Lesson 7: Basic Geospatial Analysis Techniques

## INTRODUCTION

## LESSON OBJECTIVES

By the end of this lesson, you will be able to:

1. Identify the basic concepts of spatial analysis.

2. Apply basic geospatial analysis techniques.

**LEARNING SEQUENCE**

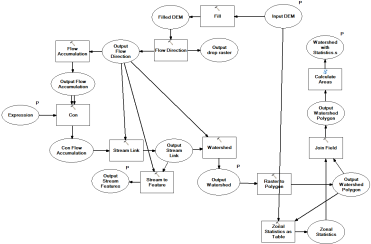
|  |  |
| --- | --- |
|  | |
| Required Reading | Read the following:  Basic Geospatial Analysis Techniques   * Data Analysis: Buffering and Dissolving * Geospatial Data Analysis: Overlay Operations * Geospatial Data Analysis: Table Operations |
| Resources | View the following:   * Automatic Spatial Selection |
| Assignments | Complete the following assignments:   * Lab: Basic Geospatial Analysis Techniques * Quiz: Basic Geospatial Analysis Techniques |

## INSTRUCTION

**Basic Geospatial Analysis Techniques**

## What is Geospatial Data Analysis?

Geospatial data analysis is the application of operations to manipulate or calculate coordinates and/or related attribute data. Geospatial data analysis is applied to solve problems such as bus routes, determining flood zones, and in determining suitable sites for construction. Spatial data analysis (SDA) uses spatial operations to manipulate or calculate. This is in contrast to analysis that does not use spatial operations to manipulate or calculate. As a note, SDA will be used to refer to spatial data analysis for the remainder of this lecture.

As an example of a SDA, this is an analysis flowchart for delineating watersheds using multiple input data layers, to produce an output of watersheds. While this model may look complex, you would be surprised at how each portion of the model is really not that difficult at all. The true power of SDA is the culmination of all of these different analysis methods.

## Important Considerations: SDA

For each SDA, you must consider the following:

* Input
* Operations
* Output

SDA typically involves using data from one or more layers to create an output. Therefore relationships can exist between the number of inputs, to the number of outputs: 1 to 1, one to many, many to one, and many to many. Speaking of the output it is important to note that the output may not always be spatial in nature. For instance, the output of an SDA may be a statistic, or report which is not spatial in nature.

|  |
| --- |
| **Key Facts**  SDA   * SDA typically involves using data from 1 or more layers to create output.   + One to One   + One to Many   + Many to One   + Many to Many * Output may not always be spatial in nature. The output can be a:   + Statistic   + Report |

## Basic Geospatial Analysis Techniques

There are seven basic geospatial analysis techniques that will be examined in this presentation.

* Selection
* Buffering
* Dissolve
* Overlay Operations
* Classification
* Table Operations
* Geocoding

## Selection

Selection is a basic type of geospatial data analysis that will probably be your most used operation when analyzing data within a GIS.

A selection operation is an operation that identifies features that meet certain conditions. The selection conditions may be spatial or aspatial. An aspatial selection condition selects objects based on attributes. A spatial selection condition selects objects based on position.

Whether you are performing an aspatial, or spatial selection, there are two types of selection methods available. The first is on-screen/manual selection, which is where you manually select the data with an input device. The second method is automatic where the user sets a condition, or writes a script, and lets the computer select the data that meets the conditions.

|  |
| --- |
| **Key Facts**  Selection   * Selection Operations identify features that meet conditions. * Selection conditions may be spatial or aspatial   + Select based on attributes – *aspatial*   + Select based on position - *spatial* |

## Aspatial Selection: Table Query

A table query is a very common GIS aspatial operation. A table query selects a subset of records based on values of specific attributes that meet certain criteria. Before you can do your first table query, however, you need to learn set algebra, and Boolean operators, as they are the building blocks of the table queries.

## Set Algebra

Let’s start with set algebra. Set algebra uses operations to determine whether two values are equivalent or not. The four basic set algebra operations are less than, greater than, equal to, and not equal to.

### Less than <

The less than operation checks to see whether the value on the left is less than the value on the right. The less than operation is represented by the left angle bracket symbol.

### Greater than >

The greater than operation checks to see if the value on the left is greater than the value on the right. The greater than operation is represented by the right angle bracket symbol.

### Equal to =

The equal to operation checks to see whether the values on both sides are equal to each other. The equal to operation uses the =.

### Not Equal to < >

The not equal to operation checks to see if the values on both sides are different from each other, and is equivalent to the combination of less than and greater than. The not equal to operation is represented by both the left angle bracket and right angle bracket used together.

### Important Note

Less than <

Greater than >

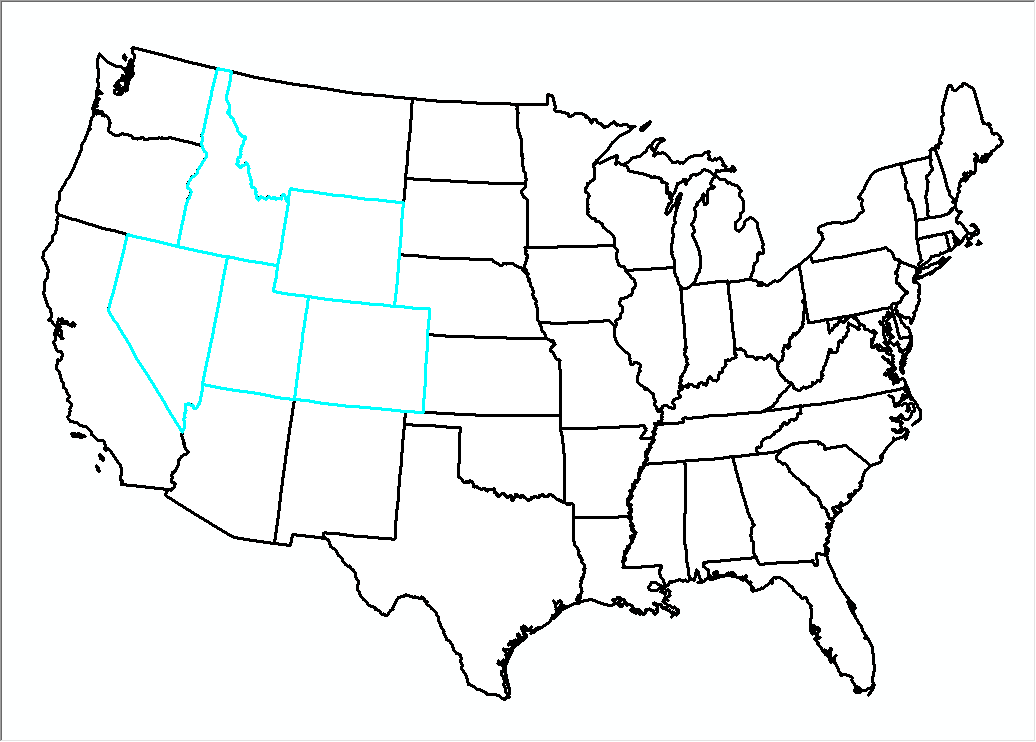
Equal to =

The three operations listed above annotated can be applied alone, or in combination. So for instance, you can perform the test of whether the value on the left side is less than or equal to the value on the right side. The symbol that would represent this operation, would be both the left angle bracket followed by the =. The result of all of these operations is either the value of “true”, or “false”. As a quick challenge question, I am making the claim that greater than and less than may not be applied to nominal attributes. Why do you think this is the case? The answer is because nominal attributes are only descriptors, and it is illogical to compare them with respect to magnitude or rank. It is, however, logical to compare them using the equal to, or not equal to operators.

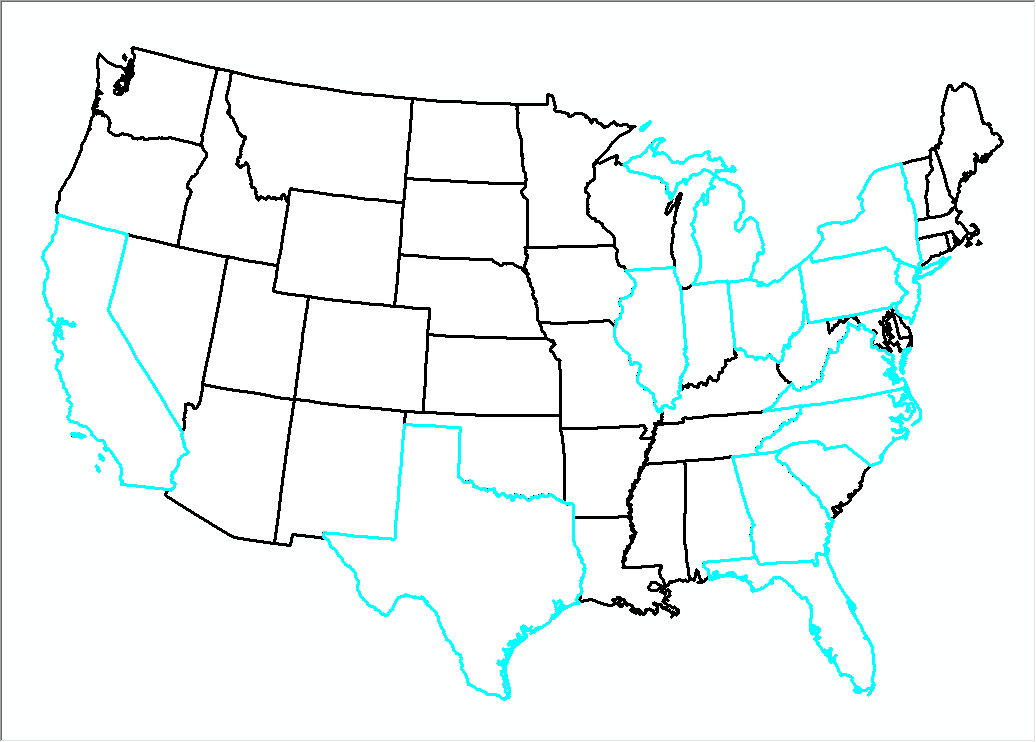
**Table Queries using Simple Set Algebra**

Provided are some examples of table queries using simple set algebra operations. Within the examples, we have the data of the lower 48 states of the United States of America. For each of the four maps, we will perform a simple set algebra operation to see which states are selected. Selected states are outlined in blue.

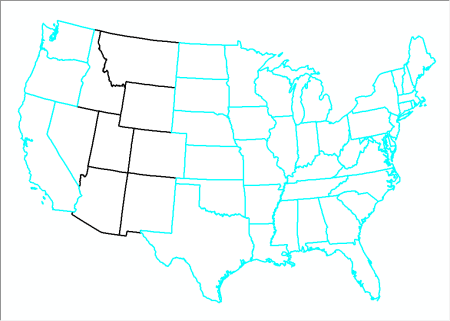
“Male Population”>”Female Population” Map

The table query is used to see whether in each state, male population is greater than female population. The way this works behind the scenes, is that the table query goes row by row in the table, and compares the value on the left, which is the male populations value, the value on the right, which is the female population, and, if the male population is greater than the female population, then the result is true, and that row, and object, are selected. Running that query on the map selects five states.

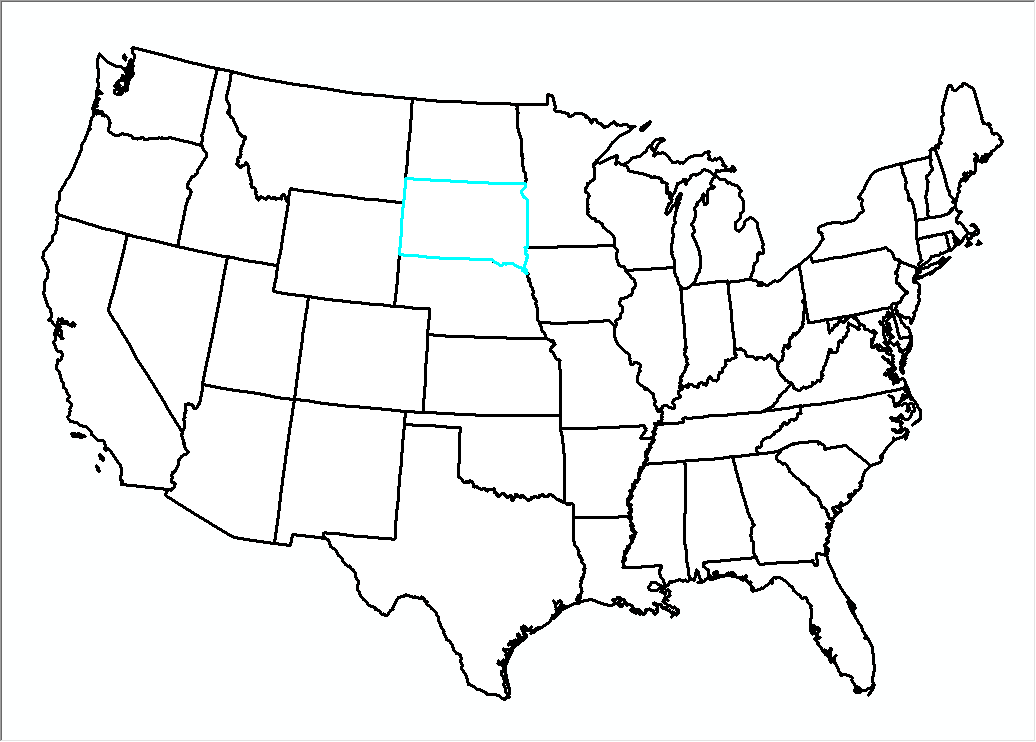
“Population” > 7,000,000 Map

This table query is comparing the population attribute with the number 7 million. So the states that are chosen are states where the population attribute is greater than the number 7 million.

“Region” <> ‘Mountain’ Map

This map selects all regions that are equal to “Mountain”. So again, it goes row by row in the table, and compares the region attribute to determine whether it is designated as a “Mountain” region. If the region attribute is equal to mountain, it is not selected. If the region attribute is not equal to “Mountain”, it is selected.

“State Name” = ‘South Dakota’ Map



This table query is state name equal South Dakota. If there is only one state with the name of South Dakota, only that state is selected from the table.

## Boolean Algebra

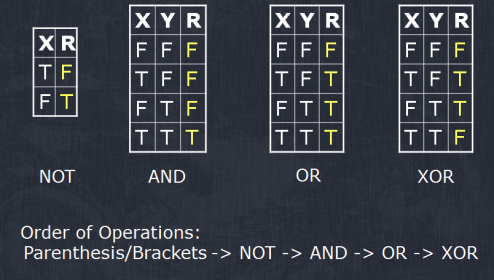
Now that we are familiar with set algebra; let’s move on to Boolean algebra. Boolean algebra is going to allow us to combine multiple set algebra operations into a more complex table query.

Boolean algebra has multiple conditional operators. The three operators we are going to focus on are the most common Boolean operators. They are: OR, AND, and NOT. The Boolean operators evaluate values on the left and right side of the operator, and then assign an outcome. The outcome is a Boolean, or binary result such as true/false, 0/1, on/off, or any other dichotomy.

In Boolean algebra, the order of operations does matter; therefore it is not uncommon to use many sets of parentheses to force a particular order of operations. It is also important to note that Boolean operators are not distributable inside of parentheses.

|  |
| --- |
| **Key Facts**  **Boolean Algebra**   * Uses conditions *OR, AND,* and *NOT* * Evaluated by assigning an outcome, true or false, to each condition * Order of operations **do matter** * Boolean operators are not distributable |

## Boolean Operators

These are the truth tables for the three Boolean operators. A truth table allows us to see what the result of the operation, ‘R’ is in the table; we can apply the input of X or X and Y. Review each of the following Boolean Operators.

### Boolean Operator: NOT

Look at the truth table for the simplest Boolean operation: not. The “NOT Boolean operator” simply reverses the result of the input. So, if the input is true, then NOT true is false. If the input is false, then the result is true. The “AND Boolean operator”, requires two inputs. One input is placed to the left of the AND Boolean operator, and one input is placed to the right of the AND Boolean operator.

### Boolean Operator: AND

The “AND Boolean operator” only assigns the value of true when the value to its left evaluates to true and the value to the right also, simultaneously, evaluates to true. In other words, in order for the “and Boolean operator” to be assigned a true value, both sides of the operation must have evaluated to true. Looking at the truth table, if the inputs are false AND false, the result is false. If the input is true AND false, the result is false. If the input is false AND true, the result is false. If the input is true AND true, the result is true. This makes the “AND operator” extremely selective.

### Boolean Operator: OR

Finally, let’s look at the “OR truth table”. With the “OR Boolean operator”, it only requires that one of the two inputs are true for the “OR operator” to return a true value. For instance, if the inputs are false OR false, the result is false, since neither of the inputs return be true. If the input is true or false, and the result is true if the input is false OR true, the result is true. If the input is true OR true, then the result is true. The order of operations of these Boolean operators are first, the innermost set of parentheses or brackets are evaluated first, followed by the Boolean operator, followed by the “AND Boolean operator”, followed by the “OR Boolean operator”.

## Practice Statements

Take a few moments, and consider each one of these statements. Determine whether each statement returns a value of true or false. To save space, false is represented by the letter F, and true is represented by the letter T.

### 1. NOT (F)

### 2. T AND T

### 3. F OR NOT (T)

### 4. T OR (F OR (T AND T)

### 5. (T AND T) OR F AND (NOT (F) AND T)

**Practice Statements: Answers**

1. NOT (F) 🡪 True

The first one states NOT false. “NOT” simply reverses the truth of the statement, therefore, NOT false equal to true.

2. T AND T 🡪 True

The second statement is true AND true. If you remember, for the “AND Boolean operator”, both sides must be true in order for the “AND operator” to return true, and it does in this case.

3. F OR NOT (T) 🡪 False

The next states false OR not true. For this one, we must remember our order of operations. Remember that not must be evaluated before the OR. Therefore, NOT true which leaves us with the statement of false OR false. The “OR operator” only returns true if one of the two sides returns true. Since a statement is false OR false, the result of the “OR operator” is false.

4. T OR (F OR (T AND T) 🡪 True

The next statement true OR open parenthesis false OR open parenthesis true AND close parenthesis followed by another close parenthesis. Remembering the order of operations, the innermost set of parentheses is evaluated first, which is the true AND true statement. True AND true evaluates to true which takes us to the next level of parentheses which is now false OR true. False OR true evaluates to true, because only one of the inputs needs to be true for the “OR operator” to return true. That leaves true OR true which evaluates to true.

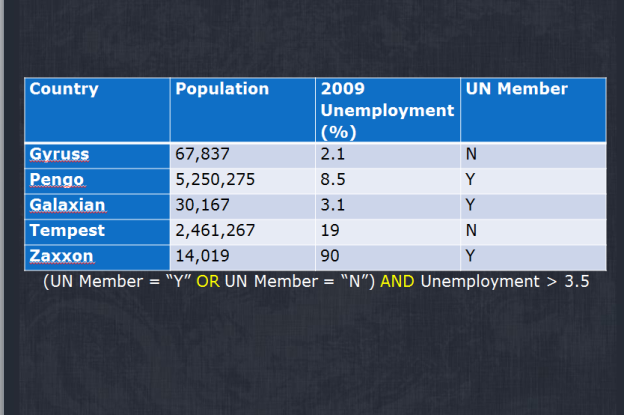
5. (T AND T) OR F AND (NOT (F) AND T) 🡪 True

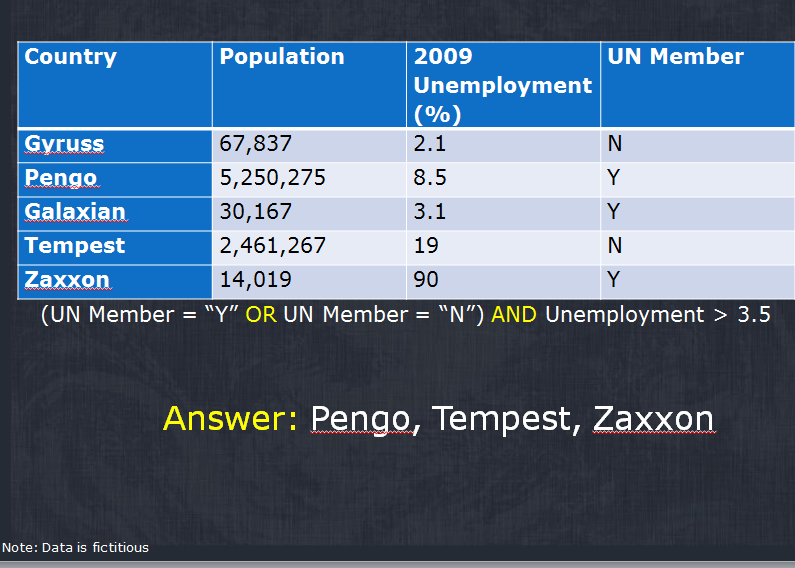
The last statement reads open parenthesis true AND true close parenthesis OR false and open parenthesis not false AND true close parenthesis. Again, we need to go to the innermost set of parentheses, and in this case, we have two options. Let’s start at the left and evaluate the true AND true. True AND true evaluates to true. Let’s go to the other set of parentheses. Order of operations states that we must evaluate the NOT before we can evaluate AND. So, NOT false equals to true which leaves us with true AND true which evaluates to true. The statement now reads in total true OR false AND true. So, once again, order of operations states that we must evaluate the “AND” first so false AND true is equal to false. That now leaves the statement true OR false. True OR false equals true is the result of the statement.

## Combining Set and Boolean Operators

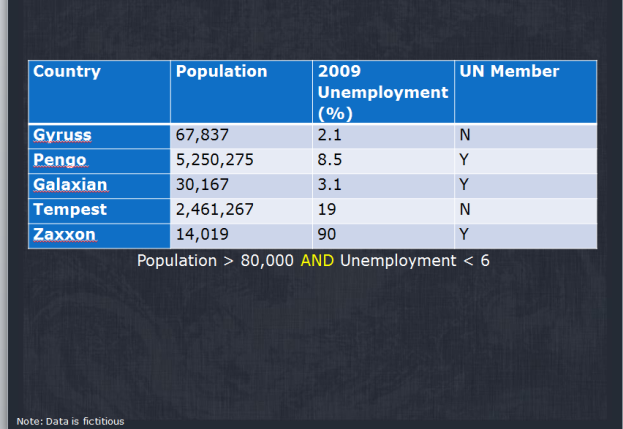
Now that you have been introduced to set and Boolean operators, let’s combine them together. If you remember, the result of a set operation is a true OR false value, and Boolean operators evaluate the inputs that are been evaluated to either true OR false. Take a look at the following charts to see how it all fits together.

### Chart 1

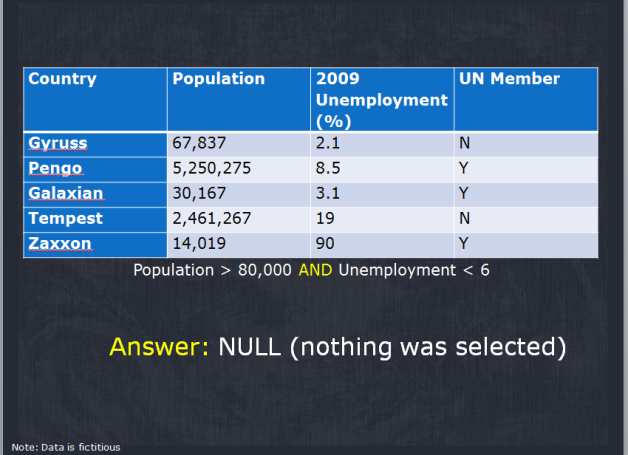
Focusing on the chart, it is an attribute table that has four attributes of country, population, 2009 unemployment, and whether the country is a United Nations member. Below the table, we have the table query we wish to execute on the table. The table query reads open parenthesis UN member equals Y OR UN member equal N close parenthesis AND unemployment is greater than 3.5. Spend a few moments and go row by row through the attribute table to see if you can determine which countries are selected based on this query.

The query selects the countries of Pengo, Tempest, and Zaxxon.

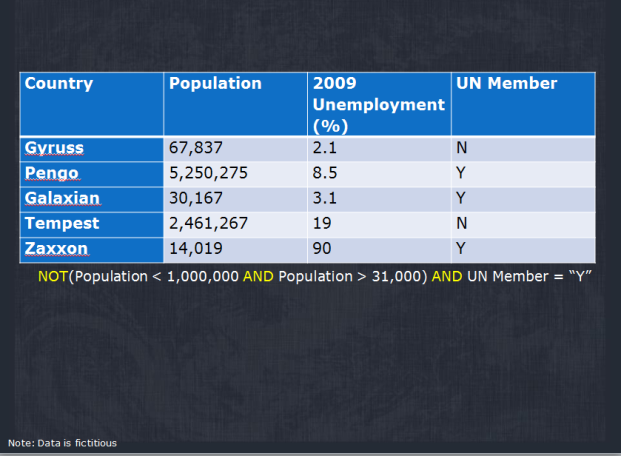
### Chart 2

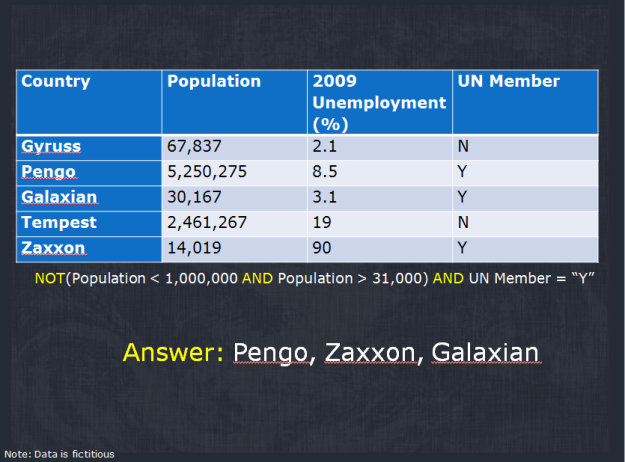


Now consider the table query of population greater than 80,000 and unemployment less than six. Spend a few moments to see if you can determine which rows were selected.

The answer in this case is NULL. The result of NULL means that no rows were selected using that query.

### Chart 3

Let’s consider one last table query. The table query reads NOT open parenthesis population less than 1 million AND population greater than 31,000 close parenthesis AND UN member equals Y. Take a few moments to determine which rows are selected by this table query.



This table query selects the countries of Pengon, Zaxxon, and Galaxian.

## Spatial Selections

When a computer performs an automatic spatial selection, there are multiple requirements that you can set. As an example it is important to introduce the spatial idea of adjacency, intersect, containment, and distance.

In a spatial selection operation, adjacency identifies features that share a boundary segment with the input feature. The intersect operation identifies features that simply touch, or even overlap the input feature. Containment identifies features that are completely within, or completely contain the input feature. Distance identifies features that are within a set distance of an input feature.

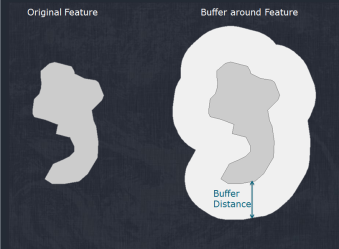
|  |  |
| --- | --- |
|  | Watch the video, [Automatic Spatial Selection](https://www.youtube.com/watch?v=P_W734x7G8M), (0:31). This video demonstrates both a manual and automatic spatial selection. The first selection made is a manual selection, where the user uses the mouse to select a country on screen. Next, the user performs an automatic spatial selection, by asking the computer to select all countries that touch the selected country. The computer then selects all countries that touch the boundary of the manually selected country. |

|  |
| --- |
| **Key Facts**  Spatial Selection Operations   * Adjacency identifies features that share a boundary segment. * Intersect identifies features that “touch”. * Containment identifies features that are “contained by” or “contain” other features. * Distance identifies features that are “within a set distance” of a feature. |

**Data Analysis: Buffering and Dissolving**

## What is a Buffer?

A buffer is a region that is less than or equal to a distance from one or more input features. Buffers can be created from point/line/polygon/raster geospatial data sets. Buffers are typically used to identify areas or objects “inside”, or “outside” the threshold distance. Can you think of any examples where you might use a buffer? One example of the use of a buffer could be to determine how many homes are within 1 mile of the coastline. In this case, the coastline is the input; 1 mile is the threshold distance, which results, in a polygon that extends 1 mile out from the coastline. We can then perform spatial selection by selecting all houses that are inside the buffer.

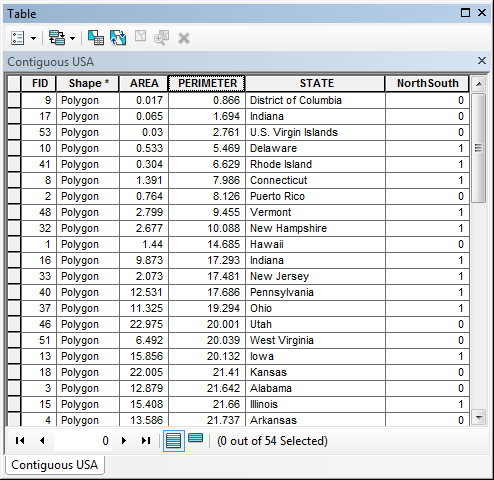
To illustrate a buffer, the polygon to the left includes the original input features. After specifying a buffer distance, and running the buffer tool, a new polygon is created at the set buffer distance and surrounds and follows the outline of the original input feature.

## Dissolve Operation

Now let’s discuss the dissolve operation. The dissolve operation combines

similar features within a data layer based on a shared attribute. For example, the data set on the left has the attribute of north-south. If the state is a northern state, it has the value of one, and if the state is a southern state, it has a value of zero. If we use this data set, and dissolve on this field, all states that have the value of one, are dissolved into a single polygon. All states that have the value of zero for the north-south attribute are dissolved into a second single polygon. The dissolve field can have as many different values as you choose; however, only records with identical values for that attribute will be combined into a single feature.

|  |
| --- |
| Dissolve combines similar features within a data layer based on an attribute. |



|  |  |
| --- | --- |
|  |  |

**Geospatial Data Analysis: Overlay Operations**

## Overlay Operations

Overlay operations involve combining spatial and attribute data from two or more spatial data layers. In other words, we are stacking the data, and looking for where the layers overlap. While the overlay operations may seem simple when used alone, they can actually be quite powerful operations when combined in a series. An example of an overlay operation would be the clip operation. For example, let’s say I have a data set that contains all the streets in the United States of America. However, I only want to have the streets in Arkansas. I can use the clip operation to cut out all the streets that are only in Arkansas by providing the state outline of Arkansas as the clipping feature. One thing to note about overlay operations, are that overlays require that the data be in the same coordinate system. This is so that the software can perform the highest accuracy of overlay. Some software packages automatically re-project on-the-fly if the data is not in the same coordinate system, but this is not the preferred method.

|  |
| --- |
| **Key Facts**  Overlay Operations   * Overlay Operations involve combining spatial and attribute data from two or more spatial data layers. “Stacking data” * Overlays require that data be in the same coordinate system. |

## Vector Overlays

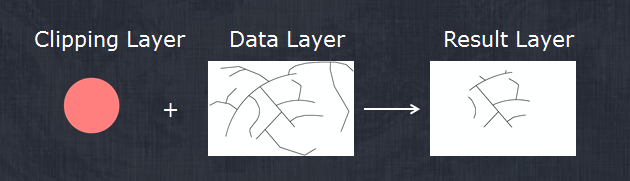
A vector overlay involves combining point, line, or polygon geometry and their associated attributes. All overlay operations create new geometry and a new output geospatial data set. You should be cautioned that with certain overlay operations, very large attribute tables may result if the overlay operations combine many layers, and each layer has a very large attribute table. Additionally, it might be possible that the combined attribute tables would cause duplicate attribute fields to exist. In these cases, you should consider reducing the number of transferred attributes to the minimum required, and rename duplicate fields so that there is no ambiguity.

## Basic Cases of Overlay

There are three basic cases of overlays: the clip function, intersection operation and the union overlay operation. Review each of the basic cases of overlay and the examples provided.

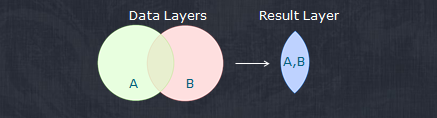
**Clip Function**

The first basic case of overlays is the clip function. The clip function defines the area for which features will be output based on a “clipping” polygon. I like to think of it in terms of cookie dough, and a cookie-cutter. The cookie dough is the data layer that we want to reduce in size based on our cookie-cutter. The clipping layer is the cookie-cutter that will determine how we cut out the part of the cookie dough we are interested in. It is important to note that in the clip operation; only the geometry and attributes of the data layer are transferred to the result layer. Neither the geometry nor attributes of the clipping layer are included in the output, just as you would not make your cookies in the oven with the cookie-cutter still surrounding the cookie dough.



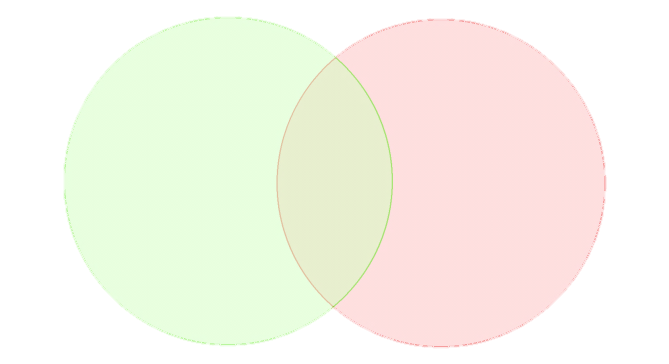
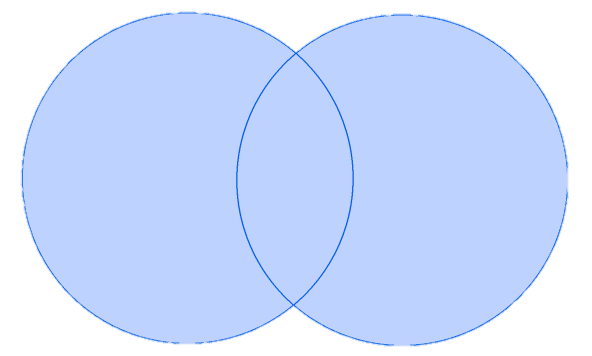
**Intersection Operation**

The next basic case of overlay is the intersection operation. The intersection operation combines data from two or more input layers, but the combination only exists for the regions where all input layers contain data. This is similar to the clip function; however the attributes and geometry of all input data layers are transferred to the result layer.



**Union Overlay Operation**

The third and final basic case of overlay is the union overlay operation. The union operation is an overlay that includes all data from all of the input data layers. Since the union operation combines all attributes, and all geometry, no geographic data is discarded in the union operation, and all corresponding attribute data is saved for all regions.



Data Layers

A

B

Result Layer

A

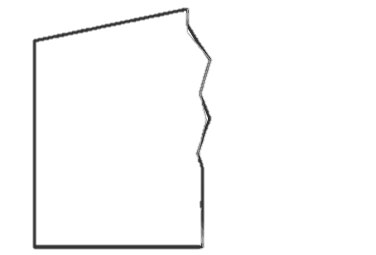
B

A,B

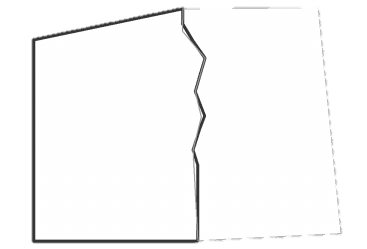
## Vector Overlay Problems

When performing vector overlays, there are a few problems that you should be aware of. In the case where the input features represent common features that are represented in both layers, they may have slightly different geometry. This creates sliver polygons when the overlay operations are performed. For instance, imagine we have one input data set the represents the national boundary of Canada created at a 1 to 5,000,000 scale. And we have a second input data set of the national boundary of the United States of America created at 12 1 million scale. If we stack these two layers on top of each other, it is probable that the boundary lines will not overlay exactly, as they were created at different scales, and therefore the Canadian boundary line has probably been simplified more than the United States boundary line. The areas where the two input data sets do not match, or result in very small, typically insignificant sliver polygons that need to be dealt with. There are several methods that exist to reduce the occurrence of sliver polygons, such as manually snapping the boundaries together, or introducing fuzziness to the edges of the overlay operation, or simply manually deleting the sliver polygons.

### Example 1

To illustrate the idea of a sliver polygon, here are two different data sets, one outlined in a bold black line, and the other outlined in a thinner grey line. Where the two data sets overlay, they do not match exactly for whatever reason. If we perform an intersect, clip, or union operation on these two input data sets, we are going to create three sliver polygons.

### Example 2

The sliver polygons created between these two input data sets are highlighted in red. The sliver polygons are probably insignificant, and are the results of data sets being produced by different agencies, at different times, and probably at different scales. It is up to you to determine how you would like to resolve for the sliver polygons.

|  |
| --- |
| **Key Facts**  Vector Overlay Problems   * Input features   + Common features represented in both layers, may have slightly different geometry * Having slightly different geometry creates:   + Sliver polygons when the overlay operations are performed * Several methods exist to reduce the occurrence of sliver polygons:   + Manually snapping boundaries together   + Introducing fuzziness to edges of overlay   + Manually deleting the sliver polygons |

**Geospatial Data Analysis: Table Operations**

## Table Operations

It is important to understand the concept of table operations and how they apply to attribute tables in a geospatial data set. Combining tables of attributes is a powerful analysis technique. There are three common types of table operations: intersect, union, and join.

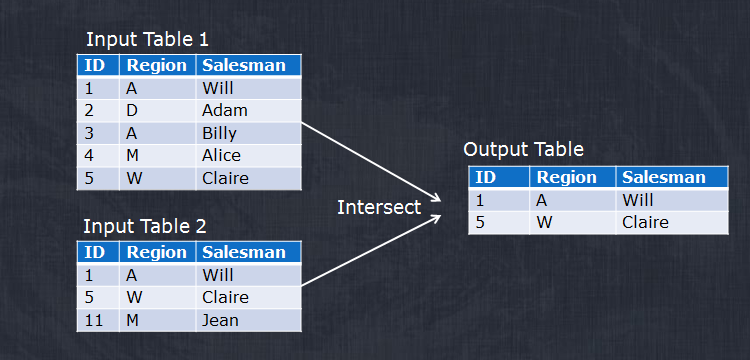
Each one of these operations can be combined in tables in different ways, and for different purposes. It is important to note, that in order for multiple tables to be inputs to intersect, or a union table operation, the tables must have the same fields, and field types, which are commonly referred to as the schema. Let’s take a look at each one of these three table operations in

more detail.

### Intersect

The intersect table operation produces an output table that only contains records that all of the input tables had in common. That means that every attribute for the entire row was identical. As an example, we have two input tables that have the same schema, as all input tables must have the same schema for the intersect operation. The output table only includes the two records that were found in both input tables. In this case, the row where the ID was one, region was A, and salesman was Will, in the second row was where the ID was five, the region was W, and the salesman was Claire.

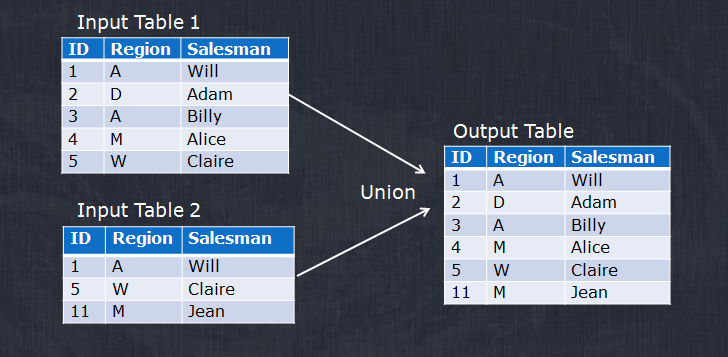
* An output table only contains records that all input tables have in common.



### Union

The union table operation produces an output table that contains all records that existed in all input tables. For example, we have the same two tables as before, and notice that the output table combines all the rows from both input tables. Again, note that both input tables have the same schema, as this is required for union.

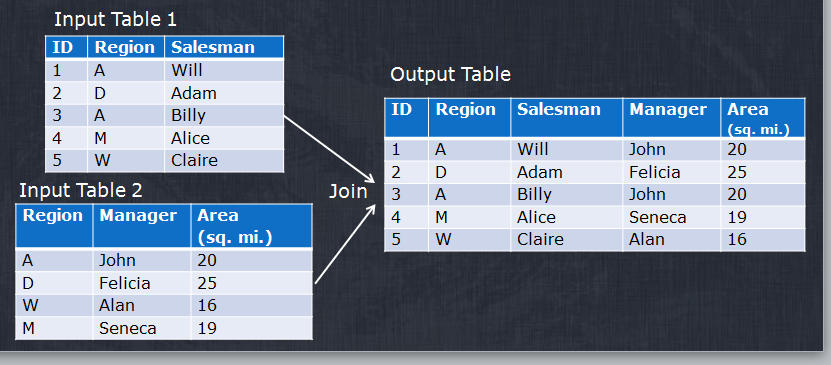
* An output table only contains all records that exist in all input tables



### Join

The third table operation is the join operation. The join operation brings together two tables based on a shared attribute. Based on the shared attribute, attributes of one input table is appended to the attributes of the other input table. For example, we have two input tables that share the common attribute of region. Based on the region attribute, the manager and area attributes of Input Table 2, are appended to the rows and input table 1. The region attribute, matches its value between the two tables, which determines which attributes of which row from Input Table 2 is copied over. For example, if we look at the output table, the row with ID 1 had a region A, which means that the manager John and area of 20 was appended. In row where the ID was 2, since the region was value of D, then the row with the same region of value D in the second input table is copied to the output table. Also note, that for the record were ID was 3, the region is A again, so once again, the manager John, and area 20 are copied to the output table for that row.

* Based on a shared attribute, attributes of one input table are appended to attributes of the other input table.



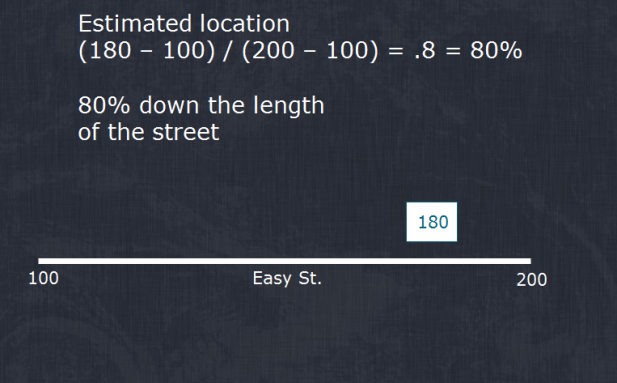
**Geospatial Data Analysis: Geocoding**

## What is Geocoding?

The process of Geocoding involves spatially referencing features by address in order to determine their position on the face of the Earth. For example, the GPS unit in your car performs geocoding every time you enter a street address. The GPS unit takes the street address, and associates it with road data sets, to determine the coordinate that that addresses exists at. There are two typical types of geocoding: linear geocoding, and area geocoding. Linear geocoding assumes that addresses vary linearly along a feature which is typically a line. Area geocoding assigns the geocoded location to the entire area that matches the input address. Let’s look at an example of linear geocoding and area geocoding.

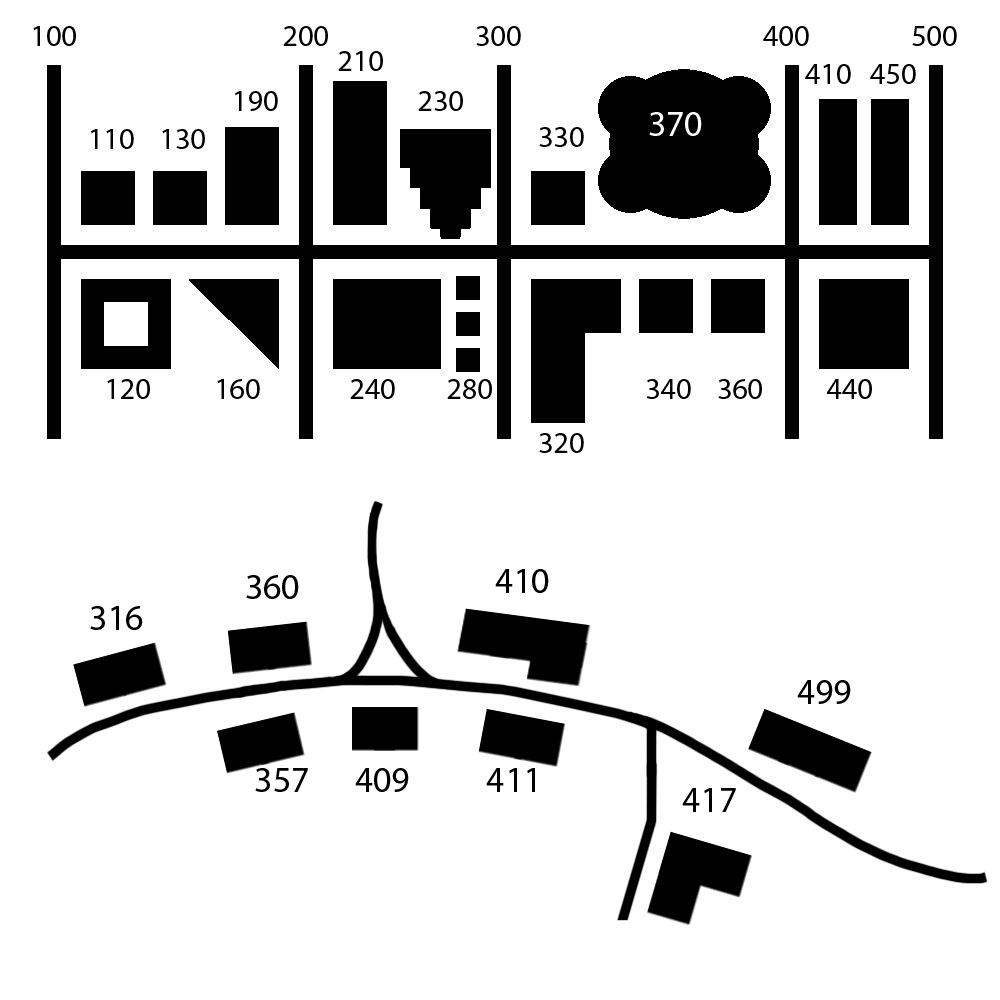
### Linear Geocoding

As an example of linear geocoding, imagine a geospatial data set that has the centerlines of roads. Between intersections, each road segment has a beginning address, and an ending address. If we geocode using that data set, since we do not know exactly where each address is on the block, we assume that the addresses are spaced evenly between the start and end range of addresses in the block, and therefore estimate where the addresses exists.

So for the example on the slide, this block of Easy Street starts at the address of 100, and ends at the address of 200. We wish to know where the address 180 Easy Street is located, we first identify the feature named Easy Street, then narrow down the results to the block where our input address exists. Next, we estimate how far down Easy Street that address is assuming equal spacing of addresses. The major downside to linear geocoding, is at the location may be misplaced if the addresses are linearly spaced. The major upside of linear geocoding is that the data sets are commonly, and often freely available and they are easy to construct.

### Area Geocoding

Now let’s look at an example of area geocoding. This illustration shows four street segments, from addresses 100 to 500, and each block has the address range of 100. Here, the building footprints have been stored in a polygon geospatial data set, and each polygon has the address of that building. As each building has a slightly different shape and size, you can easily see that the addresses are not evenly distributed along the blocks. With area geocoding, we do not need to worry about whether the addresses are evenly spaced, as we can point to the exact area that the address exists. This is a much more accurate way to geocode than linear geocoding. The major downside to area geocoding is that the data sets are not commonly available, and are very expensive to construct.



## SUMMARY

In this lesson you learned about basic geospatial analysis techniques. Here are important points to review.

* Geospatial data analysis is the application of operations to manipulate or calculate coordinates and/or related attribute data.
* Spatial data analysis (SDA) uses spatial operations to manipulate or calculate.
* Set algebra uses operations to determine whether two values are equivalent or not. There are four basic set algebra operations: less than, greater than, equal to, and not equal to.
* Booleen algebra enables one to combine multiple set algebra operations into a complex table query. Booleen algebra uses conditions: OR, AND, and NOT. The order of operations does matter.
* A buffer is a region that is less than or equal to a distance from one or more input features.
* The dissolve operation combines similar features within a data layer based on a shared attribute.
* Three common types of table operations are intersect union and join.
* The process of Geocoding involves spatially referencing features by address in order to determine their position on the face of the earth. There are two types of geocoding: linear and area.

## ASSIGNMENTS

1. Lab: Basic Geospatial Analysis Techniques

2. Quiz: Basic Geospatial Analysis Techniques