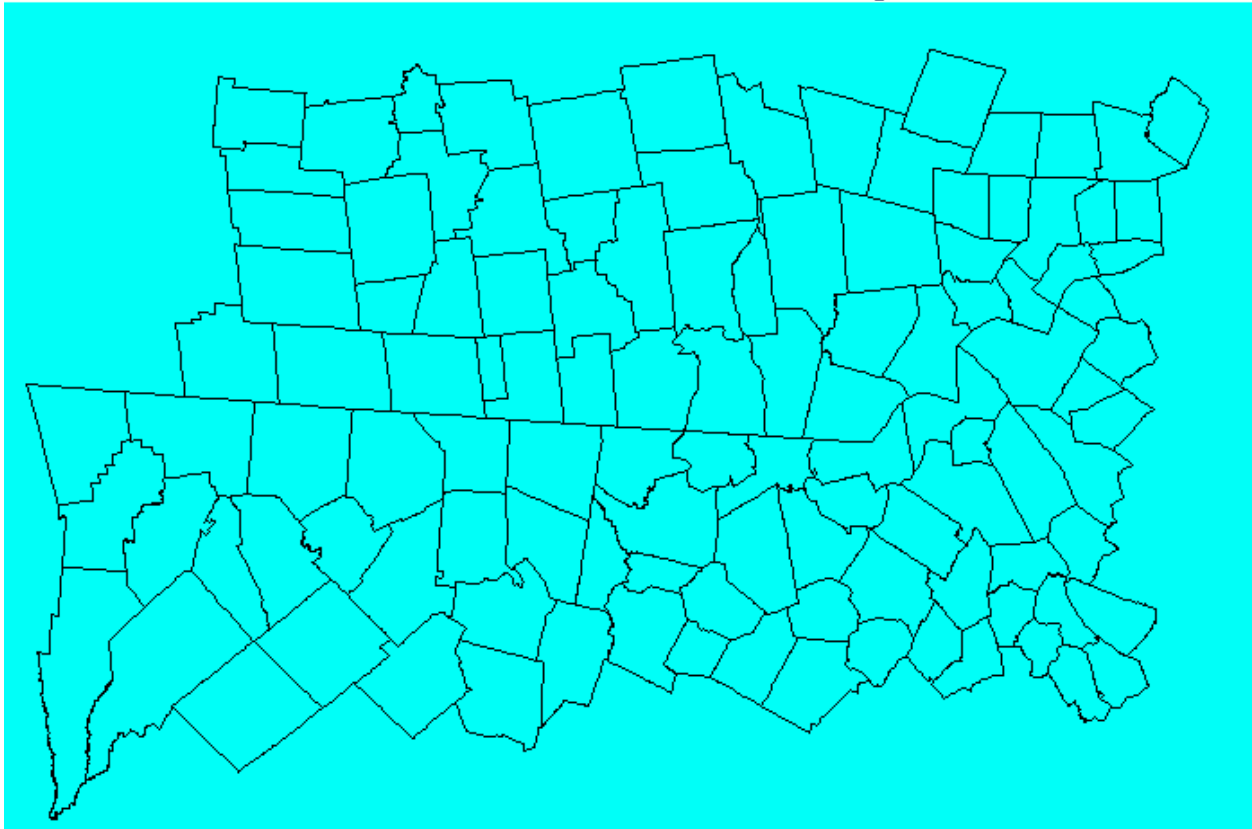
Mask: Forest reclassified to 1, others to 0



Land Use / Land Cover

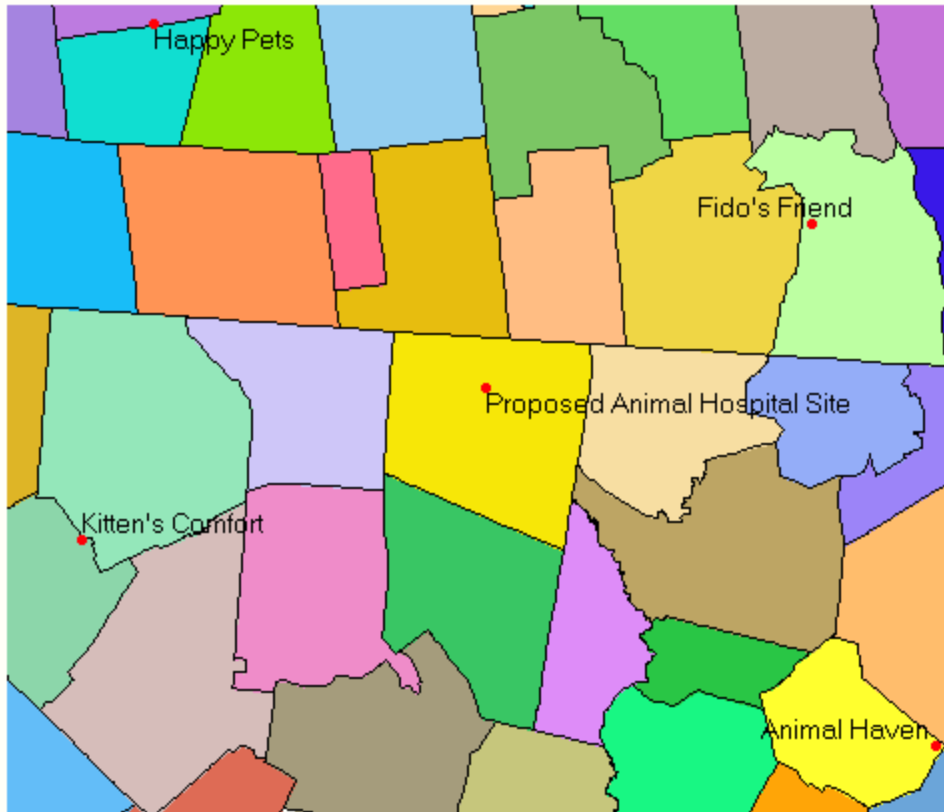


Vector Towns Data Layer

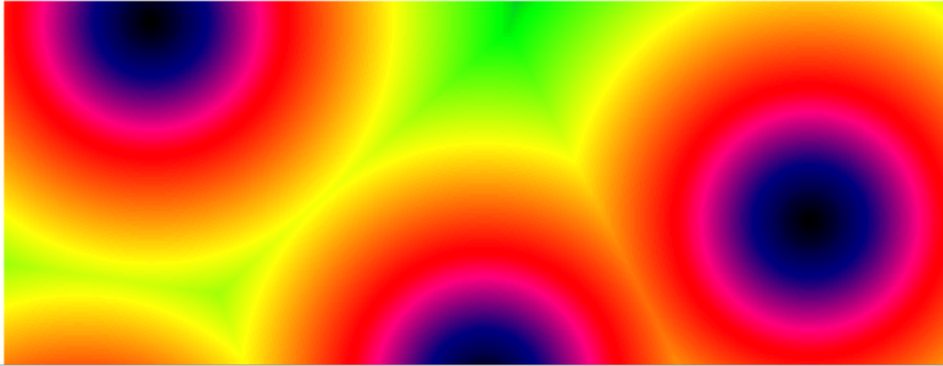


USER_ID	TOWN_POP	TOWN_AREA
1	1474	75.38544
2	5094	103.7568
3	5203	103.5642
4	9068	90.18682
5	2410	52.15469
6	1519	67.40694
7	2231	17.79875
8	5705	83.71561
9	19530	75.35339
10	5516	40.01531
11	1294	80.4007
12	99567	90.42961
13	1213	62.07863
14	22185	86.59508

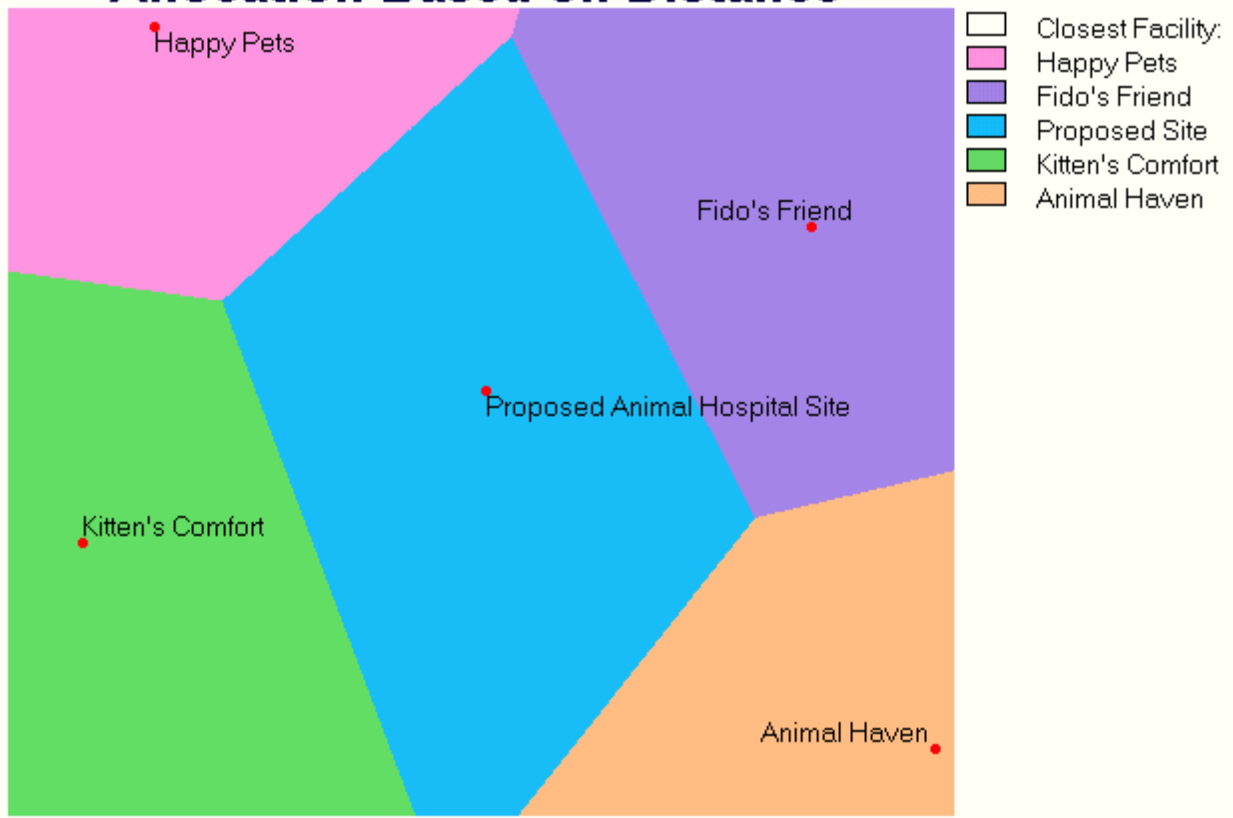
Towns in Study Area



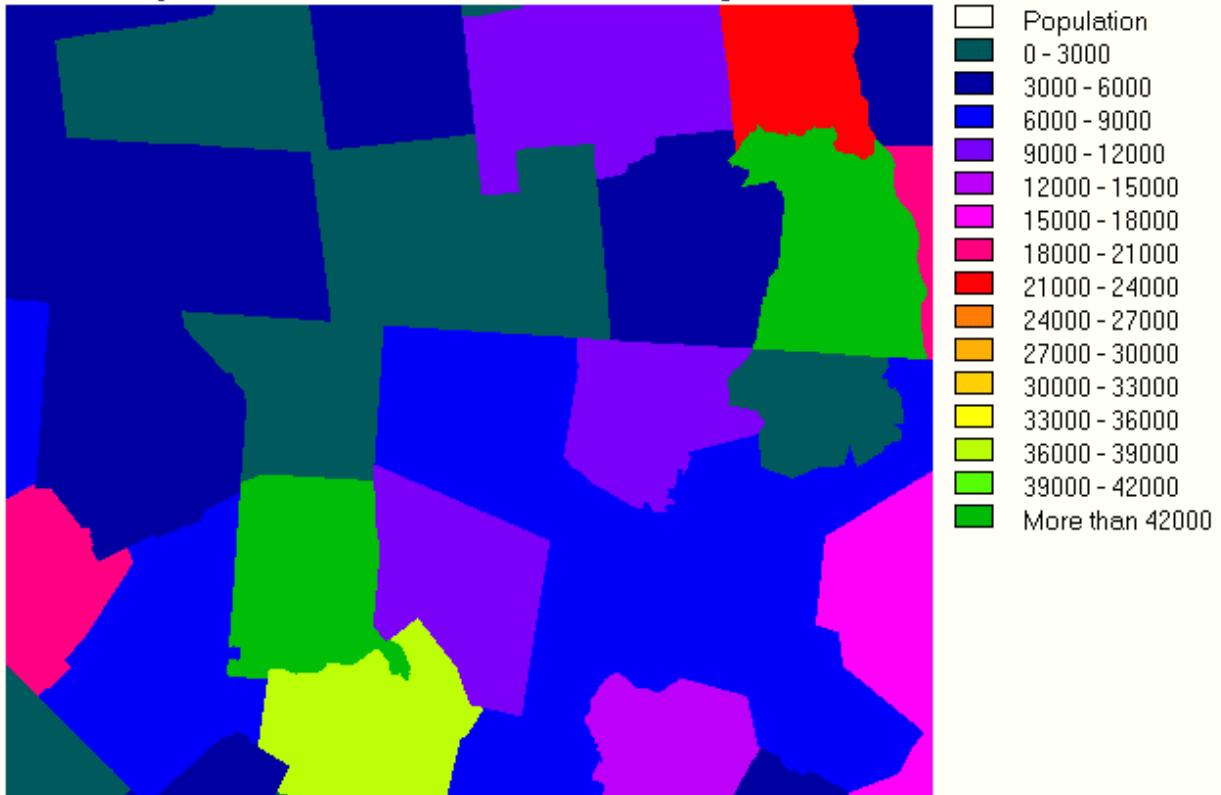
Distance from Animal Hospitals



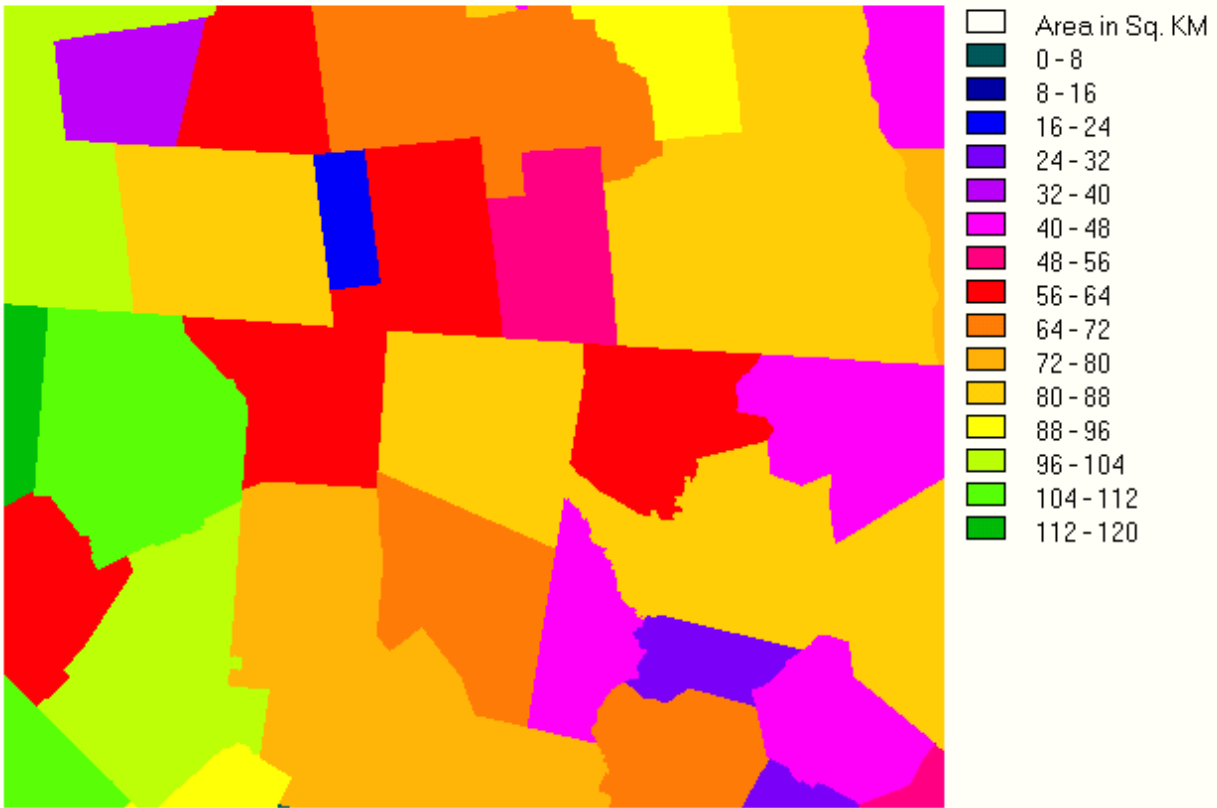
Allocation Based on Distance



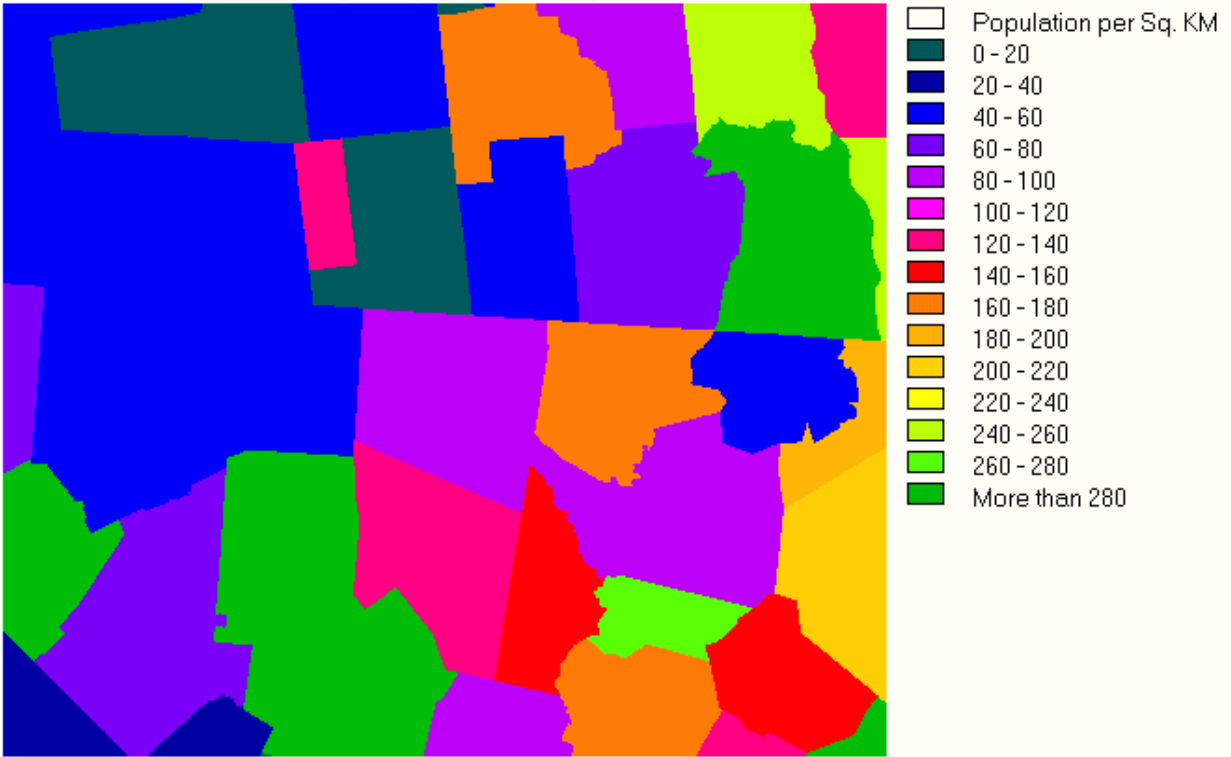
Population of Towns in Study Area



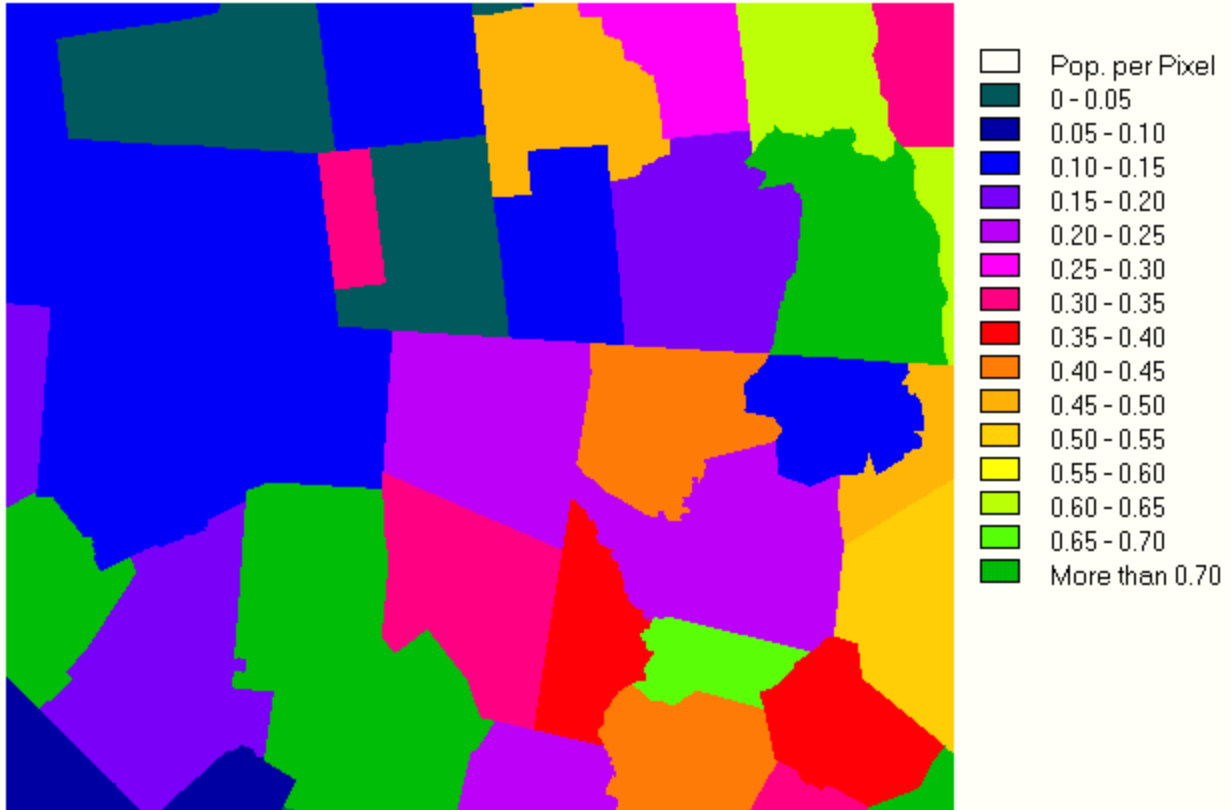
Areas of Towns in Study Area



Result of Dividing Population by Area of Town



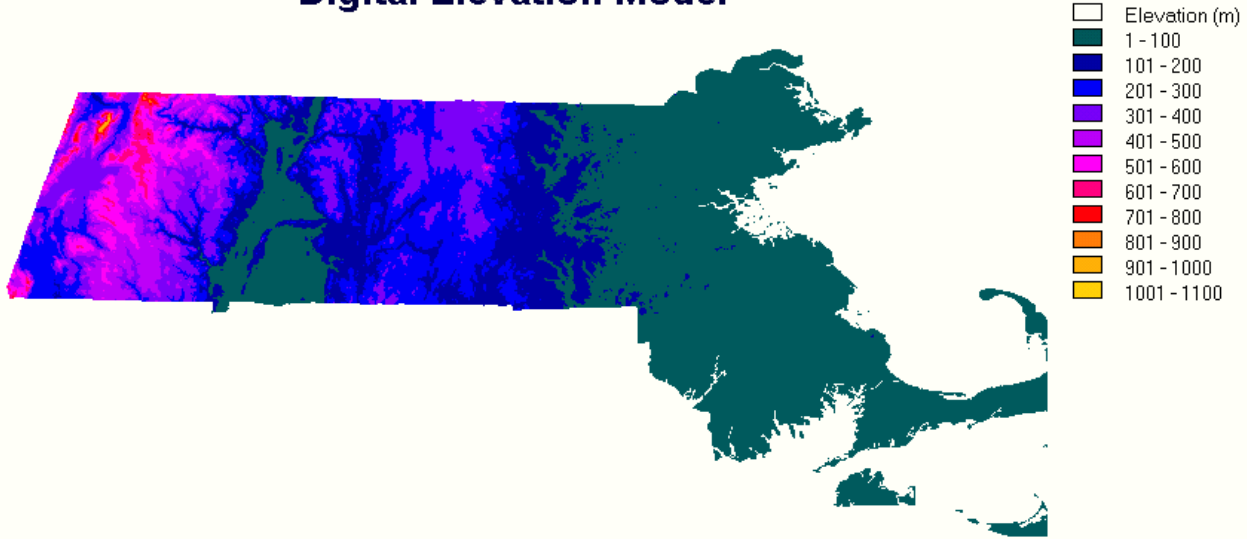
Population per Pixel



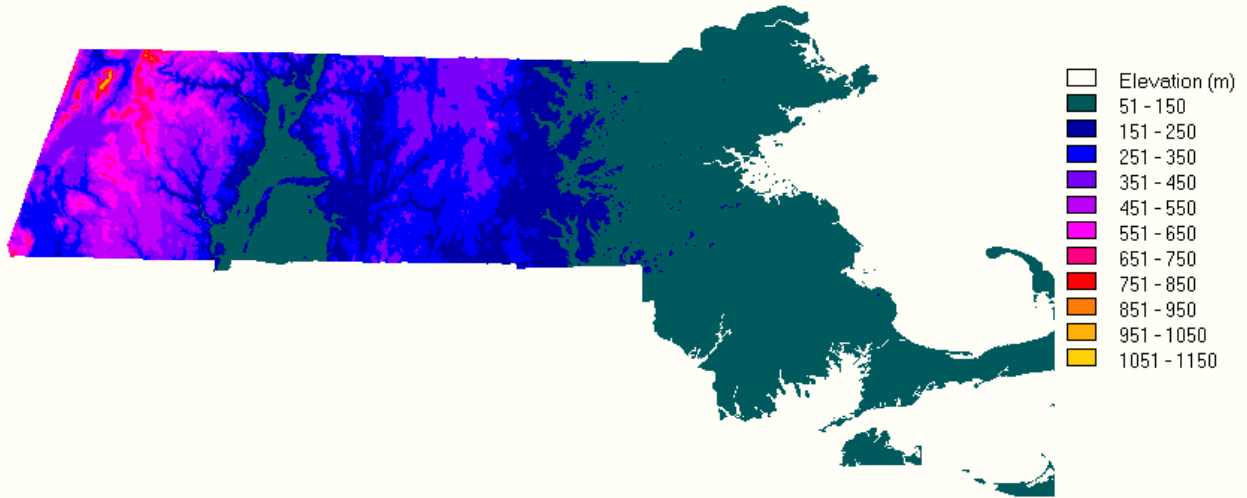
Results of Summing People per Pixel in Each Allocation Area

Category	Total	Legend
1	10568.080141	Happy Pets
2	122685.233638	Fido's Friend
3	81797.644129	Proposed Site
4	74526.583921	Kitten's Comfort
5	42500.494017	Animal Haven

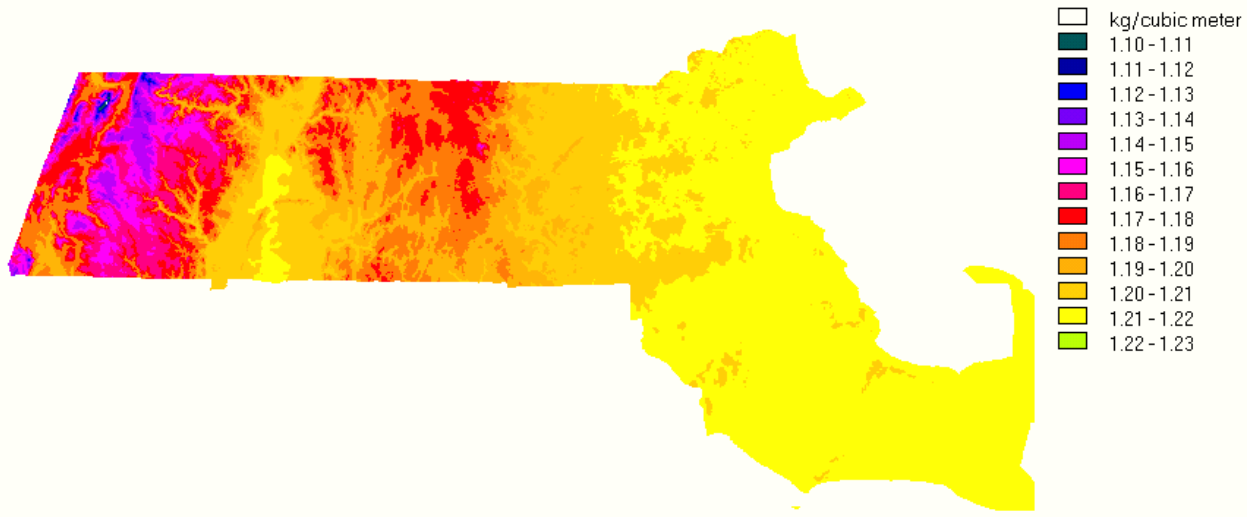
Digital Elevation Model



Elevation Surface: 50m Above Ground



Air Density



Estimated Wind Power

