Chapter 9. Spatial Joins

Objectives

- Learning the purpose and capabilities of spatial joins
- Correctly setting up spatial joins based on cardinality and feature type
- Learning to solve problems with spatial joins

Mastering the Concepts

GIS Concepts

What is a spatial join?

Chapter 6 presented attribute joins performed on tables—for example, joining information in an earthquake statistics table to a map of states to create a graduated color map of damages. This join is based on a common field, the state name, and has a **cardinality** of one-to-one. The join results in combining the two tables as if they were one table. The destination table receives the information from the source table.

A **spatial join** is similar to an attribute join, except that, instead of using a common field to decide which rows in the table match, the *locations* of the spatial features are used. The spatial join uses either a containment criterion (one feature inside the other) or a proximity criterion (one feature close to another).

Like attribute joins, spatial joins designate a source feature class and a destination feature class. Unlike an attribute join, which appends the source attributes to the existing destination table, a spatial join creates a new feature class. It retains the features from the destination layer and appends the attribute information from the source layer (Fig. 9.1). The two original feature classes are unaffected. The destination feature class determines the type of features in the output feature class. If an airports feature class is joined to a cities feature class with cities as the destination, then the output feature class contains cities.

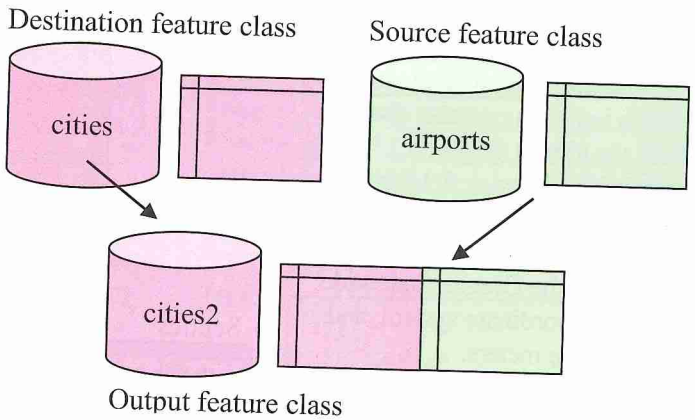


Fig. 9.1. A spatial join keeps features from the destination and appends attributes from the source.

Distance joins

A **distance join** uses a proximity criterion to link one feature and its attributes to another based on whether one feature is closest to another. Figure 9.2a shows the details of the join between airports and cities diagrammed in Figure 9.1. The source feature class, airports, has been joined to the destination feature class, cities. The output feature class contains cities. Each city has been given the attribute information from the airport that lies closest to it, and a new field has been

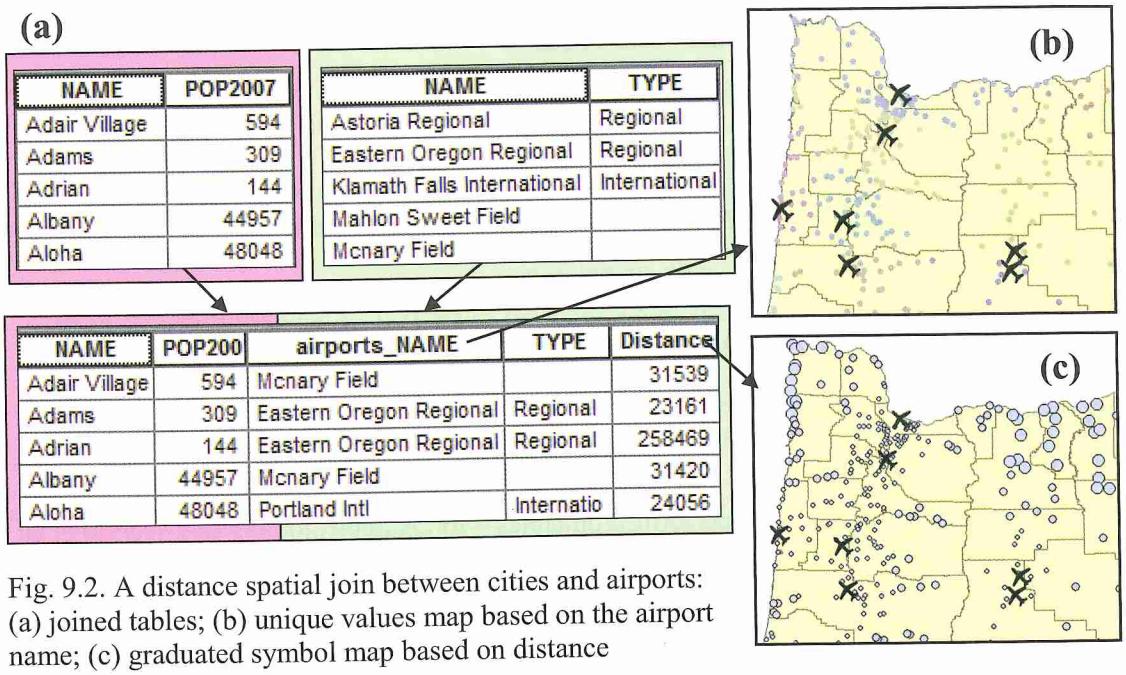


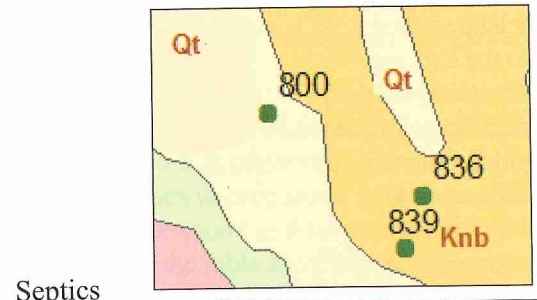
Fig. 9.2. A distance spatial join between cities and airports: (a) joined tables; (b) unique values map based on the airport name; (c) graduated symbol map based on distance

added to record the distance. The attribute table contains two parts, the original data from cities and the joined data from airports. So McNary Field is the closest airport to Adair Village, and Eastern Oregon Regional is the closest airport to Adams.

Two maps have been created from the new joined layer. Figure 9.2b is a unique values map based on the airport name, so each dot gets a color based on the closest airport. The colors indicate which cities are served by each airport, assuming that people will drive to the closest one. The second map is a graduated symbol map based on the distance field, with the larger circles indicating greater distance from the airport (Fig. 9.2c). The units of the distance field are always given in the stored coordinate system units. These data are in the Oregon Statewide Lambert coordinate system, and the units are meters.

Inside joins

In an **inside join**, the records of the feature classes are joined based on whether one feature is inside another (wholly or partly). Figure 9.3 shows a point layer containing septic system locations and a polygon layer showing geological units. Imagine that a community has a porous geological aquifer that provides the city water supplies.



Septics

ID *	cntycode	basin	gma
800	453	14	8
836	453	14	8
839	453	14	8

Geology

LABEL	NAME
Kpg	Pecan Gap Chalk
Knb	Navarro Group
Qt	Terrace deposits

ID	cntycode	basin	gma	LABEL	NAME
800	453	14	8	Qt	Terrace deposits
836	453	14	8	Knb	Navarro Group
839	453	14	8	Knb	Navarro Group

Fig. 9.3. A spatial join gives each septic system the attributes of the geology polygon within which it lies.

Extensive development outside city limits has caused the city concern about potential contamination of the aquifer by faulty septic systems. Assessing the threat requires identifying the number of septic systems that occur in the outcrop area of the aquifer.

A spatial join solves this problem perfectly. Septics is the destination layer and becomes the output feature class. The geology attributes of the polygon containing the septic system is appended to the output septics table. As shown in Figure 9.3, septic system 800 falls inside the Terrace Deposits (Qt) polygon in the geology layer, and septic systems 836 and 839 fall inside the Navarro Group (Knb) unit. The output table will be helpful in assessing how many septics fall on sensitive geological units.

Spatial joins may be performed on any two spatial data layers. A user can join points to points, polygons to polygons, lines to points, and nearly any combination of the three types of data. The output layer will always have the same feature type as the destination layer.

Cardinality

Cardinality is an important issue for spatial joins, just as it is for attribute joins. Because records in tables are being matched together, the Rule of Joining must be fulfilled in spatial joins also. Each feature record in the destination table must have one and only one matching record in the source table. This condition is met if the cardinality of destination to source is one-to-one or many-to-one. In attribute joins, we had to use a relate if the Rule of Joining was not fulfilled. With spatial joins, we must use a **summarized join** if we encounter a one-to-many relationship.

Recall that the Summarize function calculates statistics for groups of records in a table. It uses one field to divide the records into groups, and then it calculates statistics for other fields for each group. In a summarized join, each feature in the destination layer is matched to many features in the source. Statistics are calculated for that group of features, and the result is appended to the feature record.

In Figure 9.2, imagine reversing the join so that airports is the destination layer and cities is the source layer. Each airport has many cities that are closer to it than to any other airport. Instead of attaching the single closest city, a summarized join finds all of the cities closer to the airport and calculates one or more statistics, for example, the sum of the city populations. Then for each airport we would know the total number of people being served by that airport (i.e., the sum of the populations of the cities that are closer to that airport than to any other). Figure 9.4a shows the output table of the joined layer with a Count field representing the number of cities and the Sum_POP_98 field representing the total people in those cities. So Portland International has 69 cities close to it with a total of 870,598 people. Figure 9.4b shows a proportional symbol map based on the Sum_POP_98 field to represent the total potential population served by the airport.

NAME	TYPE	Count	Sum_POP
Rogue Valley Intern	Interna	22	192645
Astoria Regional	Region	14	34898
Mahlon Sweet Field		23	266623
Klamath Falls Intern	Interna	8	46110
North Bend Muni	Munici	21	99896

(a)

(b)

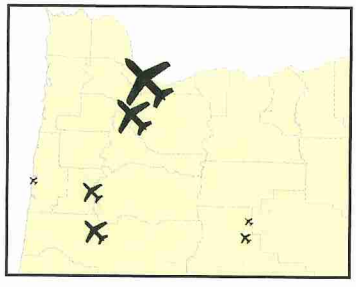


Fig. 9.4. A summarized spatial join: (a) output table; (b) map based on sum of population served

This type of problem, estimating usage of airports by surrounding populations, is known as an allocation problem, in which provision of services is being assessed. In truth, spatial joins are not

the best approach—people select airports based on more than distance: cost, schedules, and so on. More sophisticated techniques for solving allocation and logistics problems are available but require the purchase of the Network Analyst extension.

Analysis problems typically involve making assumptions about the forces at work, and nearly always these assumptions are simplifications of the actual situation. We made an assumption that people choose the closest airport based on distance, and although it is true in a general way, it is not the whole story. That does not make the analysis wrong—airports close to many people are going to be serving many more passengers than airports in a lower population area—but it is important that we are cognizant of the assumptions being made in any analysis and remain sensitive to how they may impact the results. In this case, we have ignored tourist/business air traffic as well as the fact that many people will drive further to get a better deal. Thus we are probably overestimating use at the smaller airports and underestimating use at the large ones.

Types of spatial joins

Spatial joins fall into four main types according to the cardinality of the relationship between the joined layers (simple or summarized) and the choice of spatial criteria (inside or distance). Figure 9.5 shows a matrix of the four possible combinations. A **simple join** may be used whenever the cardinality is one-to-one or many-to-one so that the Rule of Joining is maintained. In a one-to-many relationship, a summarized join must be employed.

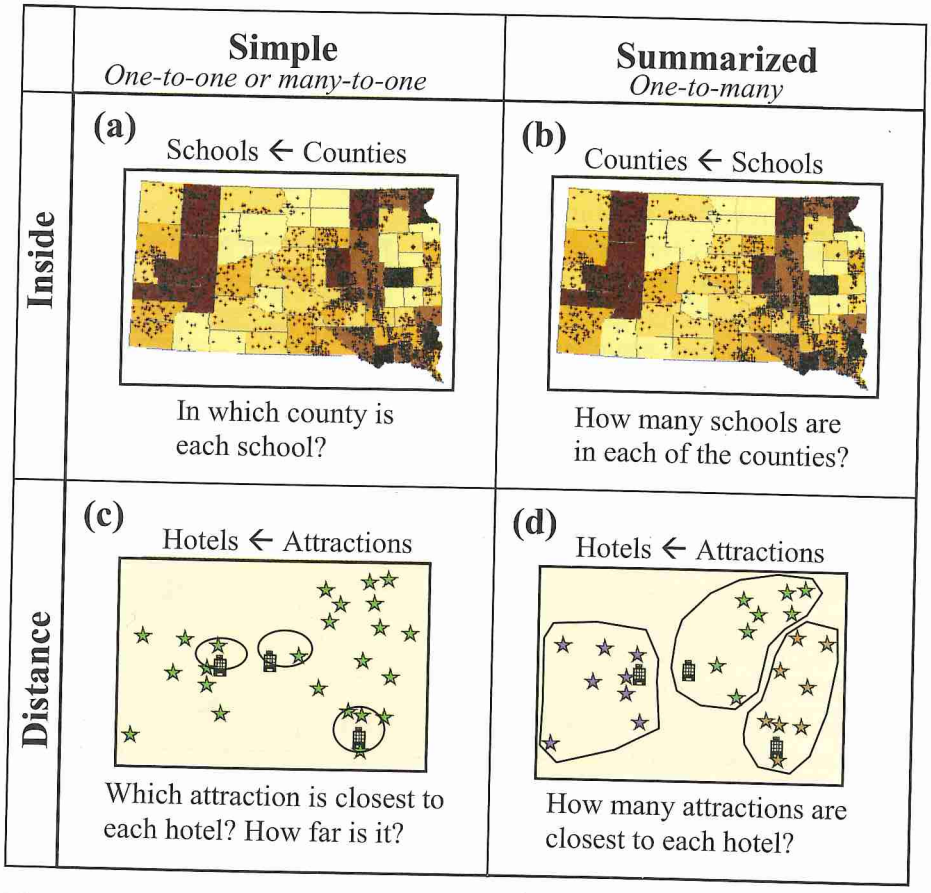


Fig. 9.5. Matrix of spatial joins resulting from different choices of destination table, spatial condition (inside or distance), and cardinality

Consider the simple join types first. In Figure 9.5a, schools and counties are being joined, with schools as the destination layer (some call it the target layer). The output layer contains schools, and each school will have the attributes of the county it falls inside. With the output layer, one could answer the question "In which county does a particular school lie?" This example is a *simple inside join.* The previous example of the septic systems is also a simple inside join.

If, however, the destination layer is reversed, the cardinality of counties to schools is one-to-many, and a simple join is not possible. A summarized join may be employed in this case. The summarized join groups the schools together based on which county they are in and generates a single record of statistics for the group. This single summarized school record can then be appended to the matching county in the destination table. A Count field is always generated, containing the total number of schools in each county. Figure 9.5b shows this as a *summarized inside join.*

Distance joins also come in simple or summarized varieties. Consider evaluating the desirability of several hotels based on their distance to local tourist attractions, using a layer of hotels and a layer of attractions with the yearly number of visitors in each one. What is the cardinality of this relationship? This question seems confusing because there are several hotel sites and many attractions. In a distance join, it is the question that dictates the cardinality. If we ask "Which attraction is closest to the hotel?" in order to find out whether that attraction has a large visitor pool, we have specified a one-to-one criterion because only one attraction can be *closest* to each hotel. A *simple distance join* suffices, and each hotel appears in the output table along with the attributes of the attraction closest to it and the distance between them. In Figure 9.5c, the circles connect each hotel site with its closest attraction.

A different question may be asked: "How many tourist attractions are closer to this hotel site than they are to any other?" In this case, we are interested in evaluating the richness of choice of attractions for each hotel or perhaps the total combined visitor pool from all the closest attractions. In this case, each hotel is connected with many attractions, as shown in Figure 9.5d, and summary statistics for the group, such as the number of attractions and the sum of the visitor pools, may be added to the record for each site. This would be a *summarized distance join.*

Feature geometry and spatial joins

The available join types will depend in part on the geometry of the features being joined. When joining points to points, for example, an inside join type does not apply. Thus, only two options are available when joining points to points, simple distance, or summarized distance. Table 9.1 lists all of the possible geometry combinations and the join types that can be applied to each. (In this table we break the convention of putting the destination first in order to match the descriptions in the spatial join window in ArcGIS.)

Let us examine some other combinations of geometry types. Consider this problem: Many counties in South Dakota do not have a hospital. The state emergency planning office wishes to know the hospital closest to each county and how far away it is. They require a list of counties, each with the closest hospital attached to it. Thus, counties is the destination layer, and hospitals is the source layer. According to Table 9.1, joining points to polygons requires either

cntyhosp

NAME	hospitals_NAME	Distance
Shannon	Battle Mountain National Sanitar	38424
Fall River	Battle Mountain National Sanitar	0
Bennett	Battle Mountain National Sanitar	110128
Pennington	Bennett Clarkson Hospital	0
Lincoln	Canton-Inwood Hospital	0
Yankton	Canton-Inwood Hospital	51493

Fig. 9.6. This table resulted from a distance join of hospitals to counties.

a summarized inside or a simple distance join. In this case, the simple distance join is the correct choice. After the join, each county has a field indicating the name of the closest hospital (Fig. 9.6) as well as the distance from the county to the hospital. When the hospital is inside the county, the distance of zero is assigned.

Table 9.1. Join types are available for each combination of feature geometries in a spatial join. The second feature type is the destination layer in each case.

Geometry Type	Join Type	Example
Points to points	Simple distance	Find the hospital closest to each town.
	Summarized distance	Find all the towns closer to one hospital than to any other hospital.
Lines to points	Simple distance	Find the water main closest to the proposed building site.
	Summarized inside	Find the total voltage of all electric lines meeting at a substation.
Polygons to points	Simple inside	Find the soil type that underlies each gas station.
	Simple distance	Find the lake that is closest to each campground.
Points to lines	Simple distance	Find the elementary school that is closest to each residential street.
	Summarized distance	Find the total number of septic systems closer to a particular stream than to any other stream.
Lines to lines	Summarized inside	Find the number of roads that cross each river.
	Simple inside	Give a section of hiking trail the attributes of the road it follows for a short distance.
Polygons to lines	Summarized inside	Give a stream the average erosion index of the soil types it crosses.
	Simple distance	Find the lake closest to a hiking trail or the national park within which a road lies.
Points to polygons	Summarized inside	Find the total number of schools and students in a county.
	Simple distance	Find the town that is closest to a lake. A point inside a polygon is given a distance of zero.
Lines to polygons	Summarized inside	Find the total number of rivers crossing a state.
	Simple distance	Find the carrying capacity of the closest power lines to an industrial site.
Polygons to polygons	Summarized inside	Find the total population of all counties that intersect part of a watershed.
	Simple inside	Find the county within which a lake falls completely.

In Figure 9.7, a unique values map was created based on the hospital name. In the northeast part of the state, the purple counties are closest to the hospital in the purple area, the green counties to the hospital in the green area, and so on. The long, hatchet-shaped county in western South Dakota (Pennington County) has several hospitals. Since all three lie inside Pennington, they all have a distance of zero, and Pennington's closest hospital was arbitrarily assigned based on which hospital was found in the table first. The lighter pink counties in the central part of the state were assigned to a different hospital in Pennington County and appear in a different color. The map in Figure 9.8 shows each county displayed according to its distance to the closest hospital.

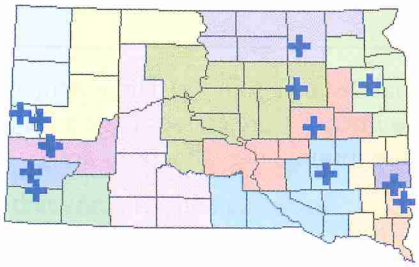


Fig. 9.7. Counties with the same color are closest to the same hospital.

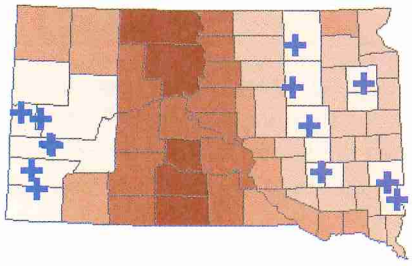


Fig. 9.8. Counties colored according to distance from the closest hospital

Figure 9.9 demonstrates an example of joining points to lines to predict the susceptibility of streams to contamination from septic systems. In this analysis, we make the assumption that, the more septic systems that are close to the stream, the greater the stream's susceptibility to contamination. The point locations represent the centers of one-mile by one-mile sections, and the size of the symbol indicates the number of septic systems in the section. We need to join each stream line to the closest septic systems, so streams is the destination layer. Joining points to lines requires either a simple distance join or a summarized distance join. In this case, a summarized distance join, which sums the total septic systems that are closest to each stream, is correct. The map shows the results: the thicker the line symbol of the stream, the more septic systems are closest to that stream and the higher the susceptibility of the stream to contamination.

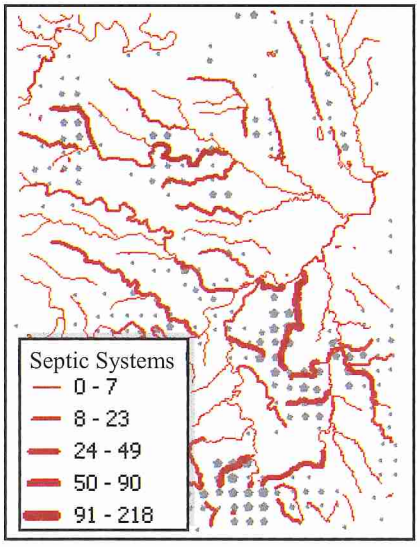


Fig. 9.9. Joining septic systems to streams to evaluate stream susceptibility to contamination

Coordinate systems and distance joins

Distance joins should always be performed on layers with projected coordinate systems that preserve distance. If a join is performed on a layer with a geographic coordinate system (GCS) and units of decimal degrees, two problems arise. First, the distances reported in the table will be in decimal degrees. Degrees cannot be easily converted into miles or kilometers because the conversion factor changes with latitude. Second, the result could be invalid. The distance algorithm relies on the relative *x-y* coordinates of the features to calculate distances and assumes a Cartesian coordinate system to do so. Degrees of latitude and longitude are spherical coordinates, not Cartesian, and the relative distances calculated may be incorrect.

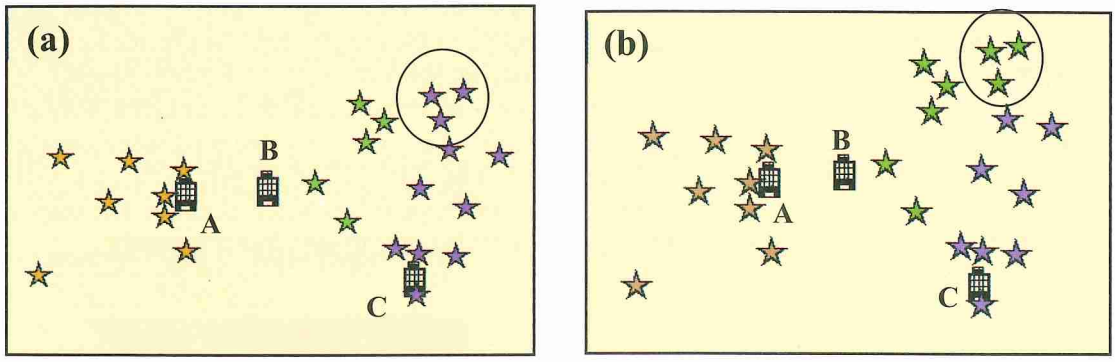


Fig. 9.10. The effect of the coordinate system on a spatial distance join. (a) Three attractions are incorrectly joined to Hotel C (purple stars) when the input feature classes have a GCS. (b) When the feature classes are projected to UTM before joining, the attractions are correctly assigned to Hotel B (green stars).

Figure 9.10 shows two cases of a summarized distance join of tourist attractions to hotels. In Figure 9.10a, the join was done with both layers in a GCS. Notice that the three attractions in the northeast corner were assigned to Hotel C based on their distance from it in decimal degrees. In Figure 9.10b, both layers were projected to UTM prior to joining, and the attractions were assigned instead to Hotel B. Since a UTM projection preserves distance within the zone, we know that the second example gives the correct spatial distances between the attractions and the hotels and is the valid result. Notice that the spatial distribution of attractions is elongated in Figure 9.10a, as a result of being displayed in a GCS.

It is not sufficient to set the data frame to a projected coordinate system; the layers themselves must be projected. Furthermore, the projected coordinate system should be one that preserves distance over the region of analysis, such as UTM, State Plane, Equidistant Conic, and so on. Otherwise, the distance analysis may again be incorrect.

About ArcGIS

Spatial joins are performed on two input feature classes and result in a new feature class that may be stored as a shapefile or geodatabase feature class. The original inputs are unchanged. Spatial joins are initiated using the same method as attribute joins.

Choosing the join type

As in an attribute join, the process begins by deciding which layer is the destination. The user right-clicks the destination layer, chooses to join the data based on spatial location (Fig. 9.11), and enters the desired source layer.

The join menu always offers two ways to decide how to match the fields, taken from the four types shown in Figure 9.5. Consider joining schools to counties, with counties as the destination layer as described in Figure 9.5a. Each county can have more than one school, so it is a one-to-many join. There are two options in the window based on the choices listed in Table 9.1. The user must choose between them based on the desired result (Fig. 9.11).

For this example, Option 1 offers an *inside summarized join,* which gives each county a statistical summary of the attributes of all the schools inside it, such as the total number of schools or the sum of the students. Option 2 specifies that a *simple distance join* should be

performed, which gives each county the attributes of the closest school. In this case, the second option is nonsensical. All of the schools are inside the county and would be assigned a distance of zero, so it avails little to find the "closest" one. In this example, the summarized join is the appropriate choice.

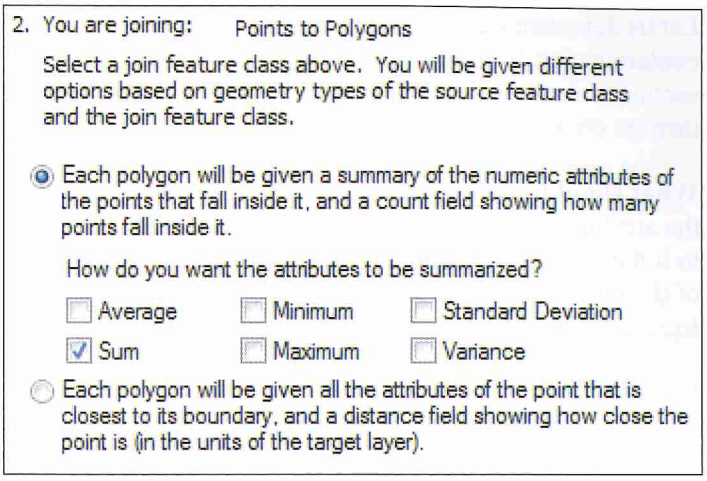


Fig. 9.11. A one-to-many cardinality is handled by either a summarized join or a distance join.

Figure 9.12 shows the result of the join. The resulting layer is a map of counties with the original county attributes plus a field called Count_ that contains the number of schools in the county. From this field, the graduated color map was created, showing the number of schools in each county. When summarizing, the user can choose from several statistics—minimum, maximum, average, and so on. All numeric attributes in the source table are summarized using the chosen statistics and are placed in the output table. String fields cannot be summed or averaged, so they are not included in the output table.

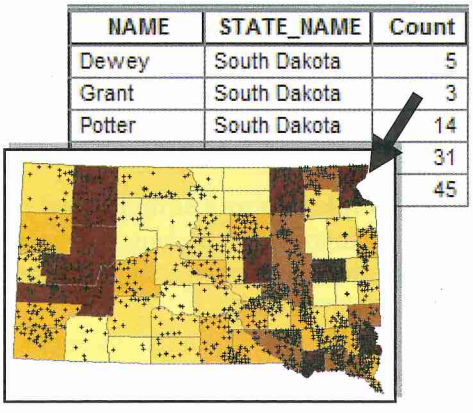


Fig. 9.12. Using a spatial join of schools to counties, one can create a graduated color map showing the number of schools in each county.

Setting up a spatial join

Performing the spatial join itself is a simple process. However, determining that a spatial join is required and identifying the destination table and the type of join can challenge beginners. This section presents a series of questions to be answered when setting up a spatial join to help produce the correct result. Too many novices simply try random combinations until they hit on the right one—this process is designed to give the right answer the first time.

When faced with a suspected spatial join problem, first make a simple sketch of the relationships between the layers to be joined. Then answer this series of questions, designed to lead to the correct solution:

What should the final output layer or table look like?

Which is the destination layer?

Should a distance join or an inside join be used?

What is the cardinality of the join?

Should a simple join or a summarized join be used?

Let us demonstrate this process by using the problem of estimating stream susceptibility to contamination from septic systems, as shown in Figure 9.9. Recall that each point represents a section and stores the total number of septic systems in the section. First sketch the problem and then go on to the questions.

What should the final output layer/table look like? The desired result is a layer of streams. In the attribute table, each stream must be assigned the total number of septic systems that are closer to it than to any other stream. Imagine the fields in the table, with the stream followed by the sum of the number of septic systems. Once the output is envisioned, it is usually easy to see which feature class is the destination layer.

Which is the destination layer? The features in the output layer are always the same features as those in the destination layer. If streams is chosen as the destination, then the output layer will contain streams. If the septics layer is chosen, then the output layer will contain septic points. Imagining our output table again, we see that what we really want is streams, each of which has septic information assigned to it. Thus, streams is the destination layer.

Should a distance join or an inside join be used? Since we're looking for septic systems closest to each stream, this is clearly a distance join.

What is the cardinality of the join? Assignment of cardinality depends on the destination layer, in this case streams, so consider the relationship between streams and the septic systems. Since one stream can have multiple septic points closer to it than to any other stream, this is a one-to-many cardinality.

Should a simple join or a summarized join be used? Since the cardinality is one-to-many, a simple join cannot work. Because the goal is to sum all the septic systems closest to the streams, rather than simply finding the single closest septic to the stream, a summarized join is indicated, with Sum as the statistic.

Now that the questions are answered, the problem setup becomes clear: We need to do a summarized distance join with streams as the destination layer and septics as the source layer. Even experienced users may find that following this suggested procedure when setting up a join makes it easier to find the right approach. In the next few examples, we will apply this process to set up and solve three different spatial join problems.

Problem 1

Number of earthquake deaths in congressional districts

Imagine that a representative from one of the congressional districts in California is sponsoring legislation to provide earthquake emergency planning funds and is looking for support for the bill. She is planning to throw a big party and invite all of the reps from districts with a significant number of earthquake deaths. She asks one staffer, who is a GIS specialist, to draft a list of names for the invitation list. The staffer might approach the problem as follows, using the data in the mgisdata folder.

STOP! Write the answers to the questions and then read on to see if you analyzed the problem correctly.

What should the output layer/table look like? _____

Which is the destination layer? _____

Should a distance join or an inside join be used? _____

What is the cardinality of this join? _____

Should a simple join or a summarized join be used? _____

The earthquake table has a state attribute but none for congressional district. The only way to associate the number of earthquakes with districts is to perform a spatial join. The goal is a table of districts with a field containing the number of earthquake deaths for each one. Thus, the districts will be the destination layer. The relationship between districts and earthquakes is potentially one-to-many, so a simple join is out of the question. Since the staffer wants to know the total deaths from quakes in each district, the summarize option will be the best, and the proper statistic is Sum. We will do a summarized inside join, with districts as the destination layer.

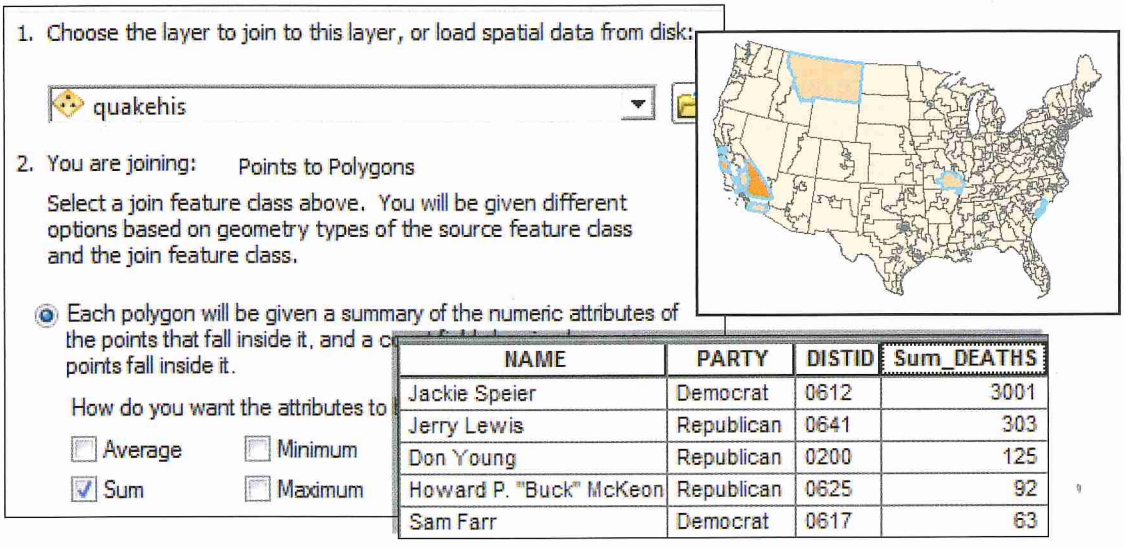


Fig. 9.13. Using a summarized spatial join to find the congressional districts that have had more than 10 earthquake-related deaths

Figure 9.13 shows the spatial join window filled out for this problem, as well as the resulting table. After performing the join, the staffer uses an attribute query to select all of the districts that have had 10 or more earthquake-related deaths. The table and the map show the selected records from the query—many of the districts are in California, as one might expect. Finding only 12 districts on the list and knowing that the boss wants a BIG party, the staffer then uses Select By Attributes to find all the districts that have had ANY earthquake deaths. This brings the total up to 29 districts. The guest list should be even bigger, so the staffer decides to use Select By Location to also select any districts that touch the districts with earthquake deaths (Fig. 9.14). This brings the guest list to 91, and the staffer can make up the invitations.

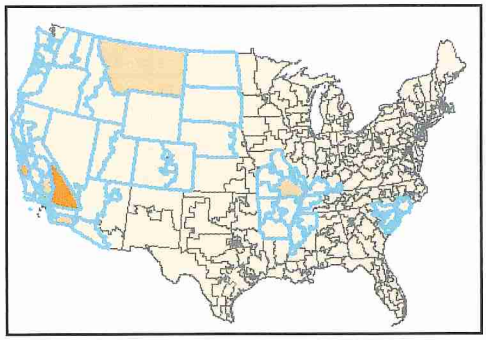


Fig. 9.14. Congressional districts with earthquake deaths or that touch a district with earthquake deaths

Problem 2

Pollution risk of rivers based on county populations

Imagine that Lindsey must do an analysis estimating the risk of pollution to rivers in the United States. She has no detailed information of the sources and types of pollutants, but she can make use of the observation that, generally speaking, large numbers of people are generally correlated with high risks of pollution. She decides to use the population of counties as a proxy for the pollution risk. For each river, she wants to find the total number of people living in counties that intersect the river.

STOP! Write the answers to the questions and then read on to see if you analyzed the problem correctly.

What should the output layer/table look like? _____

Which is the destination layer? _____

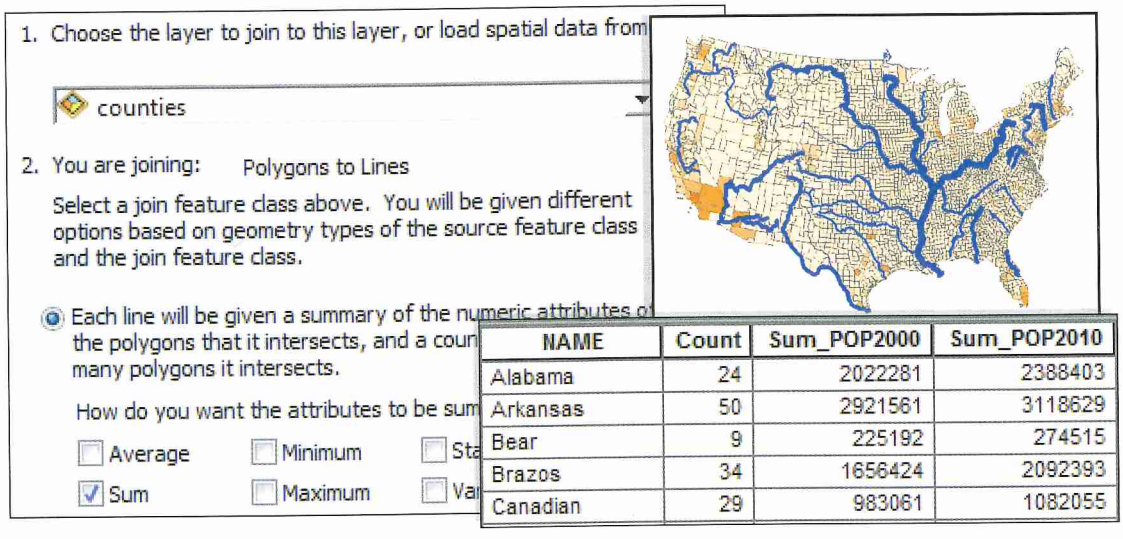
Should a distance join or an inside join be used? _____

What is the cardinality of this join? _____

Should a simple join or a summarized join be used? _____

The final output layer should contain rivers, with a table listing each river and the total population of counties adjacent to that river. Thus, rivers must be the destination layer. Since the counties must actually touch the river, this is an inside join rather than a distance join. The cardinality is one-to-many, since one river can have many counties touching it. An inside summarized join must be used with rivers as the destination layer, and the SUM statistic should be requested.

Figure 9.15 shows the spatial join window filled out for the join. The output table shows the river name, the number of counties adjacent to it, and the sum of each numeric field, including the POP2000 and POP2010 fields. The figure also shows a graduated symbol map based on the Sum_POP2010 field, in which the width of the river represents the total county population living adjacent to it. Notice that ALL of the numeric fields are summed during the join, not just the POP2000 field. One drawback to summarized spatial joins is that the chosen statistic is applied to all the fields, yielding potentially large attribute tables.



NAME	Count	Sum_POP2000	Sum_POP2010
Alabama	24	2022281	2388403
Arkansas	50	2921561	3118629
Bear	9	225192	274515
Brazos	34	1656424	2092393
Canadian	29	983061	1082055

Fig. 9.15. Estimating pollution risk for rivers based on adjacent county populations

Also note that the field names appear truncated. Because field names are limited to 13 characters, prefixing each with "Sum_" may crop other characters from the end. Notice that the POP2000 field has become Sum_POP200, the POP00_SQMI field has become Sum_POP00_, and so on. The shorter names may cause confusion, and it may be necessary to go back to the original table of the source layer to find out which field is which. The fields will be in the same order as the original. An alias can then be created, if desired, to avoid further confusion.

Problem 3

Closest volcano to a city

In this final example, imagine that a professor who specializes in volcanoes has built a web site about them. He would like to put a table on his web site so that schoolchildren all over the United States can enter the name of the city they live in and get an information page about the volcano that is closest to them. Since the professor knows nothing about GIS, his graduate student, Cody, gets to make the table.

STOP! Write the answers to the questions and then read on to see if you analyzed the problem correctly.

What should the output layer/table look like? _____

Which is the destination layer? _____

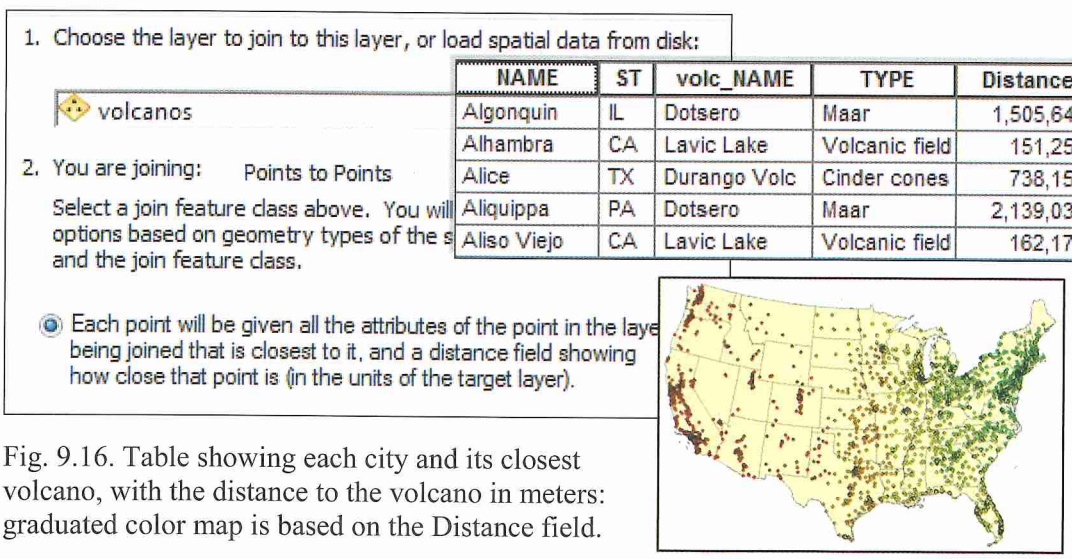
Should a distance join or an inside join be used? _____

What is the cardinality of this join? _____

Should a simple join or a summarized join be used? _____

The final goal is a cities layer with fields indicating the closest volcano and the distance to it. Thus, Cody realizes, Cities is the destination table, and a distance join to the volcanoes layer will provide the necessary information. Because the closest volcano is the target, this is a simple distance join (only one volcano can be the closest one).

Figure 9.16 shows the spatial join window settings to produce this analysis and the result of the spatial join. The table shows the city name, the state abbreviation, the volcano name, the place where the volcano is located, the volcano elevation and type, and the distance to the volcano.



NAME	ST	volc_NAME	TYPE	Distance
Algonquin	IL	Dotsero	Maar	1,505,644
Alhambra	CA	Lavic Lake	Volcanic field	151,255
Alice	TX	Durango Volc	Cinder cones	738,157
Aliquippa	PA	Dotsero	Maar	2,139,036
Aliso Viejo	CA	Lavic Lake	Volcanic field	162,176

Fig. 9.16. Table showing each city and its closest volcano, with the distance to the volcano in meters: graduated color map is based on the Distance field.

Notice the field name volcanoes_NAME in the output table. It happens that the Cities layer has a NAME field, and so does the volcanoes layer. The output table cannot have two fields with the same name, so during the join the repeated names are automatically converted into unique names. The Cities name field remains NAME, but the Volcanoes name field becomes NAME_1. In addition, an alias is created, volcanoes_NAME. It is the alias that is being displayed in the table, but if the table is set to show actual field names, then the NAME_1 field will be shown instead.

Also notice that the distance from Boone, Iowa, to the Dotsero volcano in Colorado is over one million. These units are clearly not miles or kilometers. In fact, they are meters, indicating that the coordinate system of the original data is stored with units of meters.

TIP: Always check the original coordinate system after a join to be sure you know the units for the distance values. Also, remember that distance joins should be performed only on projected coordinate systems that preserve distance, or the results will be invalid.

Summary

- A spatial join combines the records of two feature tables based on the location of the features. A new feature class is created by a spatial join.
- An inside join uses the criterion that one feature falls inside another or, in the case of points and lines, on top of each other. A distance join matches the destination layer feature to the record of the closest feature in the source layer. A distance field reporting the distance between the joined features is added to the table.
- A simple join may be used whenever the cardinality of the layers is one-to-one or many-to-one. A summarized join generates summary statistics for all the source features matching the destination features and is used when the cardinality is one-to-many or many-to-many.
- Four types of spatial joins exist based on the combination of the criterion and the cardinality of the relationship: simple inside joins, simple distance joins, summarized inside joins, and summarized distance joins.
- Distance measurements are reported in map units of the input layers.
- Distance joins should only be performed with layers having projected coordinate systems that do not distort distances. Using layers with a geographic coordinate system may yield incorrect results.
- In the joined table, fields with identical names will be renamed in the output file so that all field names are unique, such as NAME to NAME_1. Usually, the source field is renamed.
- Use the following series of questions to help you set up a spatial join properly:

  What should the output layer/table look like? _____

  Which is the destination layer? _____

  Should a distance join or an inside join be used? _____

  What is the cardinality of this join? _____

  Should a simple join or a summarized join be used? _____

Important Terms

cardinality	inside join	simple join	summarized join
distance join	logical consistency	spatial join	

Chapter Review Questions

1. What primary characteristic distinguishes a spatial join from an attribute join?
2. What two options may be used to handle one-to-many relationships in a spatial join?
3. If a polygon feature type is joined to a line layer, with the lines as the destination table, what will the feature type of the output layer be?
4. How many output fields will result if a summarized join is specified and a single statistic (e.g., Sum) is selected?
5. Why should distance joins always be performed on layers with a projected coordinate system? What kind of projection should be used?
6. What happens if the two input layers in a join each have a field with the same name?

For the following spatial join problems, answer the series of questions in the text and then state the type of join that should be used: simple inside, simple distance, summarized inside, or summarized distance.

7. Determine the number of parcels within each of Austin's watersheds.
8. Find the closest school for each house in a realtor's database.
9. Find the land use zoning type associated with each well in Atlanta.
10. Determine the number of counties and the total number of people served by each airport in the United States.

Mastering the Skills

Teaching Tutorial

The following examples provide step-by-step instructions for doing basic tasks and solving basic problems in ArcGIS. The steps you need to do are highlighted with an arrow ➔; follow them carefully. Click on the video number in the Video Index to view a demonstration of the steps.

Simple inside joins

The 'ex_9.mxd' map document is within the mgisdata\MapDocuments folder.

➔ Start ArcMap and open the map document ex_9.mxd.

➔ Use Save As to rename the document and remember to save frequently as you work.

Spatial joins produce new feature classes. We only need these for practice and don't want them to become mixed with the data in our permanent Austin geodatabase. So, for this lesson, we'll create a new file geodatabase to store the outputs.

1➔ Click the Catalog button or go to the Catalog tab if it is already docked in ArcMap.

1➔ Expand the Folder Connections entry and navigate to the mgisdata\Austin folder.

1➔ Right-click the Austin folder and choose New > File Geodatabase.

1➔ While it is highlighted, name the new geodatabase chap9results and click Enter.

TIP: When you see the word **STOP**, pause a moment and set up the problem using the questions presented in the Concepts section. Then read on to see if you analyzed the problem correctly.

- What should the output layer/table look like?
- Which is the destination layer?
- Should a distance join or an inside join be used?
- What is the cardinality of this join?
- Should a simple join or a summarized join be used?

The map currently shows the geology and water wells for Austin, TX. For each well, we would like to know the geological unit associated with it at the surface (i.e., for each well, we want to know the geological unit that the point falls inside). **STOP** and think it through.

The desired output is a table of wells containing a field with the geological unit. Therefore, the Wells layer is the destination. A well can fall into only one geology polygon, making it a one-to-one cardinality. Therefore, a simple inside spatial join is indicated.

2➔ Right-click the Wells layer and choose Joins and Relates > Join from the context menu.

2➔ In the top drop-down box, choose to *Join data from another layer based on spatial location* (Fig. 9.17).

2➔ Choose Geology as the layer to join.

2➔ Choose to join the point to the polygon that *it falls inside*.

2➔ Click the Browse button and change the *Save as type* to *File and Personal Geodatabase feature classes*.

2➔ Navigate inside the Austin\chap9results geodatabase and enter **wellgeology** as the name of the feature class to be created. Click Save and click OK.

The new feature class appears at the top of the Table of Contents.

3➔ Turn off the original Wells layer.

3➔ Right-click the wellgeology layer and choose Open Attribute Table.

3➔ Scroll to the right to find the fields from the Geology table.

TIP: The first field from the Geology feature class is named geology_OBJECTID. All fields to the right of it are from the Geology table. Look for the second OBJECTID after any spatial join to find where the appended information begins.

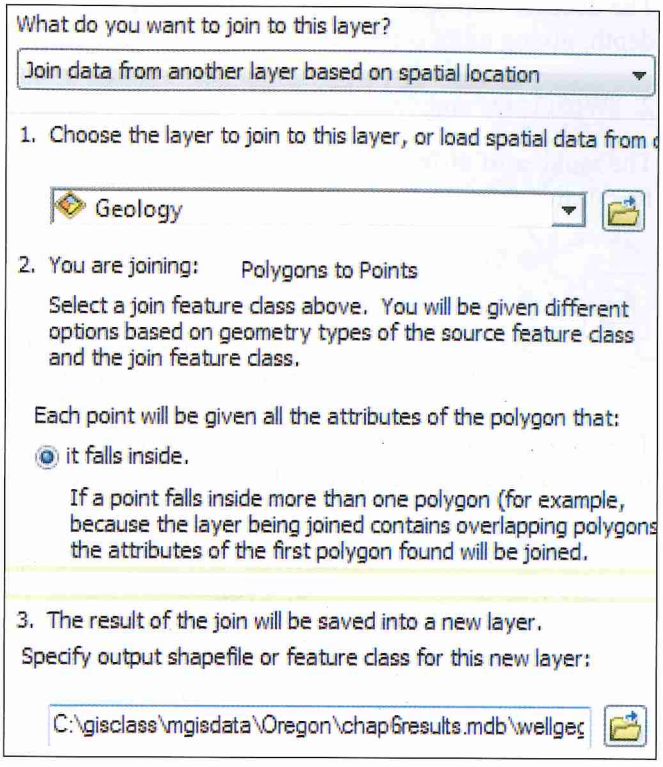


Fig. 9.17. The Join Data window for spatial joins

Locate the UNIT_NAME field at the very end of the new table, containing the geological unit. Let's find out how many of the wells are on limestone units.

3➔ Click the Select By Attributes button in the Table window and select the wells based on the expression UNIT_NAME LIKE '%Limestone%'. Close the selection window.

1. How many wells are situated on limestone? _____

4➔ Right-click the wellgeology layer and choose Zoom to Layer.

4➔ Expand the legend for the Geology layer and find the Glen Rose Limestone. Where does it outcrop on the map? (Try changing it color to one that stands out.)

4➔ Find the aquifer_code field from the wells part of the table and examine the entries. Look for codes containing "GLR".

The aquifer code field indicates the formation that supplies water to the well at an opening some depth underground. The GLR aquifer codes refer to the Glen Rose Limestone, which is a prominent aquifer (water-bearing rock) in this part of the country. Wells often get water from a different rock formation than is exposed at the surface. Let's find out how many of the wells ON the Glen Rose Limestone actually get water from the Glen Rose Limestone.

4➔ Open the Select By Attributes window.

4➔ Enter the new expression

UNIT_NAME = 'Glen Rose Limestone' AND aquifer_code LIKE '%GLR%'

The deepest well in this group lies within the Glen Rose Limestone both at the surface and at depth, giving us an estimate of the aquifer thickness. The well_depth field gives the depth in feet.

2. What is the approximate thickness of the Glen Rose Limestone near Austin? _____

The number of different fields resulting from this join makes the table unwieldy to navigate. Recall that you can easily turn fields on and off using the layer properties.

5➔ Open the properties for the wellgeology layer and click the Fields tab.
5➔ Click the *Turn all fields off* button.
5➔ Check these fields to turn them back on again: state_well_number, aquifer_code, well_depth, and UNIT_NAME. Click OK and examine the table.

The table is much easier to work with when it shows only the fields in which we are currently interested. Use this technique whenever you wish during this tutorial and afterwards.

6➔ Close the Table window and clear all selected features.
6➔ Turn off the wellgeology layer and turn on the Creeks layer.
6➔ Right-click the Creeks layer and choose Zoom to Layer.

The geological substrate of a creek affects its connection with groundwater. A creek on porous limestone will lose more water to the rock than a creek on clay. We assigned geology information to wells using a spatial join; we can also do it for creeks. **STOP** and think through the problem.

We want a table containing creeks with a field indicating the geological unit that the creek is on. Thus, Creeks is the destination feature class. A simple inside join is called for.

7➔ Right-click the Creeks layer and choose Joins and Relates > Join.
7➔ Set the source layer to Geology.
7➔ Use the simple join option. (*Each line will be given the attributes of the polygon that it falls completely inside.*)
7➔ Store the output feature class in the chap9results geodatabase and name it creekgeology. Click OK.

Right click on it, go into Properties, then Symbology tab.

8➔ Collapse the Geology layer legend and turn it off.
8➔ Open the Symbology properties of the new creekgeology layer and choose a Categories: Unique values map based on the UNIT_NAME field.
8➔ Click the Symbol heading to change properties for all symbols to a 2-pt.-thick line.
8➔ Choose a color scheme with dark, bold colors that will show up well. Click OK.

In the Table of Contents, notice that one of the symbols has a blank, or <Null>, value instead of a unit name.

9➔ Right-click the symbol with the blank and set it to a light gray color (Fig. 9.18).
9➔ Turn on the Geology layer again.

You're mapping nominal data by the visual variable of color! Notice that there are other options in the Symbology tab—it's similar to the 'Change Style' pane in ArcGIS Online

The gray creeks cross geological contacts. We assumed that each creek was within a single geological unit, but we can see that it is not true in all cases. When a creek crossed a boundary, no unit was joined to it, and it received a <Null> value in the attribute table for the geology attributes.

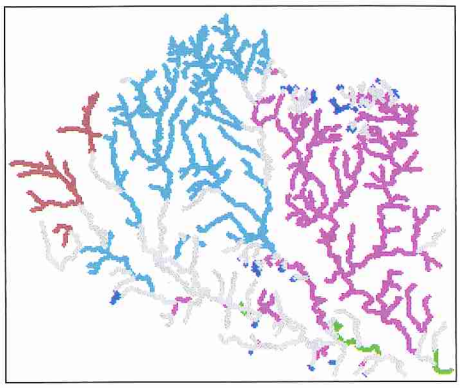


Fig. 9.18. Creeks displayed by geological unit

9➔ Open the creekgeology attribute table. Examine the fields, particularly the UNIT_NAME field containing the geology.

9➔ Right-click the UNIT_NAME field and choose Sort Ascending to see the <Null> values. Then close the Table window.

One weakness of inside spatial joins is that they join only when features fall completely inside other features. In Chapter 10, we will learn about the Intersect tool, which is similar to a join but can split the features when they cross. Let's take a sneak peek ahead.

10➔ Open the ArcToolbox > Analysis Tools > Overlay > Intersect tool.

10➔ Click the dropdown button for input features and select Creeks.

10➔ Click the dropdown button again and select Geology.

10➔ Name the output feature class creekgeolint and save it in chap9results.

11➔ Symbolize the creekgeolint layer as you did creekgeology. Compare the two layers and also examine the tables.

The creekgeolint layer has no <Null> values in its table. Creeks that crossed a geologic unit boundary have been split into pieces, and the geologic unit has been assigned to each piece.

A summarized inside join

A simple join is appropriate for a one-to-one or a many-to-one cardinality. A summarized join is needed for a one-to-many cardinality. Think about a watershed, the collection area for surface water. Runoff that comes from a watershed with many people is likely to have more pollutants than runoff from a watershed with fewer people. Let's analyze which watersheds in the Austin area are at greatest risk for polluted runoff.

12➔ Remove the creekgeology, creekgeolint, and the wellgeology layers from the map. They are still saved in the chap9results geodatabase if you want them later.

12➔ Turn off the Geology and Creeks layers.

12➔ Turn on the Watersheds and Block Population layers.

12➔ Right-click the Block Population layer and choose Zoom to Layer.

12➔ Open the table for the Block Population layer.

Block Population contains points representing the centroids of block groups, the smallest unit for which the Census Bureau summarizes population data. Each point represents numbers of people and households. We will use these points to determine the number of people in each watershed. **STOP** and think it through.

This time we want to have a list of watersheds with the total number of people in it. Watersheds is the destination layer. One watershed can have many block points in it, so the relationship is one-to-many. A summarized inside join must be used. Since we want to know the total number of people, we need to request the Sum statistic.

13➔ Close the Table window.

13➔ Right-click the Watersheds layer and choose Joins and Relates > Join.

13➔ Set Block Population as the layer to join to.

13➔ Choose the option to summarize the points inside the polygons and check the box for the Sum statistic.

13➔ Name the output feature class watershedpop and put it in the chap9results geodatabase. Click OK.

14➔ Drag the watershedpop layer just below the Block Population layer.

14➔ Open the watershedpop table and examine the fields on the far right.

14➔ Close the Table window.

The Count field records how many block points were found in the watershed. The fields prefixed with "Sum_" show the totals for each of the numeric fields in the source table. Sum_POP2000 tells us the total number of people living in each watershed.

Think about this as you do it, don't just click!

You're learning to make classified, standardized choropleth maps.

Look around the Symbology tab, too.

15➔ Open the symbology properties for the watershedpop layer and create a Quantities: Graduated color map based on the Sum_POP2000 field and the Jenks classification.

15➔ The population values are influenced by polygon area, so normalize the classification using the Shape_Area field.

15➔ Symbolize it with a monochromatic color ramp and click OK.

15➔ Name the layer Watershed Hazard.

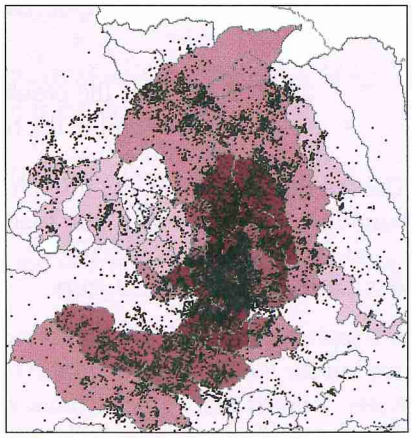


Fig. 9.19. Watershed pollution risk from population

The map should look similar to Figure 9.19. Note that the watersheds extend beyond the block population data. If this were a real project, you would need to address this issue by dropping polygons with incomplete data or expanding the block group data to the full extent of the watersheds.

16➔ Turn off the Watershed Hazard, Watersheds, and Block Population layers.

16➔ Turn on the Streets layer. Add the police_districts feature class from the mgisdata\Austin\Austin\Administrative feature dataset.

Imagine that the city has been reviewing its police department staffing levels. To help assess needs, it would be helpful to know the total length of streets for which each district is responsible. You have been given the task of finding this information. **STOP** and think it through.

You don't need to do this part in blue.

You don't need to do this part in blue.

You want to produce a list of the police districts, each of which has the total street length. Thus, the districts are the destination layer. Each district contains many streets, so the cardinality is one-to-many. A summarized inside join must be used, with Sum as the statistic. First, notice that many streets do not fall in any of the districts. You can reduce the processing time for the spatial join by first selecting only the streets that are within the districts.

17➔ Open the Select By Location window.

17➔ Set the target layer to Streets and the source layer to police_districts.

17➔ Set the spatial selection method to *are within*. Click OK.

Now when you do the spatial join, only the selected streets will be used, saving time. You will also have fewer streets with unassigned districts.

18➔ Right-click the police_districts layer and choose Joins and Relates > Join.

18➔ Set the source layer to Streets and choose the summarized option. Check the Sum statistic.

18➔ Place the result in chap9results and name it policestreets.

18➔ Click OK and wait. This join may take several minutes.

19➔ Clear the selection and turn off the Streets layer.

19➔ Open the policestreets table. The original Streets layer contained a field called MILES with the length of the street. Find the Sum_MILES field in the joined table.

19➔ Close the Table window.

20➔ Create a graduated color map of the policestreets layer based on the Sum_MILES field. For this map, it is best to classify the distances using a defined interval of 50 miles.

The downtown districts are smaller and have fewer miles of road than the outlying districts (Fig. 9.20). The next step might be to analyze the block population for each district to see if they are balanced in that way instead. We will leave that as an exercise, and move on.

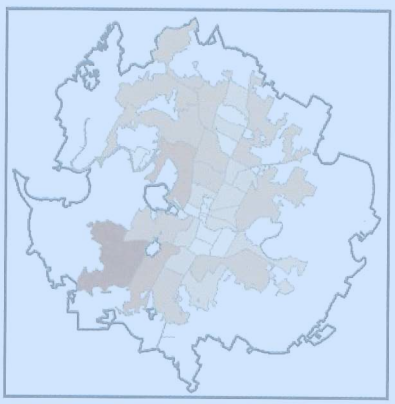


Fig. 9.20. Total road mileage each police district covers

21➔ Turn off the police_districts layer.

21➔ Remove the policestreets layer from the map document.

Simple distance joins

Distance joins combine records for features that are closest to each other. Consider the problem of post office usage. One might assume that, generally, people will go to the closest post office. For each street, it would be nice to know which post office to go to. **STOP** and think it through.

We want a list of streets with a field indicating the closest post office, so Streets is the destination layer. Since only one post office can be "closest," this is a one-to-one relationship. A simple distance join is appropriate.

22➔ Turn on the Streets and Post Office layers.

22➔ Right-click the Streets layer and choose Joins and Relates > Join.

22➔ Set Post Office as the layer to be joined.

22➔ Choose the simple join option, where *Each line will be given the attributes of the point that is closest to it.*

22➔ Name the output file **streetpost** and save it in chap9results. Click OK.

23➔ Open the streetpost attribute table and examine all of the fields. Find the field that contains the names of the different post offices.

23➔ Close the Table window.

3. Which field contains the post office names? _____

23➔ Create a unique values map of the streetpost layer based on the FACILITY_N field, so that each street is symbolized by the post office it is closest to. Use thick lines to see the colors better (Fig. 9.21).

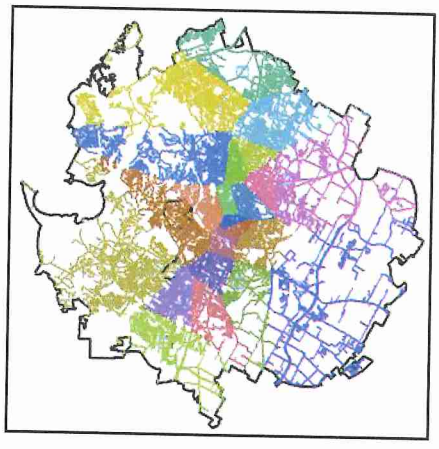


Fig. 9.21. Post office service areas

A distance join also creates a distance field. In this case it shows how far each street lies from the nearest post office (as the bird flies, not driving distance). The units will match whatever the storage units are for the feature class coordinate system.

4. What are the coordinate system and units for the streetpost feature class? _____

5. How many streets are more than two miles from a post office (as in Fig. 9.22)? _____

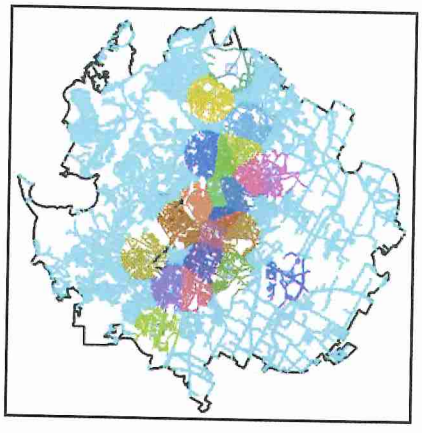


Fig. 9.22. Streets farther than two miles from a post office

24➔ Clear the selected features.

24➔ Create a graduated color map of the streetpost layer based on the Distance field, using a thicker line and a monochromatic color ramp (Fig. 9.23).

25➔ Remove the streetpost layer and turn off the Post Office layer.

25➔ Add the trailheads and restrooms feature classes from the Parks feature dataset in the Austin geodatabase.

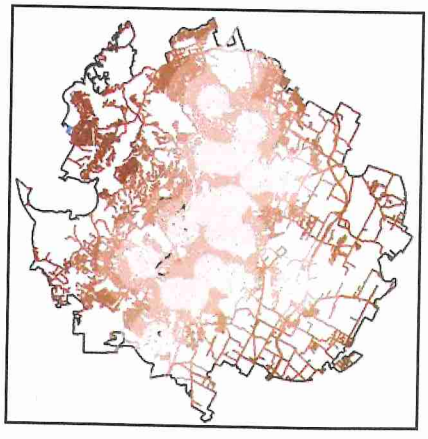


Fig. 9.23. Distance of streets from nearest post office

Before people start a hike, or after they finish, they like to find a restroom. Imagine that a hiking club wants to analyze the proximity of trailheads to restrooms and make

Stop here for now!

recommendations for adding more restrooms to serve hikers. **STOP** and think through the problem.

The club needs a list of trailheads that includes the distance of each trailhead to the closest restroom. Thus, trail_heads is the destination layer. There can be only one closest restroom to a trailhead, so the cardinality is one-to-one, and a simple distance join will suffice.

26➔ Right-click the trail_heads layer and choose Joins and Relates > Join.

26➔ Set restrooms as the layer to join.

26➔ Choose the simple join option to give each point the attributes of the closest point.

26➔ Name the output feature class trailrest and save it in chap9results. Click OK.

The club is most concerned about the trailheads that are more than 1000 feet from a restroom. You will create a map that will help the club identify optimum locations for new restrooms.

27➔ Open the trailrest table and examine the fields. Find the Distance field.

27➔ Use Select By Attributes to select the trailheads that are more than 1000 feet from a restroom. Close the Table window.

27➔ Right-click the trailrest layer and choose Selection > Create Layer From Selected Features. Name it Problem Trails.

27➔ Clear the selected features.

28➔ Add the parks and the trails feature classes from the Parks feature dataset.

28➔ Symbolize the parks in a light green color and the trails as a dark green thick line.

29➔ Click on the restrooms layer symbol to open the Symbol Selector.

29➔ Type restroom in the search box at the top and click the Search button. Choose the symbol you prefer and make it about 16 pt.

29➔ Open the restrooms layer properties, click the General tab, and set the minimum scale to 1:60,000.

30➔ Remove the trailrest and trail_heads layers.

30➔ Create a graduated color map of the Problem Trails layer based on distance from the restrooms. The values are skewed, so use the Jenks classification method. Make the symbol slightly larger so you can better see the colors (Fig. 9.24a).

30➔ Zoom in to an area with clusters of Problem Trails to test the map (Fig. 9.24b).

Let's save this work as a group layer file for future use by the club.

31➔ Click on the Problem Trails layer name to highlight it. Hold down the Ctrl-key and click the restrooms, parks, and trails layers.

31➔ Right-click one of the highlighted layers and choose Group.

31➔ Name the new group layer Restroom Map.

31➔ Right-click the Restroom Map group layer and choose Save As Layer File. Save it in the Austin folder as Restroom Map.lyr.

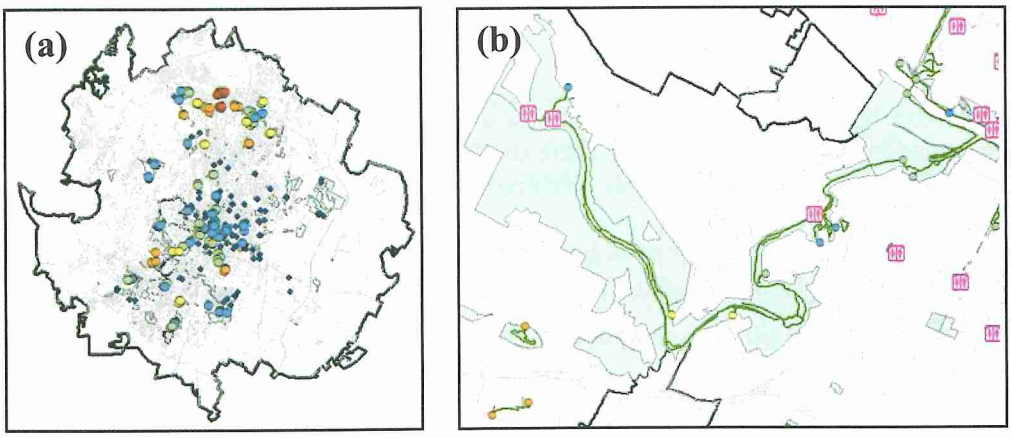


Fig. 9.24. Map showing trailheads more than 1000 feet from a restroom

32➔ Turn off the Restroom Map group layer and collapse it.

32➔ Right-click the City Limit layer and zoom to it.

Summarized distance joins

A simple distance join asks "Which feature is closest?" A summarized distance join is needed to address the question "How many features are closer to one location than they are to another?" Imagine that the Department of Recreation is applying for a grant to renovate its recreation centers. For guidance, they would like to know the potential population served by each center, making the assumption that most people will go to the closest center. The Block population data will provide the base data set for the analysis. **STOP** and think it through.

We want a list of the recreation centers with a field containing the total number of people who are closest to it. Centers is the destination layer, and the cardinality is one-to-many. A summarized distance join will be used.

33➔ Add the facilities feature class from the Facilities feature dataset in the Austin geodatabase.

33➔ Use Select By Attributes to select recreation centers using the expression FACILITY_T = 'RECREATION CENTER'.

33➔ Right-click the facilities layer and choose Selection > Create Layer from Selected Features. Name the layer Rec Centers.

33➔ Remove the facilities layer.

TIP: Creating intermediate selection layers is often a smart thing to do. It provides a visual test of your query, and it saves the selection in case a later mistake requires redoing the analysis. Give each selection layer a descriptive name, to avoid confusion later.

34➔ Right-click the Rec Centers layer and choose Joins and Relates > Join.

34➔ Set Block Population as the layer to join.

34➔ Choose the summarized option. You want to know the total number of people, so check the Sum statistic.

34➔ The default save folder was changed to Austin when you saved the Restroom Map layer file. Click the folder Browse button to change it back.

34➔ Change the Save as type to *File and Personal Geodatabase feature classes*.

34➔ Navigate inside the chap9results geodatabase. Name the output recpop and click Save and OK.

35➔ Open the recpop table and examine the Sum fields.

35➔ Examine the statistics for the Sum_POP2000 field.

6. What are the minimum, maximum, and average numbers of people served by the recreation centers? _____

Clearly there are large discrepancies in the potential usage of each center. What is the spatial distribution of this potential usage?

35➔ Close the Statistics and Table windows.

35➔ Create a graduated symbol map of the recpop layer based on the Sum_POP2000 field. Use Jenks Natural Breaks.

The map is interesting (Fig. 9.25). Small centers clustered in the downtown area appear to compete with one another, while the outlying centers draw large groups over larger distances. The department decides to adopt a strategy to turn the downtown cluster into special-interest centers that may draw more people, while expanding standard activities in the outlying centers. Of course, the final plan would take into account many more issues, such as parking, public transportation, and current facilities and usage.

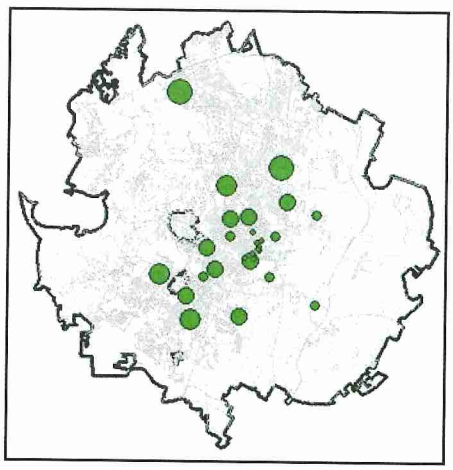


Fig. 9.25. Potential usage for recreation centers

Meanwhile, another group is concerned about the effects of population on stream pollution. The number of people living close to a stream impacts the likelihood for the stream to be affected by pesticides and other pollutants. This group wants to characterize the basic threat to each stream by summing the population closer to that stream than to any other. **STOP** and think it through.

We want a feature class of streams that contains a field with the total population close to it, so streams are the destination layer. The cardinality is one-to-many, so a summarized distance join with the Sum statistic is needed.

36➔ Add the named_creeks feature class from the Environmental feature dataset in the Austin geodatabase.

36➔ Turn off all layers except the City Limit and named_creeks layers.

36➔ Turn on the Block Population layer, but change the color to Gray 10%.

37➔ Right-click the named_creeks layer and choose Joins and Relates > Join.

37➔ Set the source layer to Block Population.

37➔ Choose the summarized join option. Fill the button to use the closest features and check the Sum statistic.

37➔ Name the output creekpop and save it in chap9results. Click OK.

38➔ Open the creekpop attribute table and examine the Sum_ fields.

38➔ Sort the Sum_POP2000 field in descending order.

7. Which three creeks have the highest risk based on total population?

_____

38➔ Rename the creekpop layer Population Load.

38➔ Create a graduated color map for the Population Load layer based on the Sum_POP2000 field. The data are fairly skewed, so use Jenks Natural Breaks.

38➔ Use a monchromatic color ramp and thick lines so the colors show (Fig. 9.26a).

It may occur to you that the longer creeks are naturally exposed to more people. You decide to create a hazard index based on the number of people per unit length of stream.

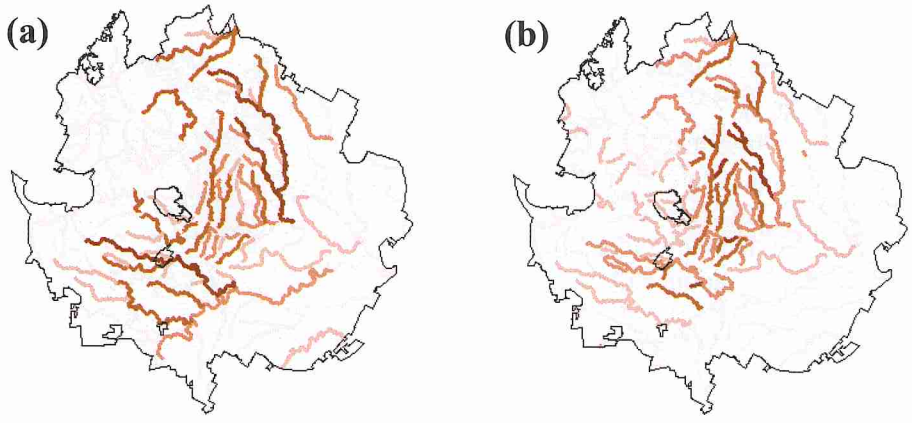


Fig. 9.26. Maps showing pollution hazard of Austin creeks based on (a) population load and (b) hazard index

39➔ Click the Table Options menu and choose Add Field.

39➔ Name the field HazIndex and set the field type to Float. Click OK.

39➔ Right-click the empty HazIndex field and choose Field Calculator. Enter the formula Sum_POP2000 / Shape_Length and click OK.

39➔ Examine the HazIndex field and sort in descending order.

8. Which three creeks have the highest risk based on the index?

_____

Notice that most of the creeks have an index near or below 1, but the top three are in the range of 3–6. Notice the Shape_Length field also. These high-index creeks have very short lengths, and they are probably scraps of creek remaining after they were clipped to the city boundary. These

high-index values are probably a fluke and should be eliminated from consideration. To avoid having them skew the map, you will exclude short creeks from the map.

40➔ Close the Table window.

40➔ Right-click the Population Load layer and choose Copy.

40➔ Right-click the Layers data frame name and choose Paste Layer(s). Rename the new layer Hazard Index.

41➔ Open the layer properties for the Hazard Index layer.

41➔ Click the Definition Query tab and enter the expression Shape_Length > 1000. Click Apply so that it takes effect before you set the symbols.

41➔ Click the Symbology tab and create a graduated color map based on the HazIndex field using the same symbols as the Population Load layer (Fig. 9.26b). Click OK.

As you can see, there are some differences between the two approaches for evaluating pollution hazards. (Some color differences may also be due to differences in the Jenks breakout of the different values.) Which model do you think might be a more realistic approach?

A hydrologist who specializes in surface water quality would probably question both models, because she knows that surface water runoff is controlled more by elevation than by distance, and that a watershed provides a better unit for summing the population impacts. You already did this using a summarized inside join.

42➔ Turn on the Watershed Hazard layer and open its properties.

42➔ On the Display tab, set the transparency to 50%.

42➔ Turn off the Block Population layer.

42➔ Compare the Population Load and Hazard Index maps visually against the Watershed Hazard map by turning the upper one on and off several times.

Which of the two creek-based methods appears to agree better with the watershed method? Overall, do the three methods agree on which watersheds and creeks bear closer scrutiny?

This analysis brings up an important consideration in GIS analysis. A problem may be approached several ways, using different models based on different assumptions. All three approaches had potential problems. In the watershed-based model, the population data set did not cover the entire watershed area. In the creek-based analysis, the fundamental assumption that distance was an appropriate predictor of influence is flawed, and short creeks resulting from data problems could have impacted the results. Each approach yielded similar, but not the same, results. Three important lessons should be gleaned from this example.

First, ***data issues and problems affect nearly all analysis procedures.*** You cannot mitigate all issues, but it is important to be aware of what they are, as well as what impacts they might have.

Second, ***the fundamental assumptions behind a model should be the best available.*** The watershed model had the most realistic assumption about water flow and is the best of the three models (or would be if the population data covered all the watersheds).

Third, ***do not push the results of a model beyond its limits.*** Even the watershed model neglects important variables. For example, industrial areas produce different amounts and types of

pollutants than residential or farming areas do. Stream flow impacts the ability of a stream to dilute pollutants. Although we mapped five hazard levels, whether the model can realistically separate the hazard into many categories with distinct boundaries is questionable.

In the long run, the best you can say about this analysis is that it highlights potential problem areas that would benefit from further scrutiny. Toward that goal, all three models performed equally well. This observation yields a fourth lesson, that ***the most detailed and accurate model is not always necessary to achieve the objective.***

Data quality issues with spatial joins

Now we will do one more polygon-to-polygon join to demonstrate some issues to consider when joining. We have a census tracts layer for Austin that has a county FIPS code, but no county name. We would like to do a spatial join of counties to the tracts in order to provide a field with the county name. This is a simple inside join with a many-to-one cardinality.

43➔ Collapse the legends of all of the layers in the data frame and turn them off.

43➔ Add the tracts and the counties feature classes from the Administrative feature dataset of the Austin geodatabase.

43➔ Put tracts above counties in the Table of Contents and zoom to the full extent.

44➔ Right-click the tracts layer and choose Joins and Relates > Join.

44➔ Set the join layer to counties.

44➔ Choose the simple join option.

44➔ Name the output `tractcounty` and place it in the chap9results geodatabase.

45➔ Change the counties symbol to a hollow shade with 2-pt. purple borders, and place it above the new tractcounty layer in the Table of Contents.

45➔ Create a unique values map for the tractcounty layer, using the joined COUNTY field containing the county names (it is near the end of the table).

Notice that the unique values legend has a <Null> entry, and many of the tracts are symbolized with this value. What happened?

46➔ Open the tractcounty table and scroll far to the right where the COUNTY field is.

46➔ Scroll down to examine all the rows in the table.

You can see many <Null> fields in the table. These tracts were not matched to a county at all. This fact is puzzling, because, from the map, the tracts appear inside the counties and should have been matched with one. However, the map shows that the affected tracts all share a boundary with the county (Fig. 9.27).

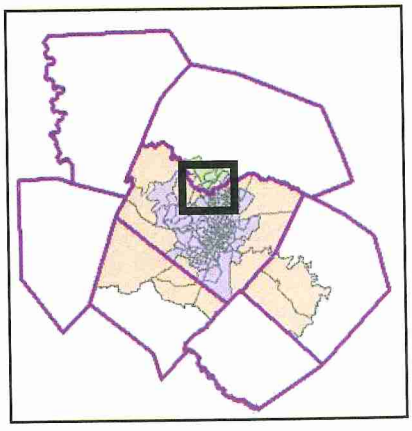


Fig. 9.27. The orange tracts received no match during the join.

46➔ Close the Table window and zoom in to the area shown in the black box in Figure 9.27.

46➔ Switch to the List by Selection view in the Table of Contents and make tractcounty the only selectable layer.

46➔ Click the Select Features by Rectangle tool and click on one of the tracts that share the county border. Click several more.

It becomes apparent that the tract boundary and the county boundary are not exactly the same (Fig. 9.28). In the real world, a tract always falls inside a county. In this GIS data set, the tracts and counties came from two different sources, and as a result their boundaries do not coincide. To be joined, the tract must fall within the county. Even the small discrepancies seen here are sufficient to prevent the county from being matched to the tract during the join.

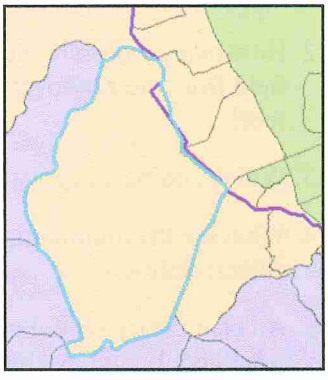


Fig. 9.28. Part of this tract lies outside the county.

How well the GIS data features reflect relationships in the real world is called **logical consistency**. In this case, the data sets are not logically consistent because the data features with mismatched boundaries do not reflect the real-world boundaries, which are identical.

TIP: Always keep in mind that data accuracy and logical consistency issues may have an impact on an analysis. Be alert for potential problems.

This is the end of the tutorial.

➔ Close ArcMap. You can save your changes.

Exercises

Use the data in the mgisdata\Austin\Austin geodatabase to answer the following questions.

1. Give the name of the watershed in Austin that contains the most wells. How many wells does it contain? Which watersheds have the deepest and shallowest average well depths?

2. How many wells are there in the WATERFRONT zoning code in Austin? Use the O_NAME field from the zoneoverlays feature class. How many of the wells have depths less than 100 feet?

3. Which zoning category (O_NAME) in Austin contains the most wells?

4. What are the minimum, maximum, and average distances from barbeque (bbq) pits to the closest restroom?

5. An elementary school pool fun day is planned. If each school goes to the closest pool to it, which pool will have the most schools attending? Create a map that would be helpful to planners in reassigning schools to less crowded pools. (**Note:** You must export the elementary school selection to a new feature class before joining, because the distance join does not honor the selected set. This may be a bug.)

6. Examine the table from Exercise 5 closely, and you will find a problem with your initial analysis. What is it? What would you need to change to get a better result?

7. Which post office potentially serves the greatest number of people, based on the block group population data? Which one serves the least? (**Note:** The post office locations are one of the types of information in the facilities feature class.)

Use the data in the mgisdata\Oregon\oregon geodatabase to answer the following questions.

8. Which city in Oregon is farthest from an airport? What is the distance in *kilometers*?

9. Assuming that an airport's service area includes all of the cities that are closer to it than to any other airport, determine how many cities each airport serves. Which airport serves the most cities? How many cities? Which serves the most people? How many people? Why does the Sum_POP2000 field contain negative values?

10. A boating club would like to know which parks in Oregon have the best access to lakes. Find the number of lakes and total lake area in each park. Which park(s) have the most lakes and how many? Which park has the greatest area of lakes? Examine parks with only one lake. The sum of the area fields for these lakes keeps repeating the same few numbers. Explain why. How is the Columbia River affecting this analysis?

Challenge Problem

Imagine that a national body recommends the minimal staffing levels for a police district as one officer for every 1000 people plus one officer for every 25 miles of road. The police department has asked you to calculate the minimum recommended staffing levels for each district. Create a table showing the largest 15 to 20 districts with the district name, the population, the miles of road, and the number of officers. Place the table on a supporting map layout (screen capture is the easiest method), along with a graduated color map of population and labels for the staff numbers.