From the Content Archives of:

Data Science For Dummies ED 1 & 2

Making Maps from Spatial Data

In This Chapter

* Working with spatial databases, data formats, map projections, and coordinate systems in GIS

* Analyzing spatial data with proximity, overlay, and reclassification methods

* Using QGIS to add and display spatial data

Advanced statistical methods are great tools when you need to make predictions from spatial datasets, and everyone knows that data visualization design can help you present your findings in the most effective way possible. That's all fine and dandy, but wouldn't it be great if you could combine the two approaches?

I'm here to tell you that it can be done. The key to putting the two together involves envisioning how one could map spatial data — *spatial data visualization,* in other words. Whether you choose to use a proprietary or open-source application, the simplest way to make maps from spatial datasets is to use Geographic Information Systems (GIS) software to help you do the job. This chapter introduces the basics of GIS and how you can use it to analyze and manipulate spatial datasets.

<tip>

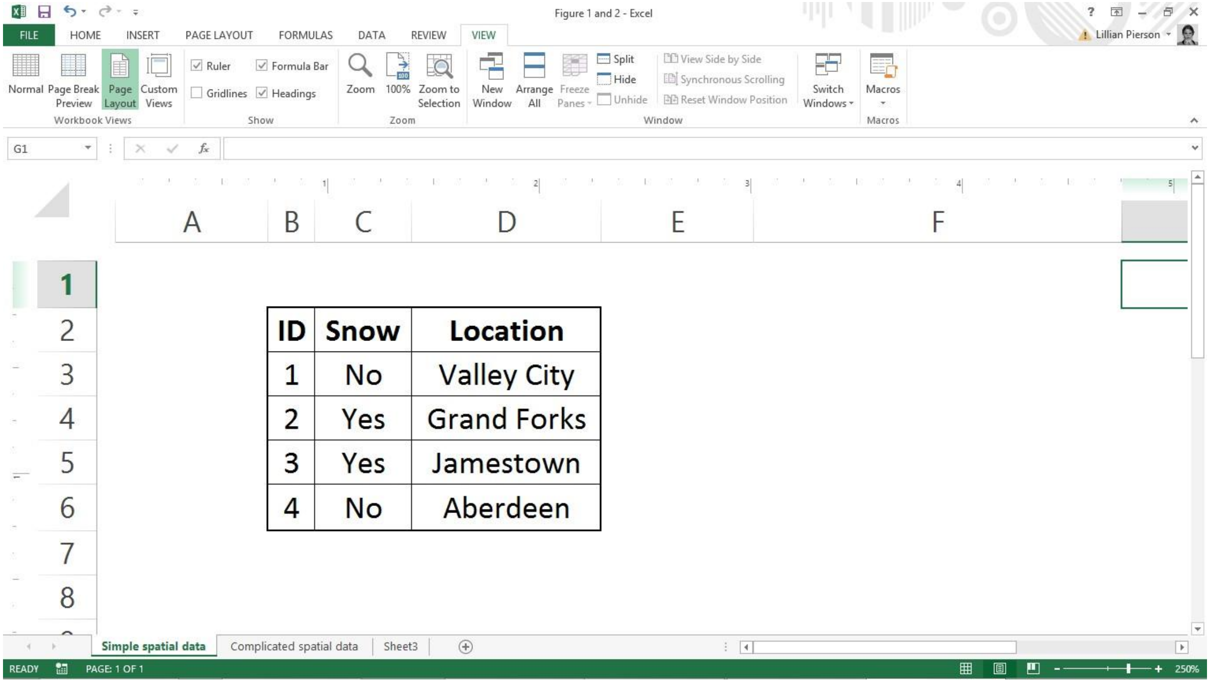
The proprietary GIS software, ESRI ArcGIS for Desktop, is the most widely used mapmaking application. It can be purchased from the ESRI website (at www.esri.com/software/arcgis/arcgis-for-desktop). But if you don't have the money to invest in this solution, you can use open-source QGIS (at www.qgis.org) to accomplish the same goals. GRASS GIS (at http://grass.osgeo.org) is another good open-source alternative to proprietary ESRI products. In practice, all these software applications are simply referred to as *GIS*.

Getting into the Basics of GIS

People use GIS for all sorts of purposes. Some simply want to make beautiful maps, and others could not care less about aesthetics and are primarily interested in using GIS to help them make sense of significant patterns in their spatial data. Whether you're a cartographer or a statistician, GIS offers a little bit for everyone. In the following sections, I cover all the basic concepts you need to know about GIS so that you can get started making maps from your spatial data.

To get a handle on some basic concepts, imagine that you have a dataset that captures information on snow events. When most people think of snow events, they may think of making a snowman, or of scary avalanches, or of snowskiing. When a GIS person thinks of a snow event, however, she more likely thinks about snowfall rates, snow accumulation, or in what city it snowed the most. A spatial dataset about snow might provide just this kind of information.

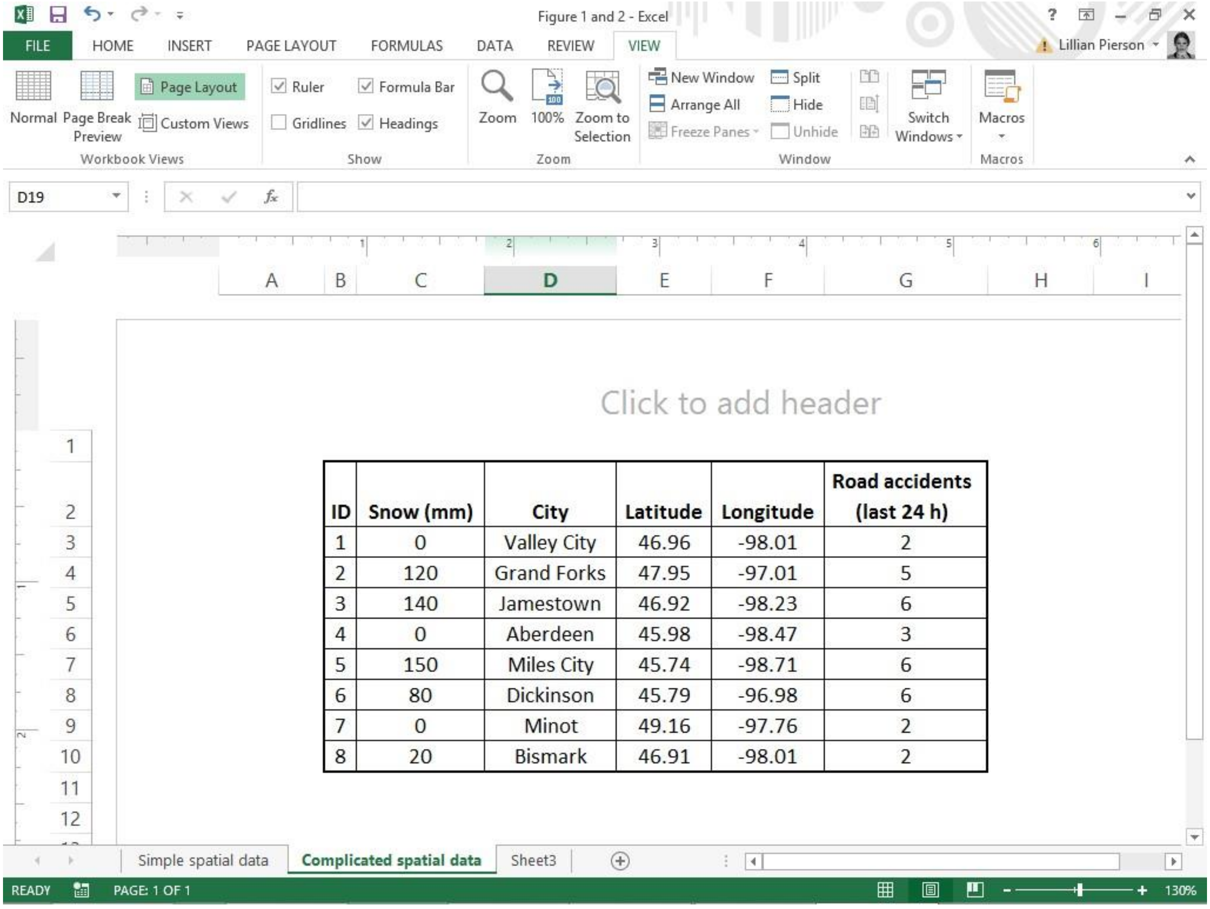
Check out the simple snow dataset shown in Figure 13-1. Although this table provides only a small amount of information, you can use the data in this table to make a simple map that shows what cities have snow and what cities don't.



ID	Snow	Location
1	No	Valley City
2	Yes	Grand Forks
3	Yes	Jamestown
4	No	Aberdeen

Figure 13-1: A sample of basic spatial data.

Although you can use this data to make a simple map, you need to have your location data in a numerical format if you want to go deeper into GIS analysis. Imagine that you go back to the table from Figure 13-1 and add three columns of data — one column for latitudinal coordinates, one column for longitudinal coordinates, and one column for number of road accidents that occur at the locations where these coordinates intersect. Figure 13-2 shows what I have in mind.



ID	Snow (mm)	City	Latitude	Longitude	Road accidents (last 24 h)
1	0	Valley City	46.96	-98.01	2
2	120	Grand Forks	47.95	-97.01	5
3	140	Jamestown	46.92	-98.23	6
4	0	Aberdeen	45.98	-98.47	3
5	150	Miles City	45.74	-98.71	6
6	80	Dickinson	45.79	-96.98	6
7	0	Minot	49.16	-97.76	2
8	20	Bismark	46.91	-98.01	2

Figure 13-2: Spatial data described through coordinates and additional attributes.

When you have spatial data coordinates that specify position and location, you can use GIS to store, manipulate, and perform analysis on large volumes of data. For the snow example, you could use GIS to calculate how much snow has fallen and accumulated in each city or to determine the cities where it's particularly dangerous to drive during snow events.

To do all that neat stuff, you need to bone up on some core GIS concepts that are central to helping you understand spatial databases, GIS file formats, map projections, and coordinate systems. The following sections help you accomplish that task.

Spatial databases

The main purpose of a spatial database is to store, manage, and manipulate attribute, location, and geometric data for all records in a feature's database. With respect to GIS, an *attribute* is a class of fact that describes a feature, *location* describes the feature's location on Earth, and *geometric data* describes the feature's *geometry type* — either a point, a line, or a polygon.

Imagine that you want to make a map of all Dunkin' Donuts restaurants that also sell Baskin-Robbins ice cream. The feature you're mapping is "Dunkin' Donuts restaurants," the attribute is "Baskin-Robbins Vendor? (Y/N)," the location fields tell you where these restaurants are located, each store is represented by its own *record* in the database, and the geometric data tells you that these restaurants must be represented by points on the map.

<remember>

A spatial database is similar to a plain relational database, but in addition to storing data on qualitative and quantitative attributes, spatial databases store data about physical location and feature geometry type. Every record in a spatial database is stored with numeric coordinates that represent where that record occurs on a map and each feature is represented by only one of these three geometry types:

* Point
* Line
* Polygon

Whether you want to calculate the distance between two places on a map or determine the area of a particular piece of land, you can use spatial database querying to quickly and easily make automated spatial calculations on entire sets of records at one time. Going one step further, you can use spatial databases to perform almost all the same types of calculations on — and manipulations of — attribute data that you can in a plain relational database system.

File formats in GIS

To facilitate different types of analysis and outputs, GIS accommodates two main file formats: raster and vector. Since these are the main two file format types used in GIS, both proprietary and open-source GIS applications have been specifically designed to support each.

Raster data is broken up and plotted out along a 2-dimensional grid structure so that each grid cell gets its own attribute value. (See Figure 13-3.) Although most people know that rasters are used to store image data in digital photographs, few people know that the raster file format is useful for storing spatial data as well.

(original graphic which was remade by Wiley for publishing – source: unknown)

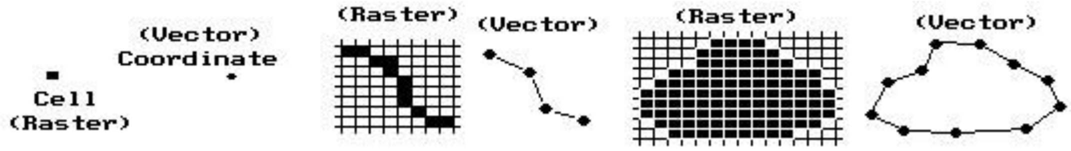


Figure 13-3: Raster and vector representations of different geometric features used in GIS.

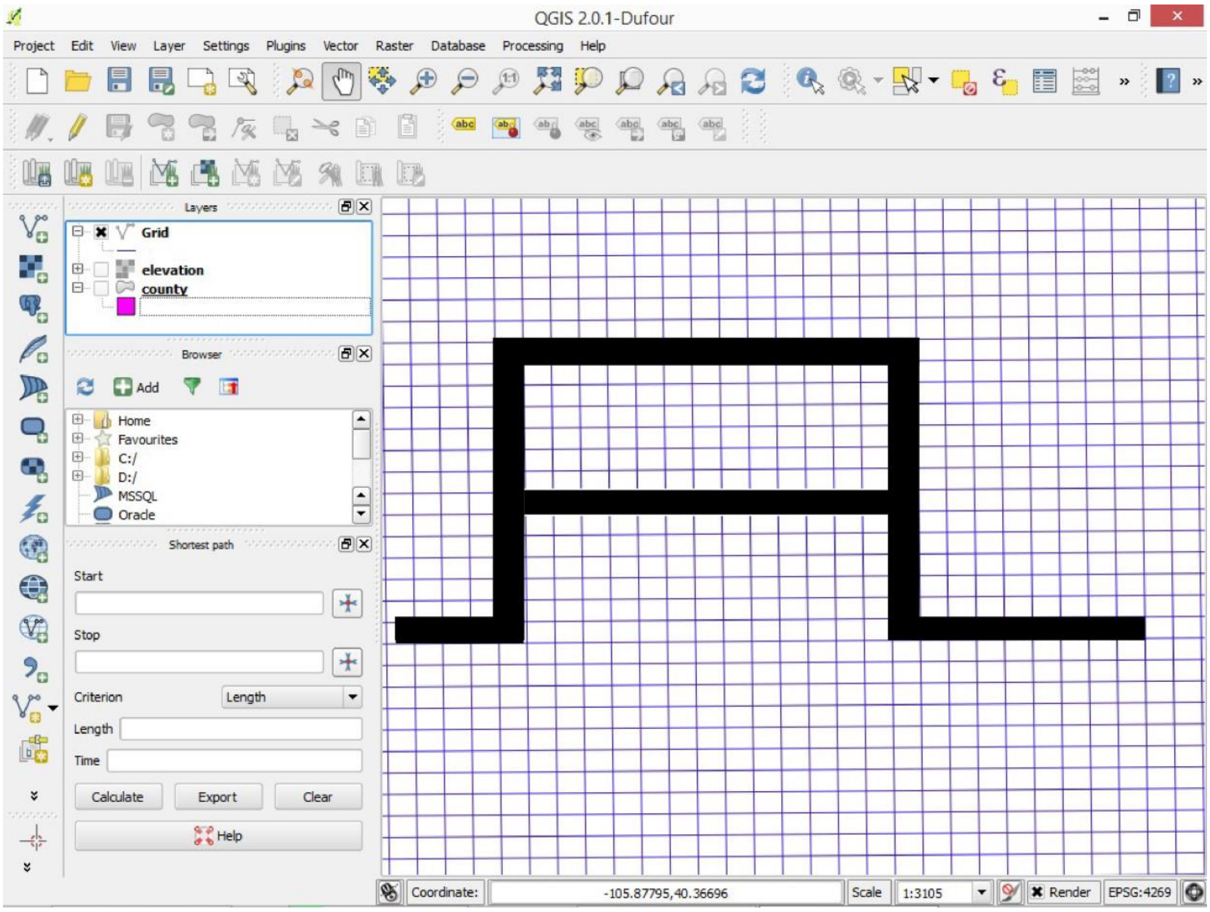
<remember>

Raster files can be used to store data for only one attribute at a time. In GIS, data for a single attribute is stored and plotted in a 2-dimensional grid, where the horizontal dimension represents longitude and the vertical dimension represents latitude. Digital photographs and Doppler weather radar maps are two common examples of raster files in the modern world.

Vector format files, on the other hand, store data as either points, lines, or polygons on a map. (Refer to Figure 13-3.) Point features are stored as single point records plotted in geographic space, whereas line and polygon features are stored as a series of vertices that comprise each record plotted in geographic space. For data in vector format, GIS can easily handle tens of attributes for each record stored in the spatial database. Google Maps, modern digital graphics, and engineering computer-aided design (CAD) drawings are some prime examples of vector graphics at use in the real world.

To conceptualize the raster versus vector idea, imagine that you have some graphing paper and a pencil and you want to draw a map of a street cul-de-sac that's in your neighborhood. You can draw it as a series of polygons — one representing the area covered by the street and the others representing the parcels that abut the street. Or, you can fill in the squares of the graph paper, one after the other, until you cover all the areas with one, single multicolored surface.

Vector format data is like drawing the street and parcels as a set of separate *polygons*. Raster data format is like making one *surface* by coloring the entire area around the cul-de-sac so that all street areas and the adjoining parcel areas are covered in their own, representative color. The difference between the two methods is shown in Figure 13-4.



Criterion: Length

Coordinate: -105.87795,40.36696

Scale 1:3105

EPSG:4269

Figure 13-4: A street and neighborhood represented as vector polygons and as a raster surface.

If you use GIS to create maps that show municipal boundaries, land cover, roads, attractions, or any other distinct spatial features, as shown in Figure 13-5, this type of spatial data is best stored in the vector file format. If you need to perform complex spatial analysis of multiple attributes for each feature in your dataset, keep your data in vector format. Vector data covers only the spatial areas where each discrete feature from your dataset is located on Earth. But with vector data, you get a lot of options on what attributes of that feature you want to analyze or display on a map.

(original graphic which was remade by Wiley for publishing – source: unknown)

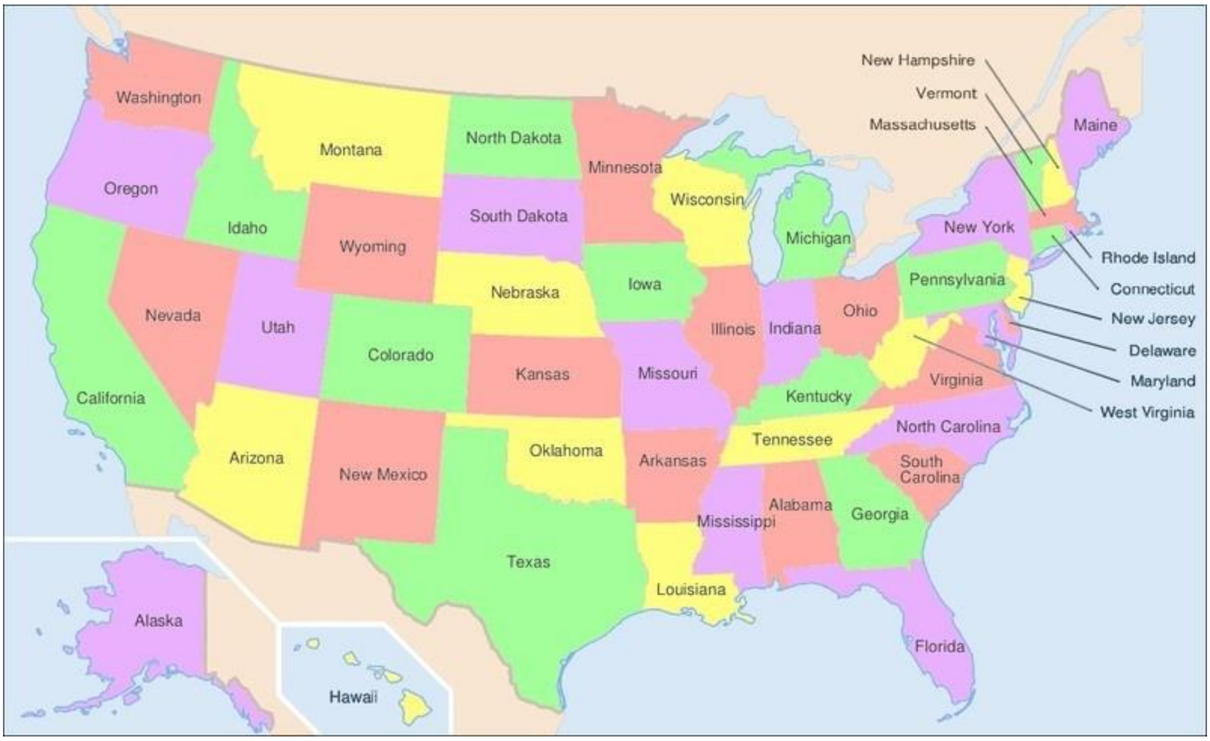
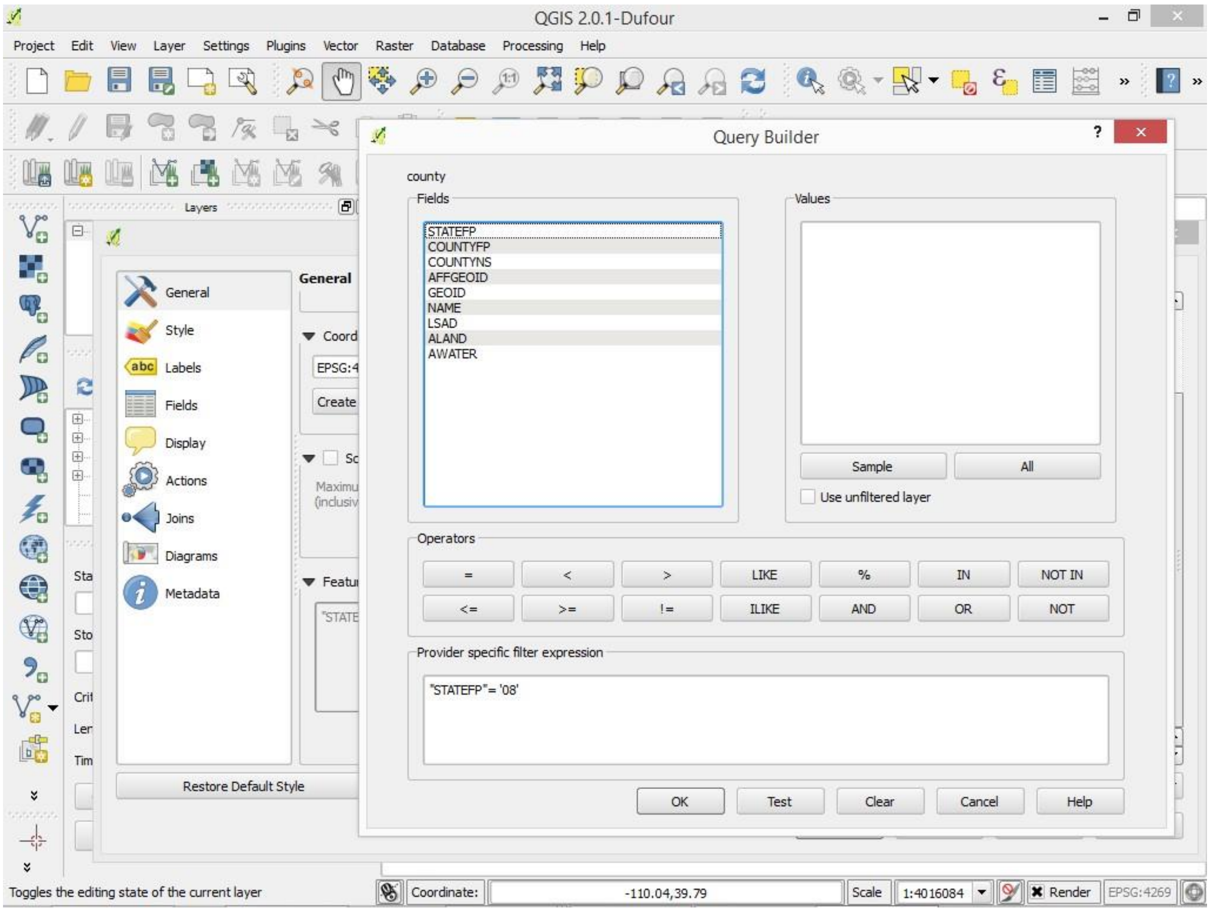


Figure 13-5: An address location map, represented as vector format points and polygons.

The easiest way to analyze several attributes (that could be derived from one or several features) that spatially overlap one another over a particular piece of land is to put your data into raster format. Because a raster file can represent only one attribute at a time, you'd layer several rasters on top of each other to see how the overlapping attributes compare in a fixed geographic region. While you can do a similar type of spatial overlap comparison using vector files, raster files will give you a full and comprehensive coverage for each set of attribute values across an entire study area.

For example, to quantify volume data across a fixed area of land, raster files are definitely the way to go. Consider the snow example again. Your attribute in this example is snow height. Given that your raster data provides all height data for the snow (on a pixel-by-pixel basis) in a particular fixed region, you can use that to calculate the volume of snow on the ground in that area: Simply multiply the area of each pixel by the difference between the average snow surface height and the average ground elevation at that location. To find the area of snow that has accumulated in a fixed area, sum up the volume of snow that has fallen in each pixel of that area, as shown in Figure 13-6.



county

Fields: STATEFP, COUNTYFP, COUNTYNS, AFFGEOID, GEOID, NAME, LSAD, ALAND, AWATER

Provider specific filter expression

```
"STATEFP"= '08'
```

Toggles the editing state of the current layer

Coordinate: -110.04,39.79 Scale 1:4016084 EPSG:4269

Figure 13-6: Interpolated surface of snow depth represented in a raster with low resolution.

<remember>

When you work with spatial data that’s in vector format, you’re focusing on *features*. You’re doing things like drawing separate line features, cutting existing features, or performing a buffering analysis to get some determination about features that are within a certain proximity of the feature you’re studying. When you work with spatial data that’s in raster format, you’re focusing on *surfaces*. You’re working with a raster surface that covers the entire spatial area you’re studying and describes the quantities, intensities, and changes in value of one attribute across an entire study area.

It’s possible to convert a vector feature to a raster surface — but, you can convert only one attribute at a time. Imagine that you have a vector file that represents gas stations with “Leaking Underground Storage Tanks,” represented as points on a map. The attribute table for this layer has data on the following four attributes: “Year Tank Was Installed,” “Year Leak Was Detected,” “Tank Depth,” and “Contaminant Concentrations.” When you convert all this data from vector to raster, you get four separate raster files, one for each attribute. The vector point format is converted to a raster surface that covers the entire study area and displays the attribute values, or lack thereof, on a pixel-by-pixel basis.

Map projections and coordinate systems

Map projections and coordinate systems give GIS a way to accurately represent a round Earth on a flat surface, translating Earth’s arced 3-dimensional geometry into flat 2-dimensional geometry.

Projections and coordinate systems *project* spatial data. That is to say, they forecast and predict accurate spatial positions and geographic scale, depending on where those features are located on Earth. Although projection and coordinate systems are able to project most features rather accurately, they don't offer a one-size-fits-all solution. If features in one region are projected perfectly at scale, features in another region are inevitably projected with at least a slight amount of distortion. This distortion is sort of like looking at things through a magnifying glass — you can see the object in the center of the lens accurately and clearly, but the objects on the outer edge of the lens always appear distorted. No matter where you move the magnifying glass, this fact remains unchanged. Similarly, you can't represent *all* features of a rounded world accurately and to-scale on a flat map.

In GIS, the trick to getting around this distortion problem is to narrow your study area, focus on only a small geographic region, and use the map projection or coordinate system that's most accurate for this region.

A coordinate system is a referencing system that is used to *define* a feature's location on Earth. There are two types of coordinate systems:

- **Projected:** Also called *map projection*, a projected coordinate system is a mathematical algorithm you can use to transform the location of features on a round Earth to equivalent positions represented on a flat surface instead. The three common projection types are cylindrical, conical, and planar.
- **Geographic:** A coordinate system that uses sets of numbers and/or letters to define every location on Earth. In geographic coordinate systems, location is often represented by latitude/longitude, decimal degrees, or degrees-minutes-seconds (if you're familiar with old-fashioned surveying nomenclature).

Figure 13-7 shows these three types, in all their glory.

(original graphic which was remade by Wiley for publishing – source: SAP)

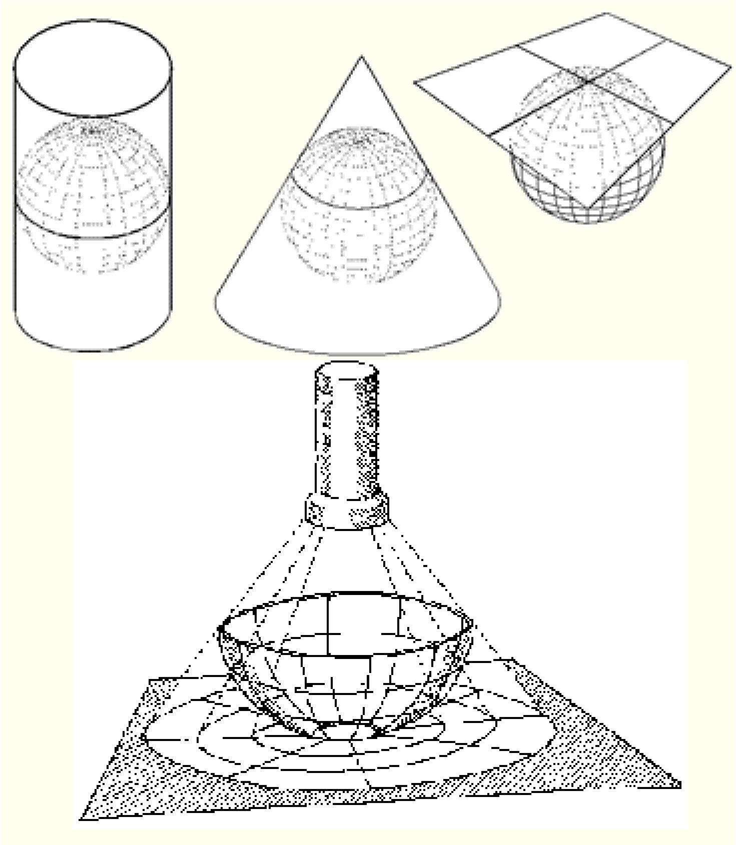


Figure 13-7: Three common projection types (left to right): cylindrical, conical, and planar.

Now that you know what types of coordinate systems are out there, it's time to take a look at how you'd make practical use of them. This is the easy part! In almost all cases, when you import a spatial dataset into GIS, it comes in with its own predefined coordinate system. The GIS software then adopts that coordinate system and assigns it to the entire project. When you add additional datasets to that project in the future, they may be using that same coordinate system or an alternative one. In cases where the new data is coming in with a coordinate system that's different from that of the project, the GIS software transforms the incoming data so that it is represented correctly on the map.

As an example of how all of this works in practice, to determine how much agricultural land has been contaminated during a tanker spill at the Mississippi Delta, you import a spatial dataset named Contaminated Land. The Contaminated Land file already has a predefined coordinate system — State Plane Coordinate System Mississippi, West MS_W 4376 2302. When you import the dataset, GIS automatically detects its coordinate system, assigns that coordinate system to the project you've started, and transforms any subsequently added spatial datasets so that they come in with correct scale and positioning. It's that easy!

<tip>

Information about a dataset's default map projection and coordinate system is stored in its metadata description. The default map projection and coordinate system are the fundamental reference systems from which you can re-project the dataset for your specific needs.

Analyzing Spatial Data

After you've imported your data, it's time to get into spatial data analysis. In the following sections, you find out how to use various querying, buffering, overlay, and classifying methods to extract valuable information from your spatial dataset.

Querying spatial data

In GIS, you can query spatial data in two ways: attribute querying and spatial querying. *Attribute querying* is just what it sounds like: You use this querying method when you want to summarize, extract, manipulate, sort, or group database records according to relevant attribute values. If you want to make sense of your data by creating order from its attribute values, use attribute querying.

Spatial querying, on the other hand, is all about querying data records according to their physical location in space. Spatial querying is based solely on the location of the feature and has nothing to do with the feature's attribute values. If your goal is to make sense of your data based on its physical location, use spatial querying.

<tip>
Learning to quickly and fluidly switch between attribute and spatial querying can help you to quickly make sense of complex problems in GIS. A situation where this is true would be if you have a spatial point dataset that represents disease risk. If you want to find the average disease risk of people over the age of 50 who live within a 2-mile distance of a major power center, you could use spatial and attribute querying to quickly generate some results. The first step would be to use spatial querying to isolate data points so that you're analyzing only those people who live within the 2-mile radius. From this reduced dataset, you'd next use attribute querying to isolate the records of people who are over the age of 50, and then perform a quick mathematical operation to get the average value for disease risk from that subset.

<tip>
You can either query the spatial database directly using SQL statements or use the simple built-in interfaces to query your data for you. Sometimes the quickest way to query results is to write the SQL statement yourself, but if your query is simple, you might as well make your life easy and use the point-and-click interface instead.

Referring back to the snow example, if you want to generate a list of all cities where snow depth was greater than 100mm, you simply use attribute querying to select all records that have a snow value that's greater than 100mm. But if you decide that you want to generate a list of cities with more than 100mm of snow that are located within 100 miles of Grand Forks, you'd use both an attribute and a spatial query.

Buffering and proximity functions

Within a GIS project, you can select or extract spatial features based on their physical *proximity,* or nearness, to a point, line, or polygon by using buffering and proximity functions. Buffering and proximity functions are fundamental, basic spatial querying methods.

Proximity analysis is a spatial querying operation you can use to select and extract features that are within a user-defined distance from your target feature. You can use proximity analysis to calculate distances between features or to calculate the shortest route in a network. *Buffering* is a proximity operation you can use to select and extract spatial features that are within a user-defined distance of your target feature. Figure 13-8 shows a schematic of a Buffer Zone Y that encompasses all areas within distance d of a target Polygon X. You can use Buffer Zone Y to isolate, extract, and analyze all spatial features within the d distance of Polygon X.

(original graphic which was remade by Wiley for publishing – source: unknown)

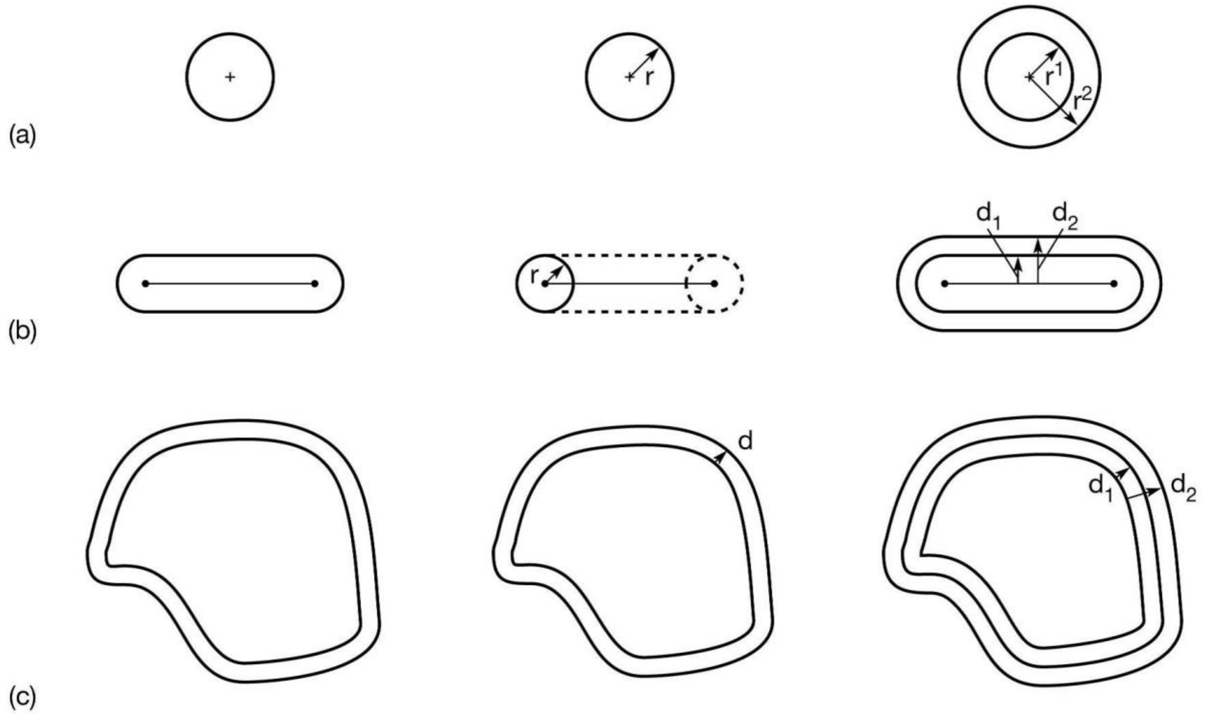


Figure 13-8: Buffered features at two different distances.

Using layer overlay analysis

One of the most powerful features of a GIS platform is its capability to overlay and derive meaning from multiple layers of data. By using *layer overlay analysis,* you can apply multiple operations to multiple layers of data that overlap the same spatial area.

Union, intersection, non-intersection, and *subtraction* are a few fundamental overlay operations. *Union* operations combine the total area of all features being overlain, whereas *intersection* operations retain only the areas of overlap between the features being overlain. *Non-intersection* operations are the reverse of intersection operations — they represent the areas of non-overlap between the features being overlain. Lastly, you can use a *subtraction* operation to subtract an area from one feature based on the area of other features that overlay it. I know this all sounds rather obscure, but it's not so bad. Take a look at Figure 13-9 to see how these operations work.

(original graphic which was remade by Wiley for publishing – source: Desktop GIS)

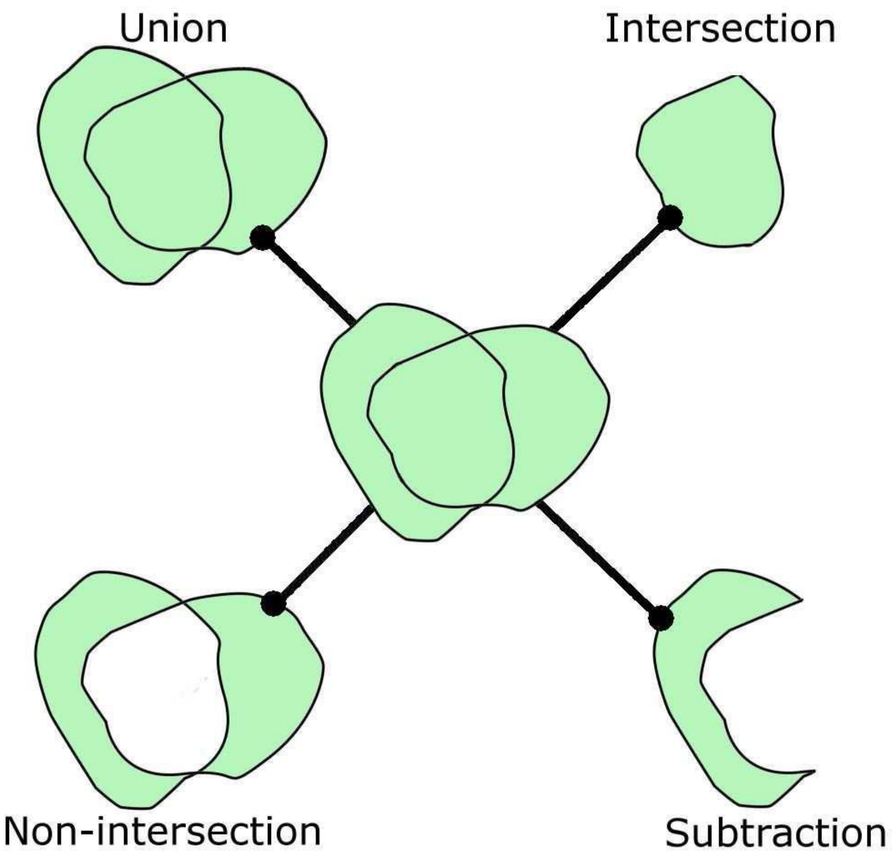


Figure 13-9: Simple operations applied on overlain features.

Overlay analysis is commonly used in suitability studies to determine which sites are best suited to support a particular function or type of business. For example, if you have to plan a new town settlement, you can use GIS to overlay spatial data layers and analyze the land's proximity to potable water sources, suitable terrain, suitable soil type, bodies of water, and so on. By overlaying these layers, you generate a map that shows which regions are more or less suitable to support the planned settlement.

<remember>

Vector format data is often a bit too big and clunky for complex overlay analyses. To reduce computing requirements, consider converting your vector data to raster data and then using overlay operations to make sense of the layers. This type of overlay analysis is called *raster algebra.*

Reclassifying spatial data

In GIS, *reclassification* is the act of changing or reclassifying the values of cells in a raster file, or the values of an attribute in a vector file. Although you can use layer overlay operations to analyze more than one layer at a time, you have to perform reclassification on a layer-by-layer basis. You can use reclassification if you want to reassign a new set of values to existing cells (in rasters) or attribute values (in vectors), but you need the newly assigned values to be proportional to, and consistent with, the current values and groupings of those cells or attribute values. Reclassification is applied to vector or raster data layers that generally represent attributes of Earth's surface (in other words, elevation, temperature, land cover type, soil type, and so on).

To fully grasp the concept of reclassifying data in a raster layer, imagine a raster surface where every cell is assigned a depth of snow. Simply by creating new groupings of depth ranges, you could easily reclassify this source data to uncover new snow depth patterns in the study area.

To illustrate the concept of vector layer reclassification, consider that you have a vector polygon layer that depicts land cover across your study area. In this layer, you have polygons that represent lakes, rivers, agricultural land, forests, grassland, and so on. Now imagine that you want to know where only the water and vegetation are located in this study area. You can simply repackage your map by reclassifying all Lake and River polygons as Water and all Agricultural, Forest, and Grassland polygons as Vegetation. With this reclassification, you can identify water from vegetation areas without needing to give the map more than a sweeping glance.

Getting Started with Open-Source QGIS

Earlier sections in this chapter focus on the basic concepts involved in GIS and spatial data analysis. The following sections let you finally get your hands dirty. I show you how to set up your interface, add data, and specify display settings in QGIS. To follow along, you must first start by downloading and installing QGIS (from http://qgis.org/en/site/forusers/index.html) and then download the QGIS Tutorial Data from the GitHub repository for this course (at https://github.com/BigDataGal/Data-Science-for-Dummies/).

Getting to know the QGIS interface

The main window of QGIS contains a lot of toolbars and menus, as shown in Figure 13-10. The toolbar on the far left is used to add data. You can add vector layers, raster layers, comma-delimited tables, and several other data types. The toolbar at the top contains many tools that allow you to navigate through the map you're creating. You can use the two toolbars below the topmost toolbar to manipulate and analyze data.

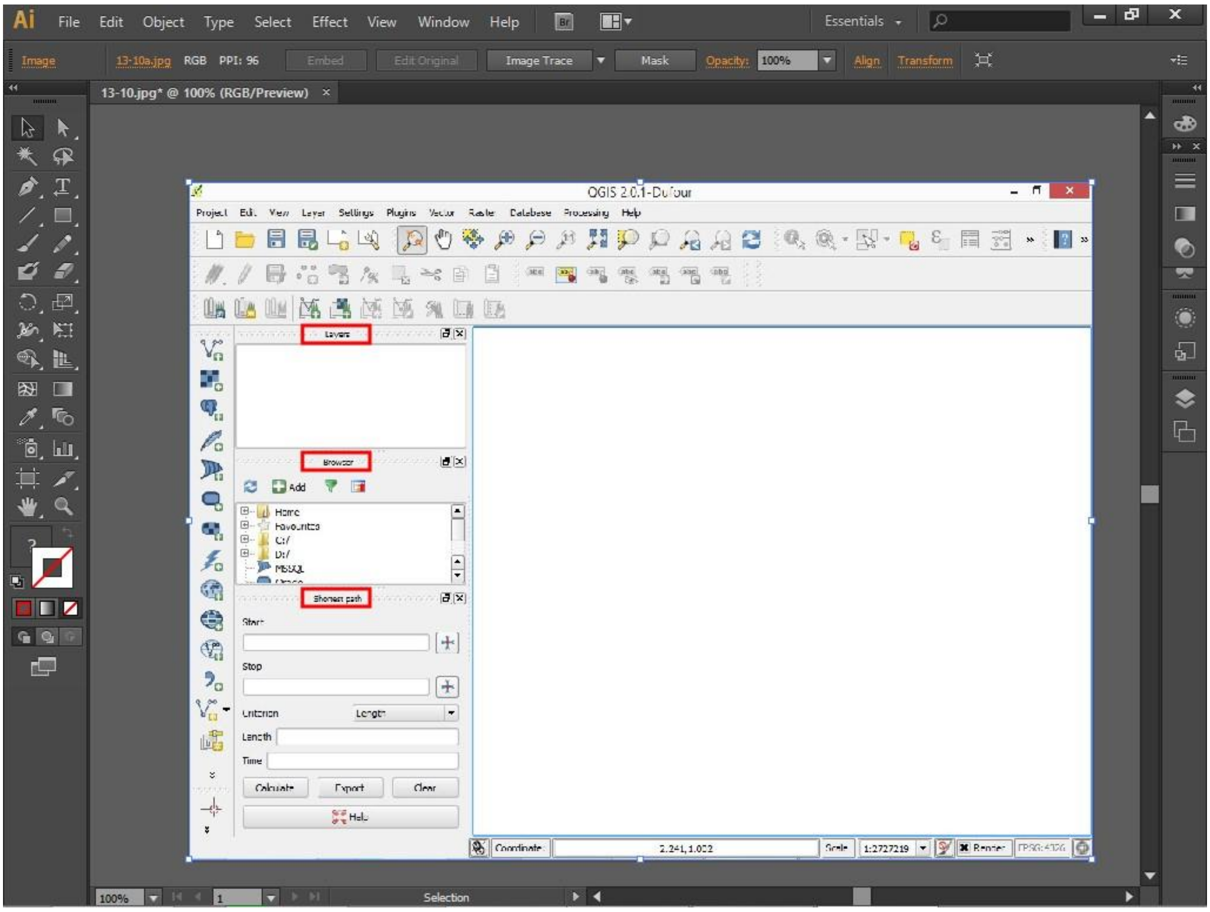


Figure 13-10: The default QGIS setup.

These three embedded windows run down the left side of the main window:

- **Browser:** Allows you to browse through your files and add data
- **Layers:** Shows you what layers are active in your map
- **Shortest Path:** Calculates the shortest path between two points on a map

You won't use the Browser or Shortest Path window in this chapter, so you can close those windows by clicking the X that appears in the top-right corner of each window.

Your screen should now look something like Figure 13-11.

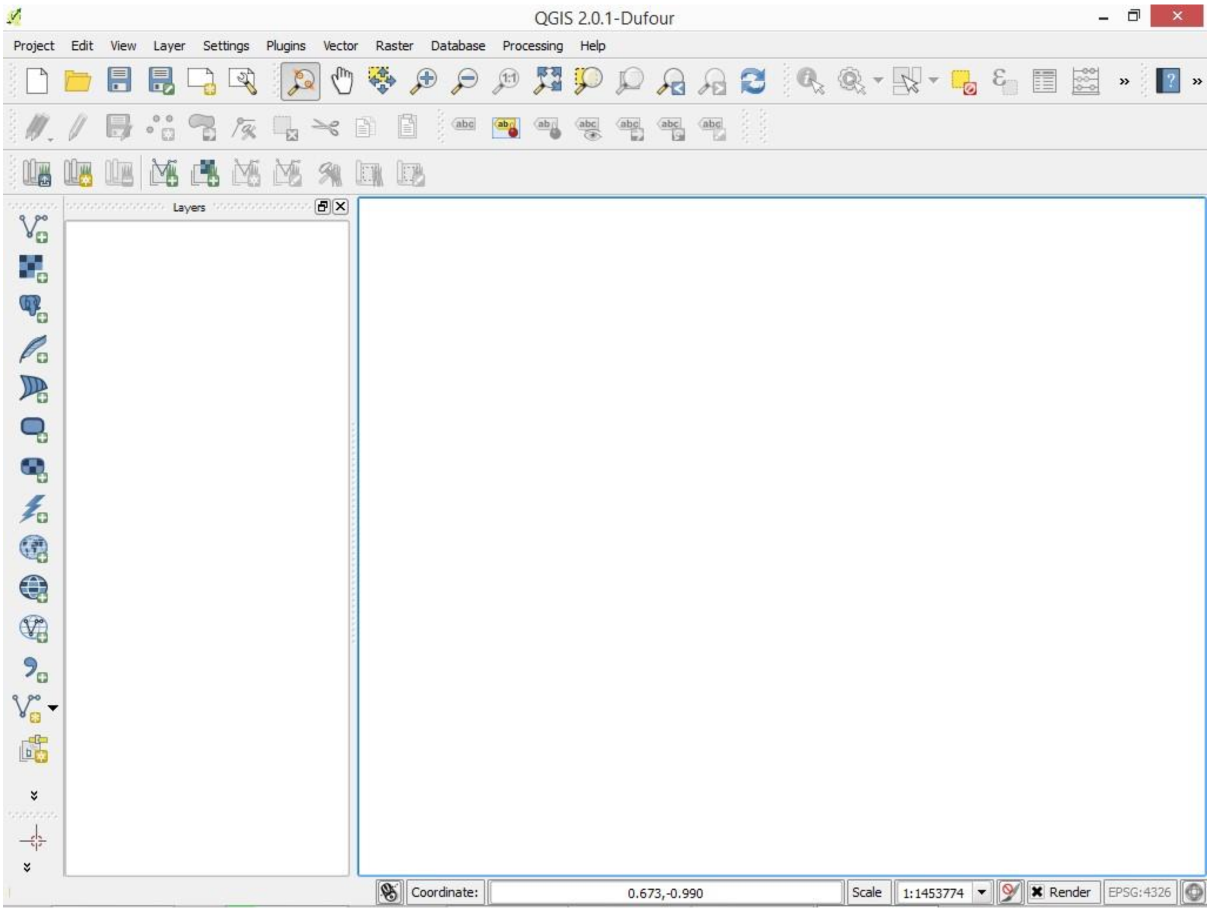


Figure 13-11: Your new QGIS setup.

Adding a vector layer in QGIS

To continue your exploration of the QGIS interface, add to your map a vector layer containing the borders for all counties in the United States by following these steps:

1. **Click the Add Vector Layer icon on the toolbar on the left of your screen.**

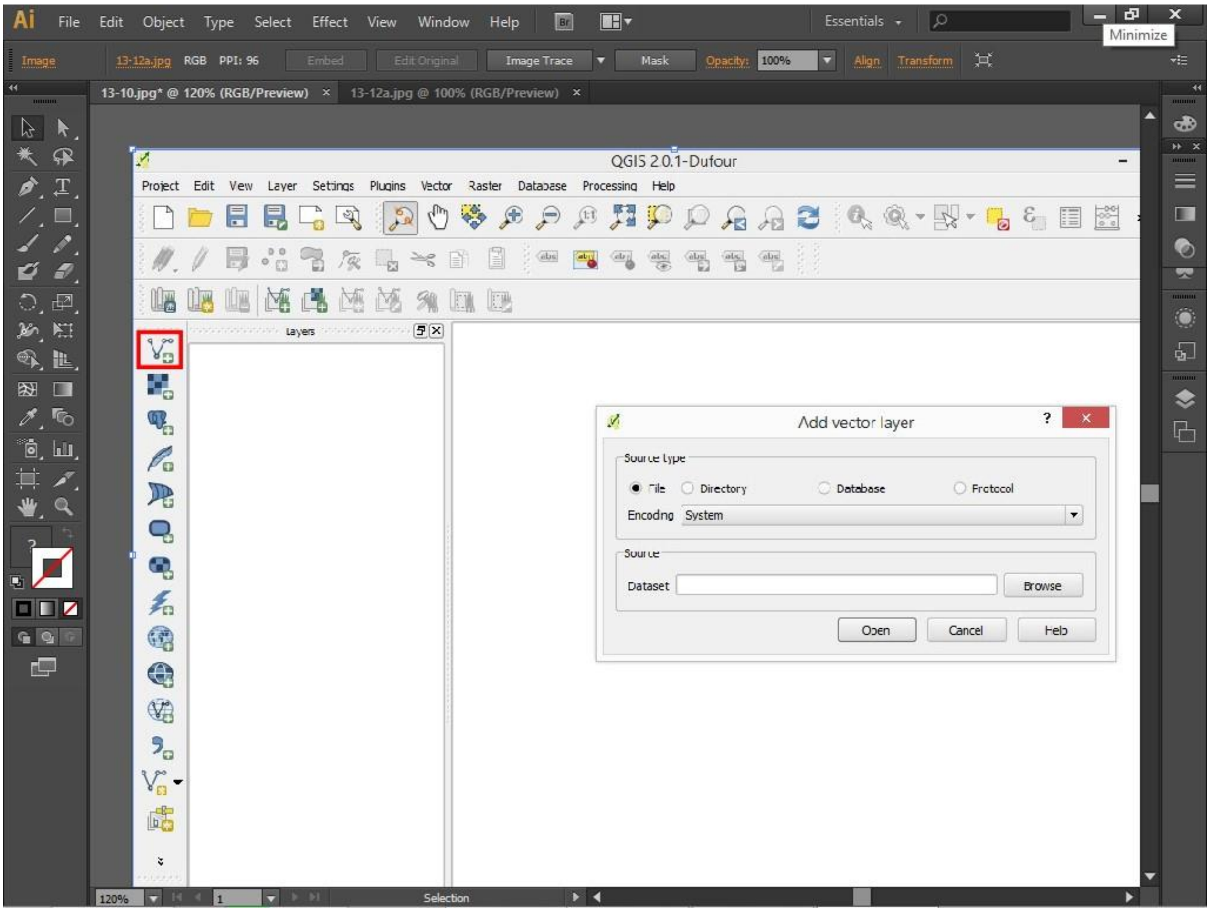
   The Add Vector Layer dialog box appears onscreen, as shown in Figure 13-12.

2. **Click the Add Vector Layer dialog box's Browse button.**

3. **In the Open an OGR Supported Vector Layer dialog box that appears, navigate to the folder where you choose to store the GIS data that you downloaded for this tutorial.**

4. **Choose the `county.shp` file, click OK, and then click Open.**

   A layer named County appears on the Layers menu, as shown in Figure 13-13.


Ai File Edit Object Type Select Effect View Window Help
Essentials
Minimize
Image 13-12a.jpg RGB PPI: 96 Embed Edit Original Image Trace Mask Opacity: 100% Align Transform
13-10.jpg* @ 120% (RGB/Preview)
13-12a.jpg @ 100% (RGB/Preview)
QGIS 2.0.1-Dufour
Project Edit View Layer Settings Plugins Vector Raster Database Processing Help
Layers
120%
Selection

Add vector layer

Source type

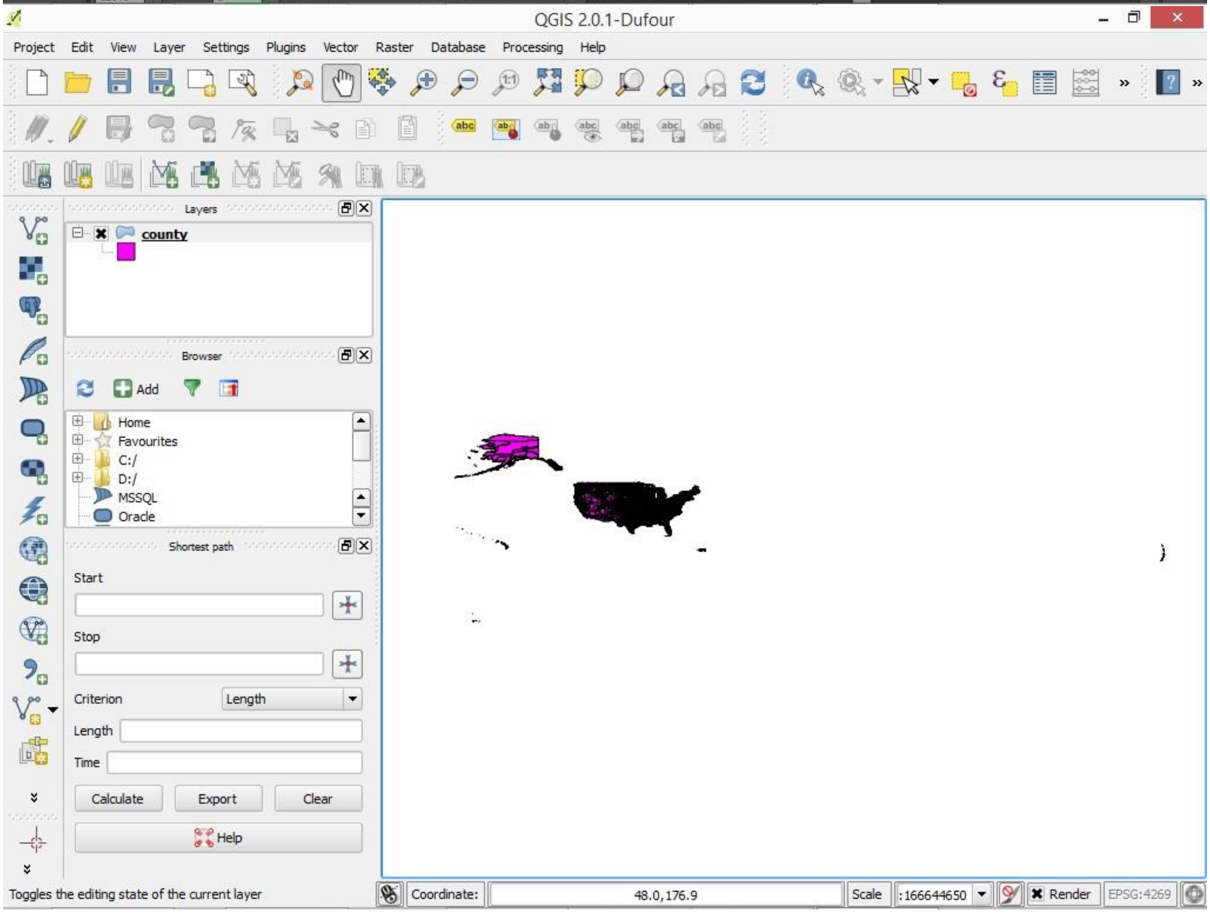
● File ○ Directory ○ Database ○ Protocol

Encoding: System

Source

Dataset: Browse

Open Cancel Help



Toggles the editing state of the current layer

Coordinate: 48.0,176.9 Scale :166644650 Render EPSG:4269

Figure 13-12: Adding a vector layer to QGIS.

Figure 13-13: A layer added into QGIS.

Displaying data in QGIS

This `county.shp` file (a vector file) displays all counties in the United States. All these polygons have Attribute data connected to and stored in the dataset's Attribute table. To see what I mean, take a look at the kind of information this table contains. Follow these steps:

1. **Right-click the County layer in the Layers window and choose Open Attribute Table from the pop-up menu that appears.**

   The Attribute Table window containing all the data appears, as shown in Figure 13-14.

<remember>

Each record in this table represents a single polygon. Every record has its own row, and each attribute has its own column. Within the QGIS Layer Properties settings, you can set up your record attributes so that they display different colors, are grouped in certain ways, and do a lot of other nifty things.

The attribute `STATEFP` contains a unique number for each state. You can use this number to do things like represent all counties in the same state with the same color. The attribute `ALAND` represents the size of each county. You can use the data that belongs to this attribute category to do things like assign darker colors to larger counties.

Say that you're interested in only the counties that fall within the state of Colorado. Therefore, you need to tell QGIS what polygons should be shown.

2. **Close the Attribute Table window by clicking the red X in the top-right corner, and then double-click the County layer in the Layers window on the left.**

   The Layer Properties window appears, as shown in Figure 13-15.

3. **Click the window's General tab (active by default) and scroll down until you find the Feature Subset section.**
4. **In the Feature Subset section, click the Query Builder button.**

   The Query Builder dialog box appears, as shown in Figure 13-16.

   The Fields box displays only those fields that are available in the Attribute table of the `county.shp` file.

5. **Double-click the STATEFP entry in the Fields box of the Query Builder dialog box.**

   The STATEFP field appears in the Provider Specific Filter Expression box, located near the bottom of the Query Builder dialog box.

6. **Type** = '08' **after the "STATEFP" entry in the Provider Specific Filter Expression box.**

   The final expression should look like

   ```
   "STATEFP" = '08'
   ```

<tip>

`STATEFP` contains codes that represent the states in America, and in that code, `08` stands for Colorado.

7. **Click OK, and then click OK again.**

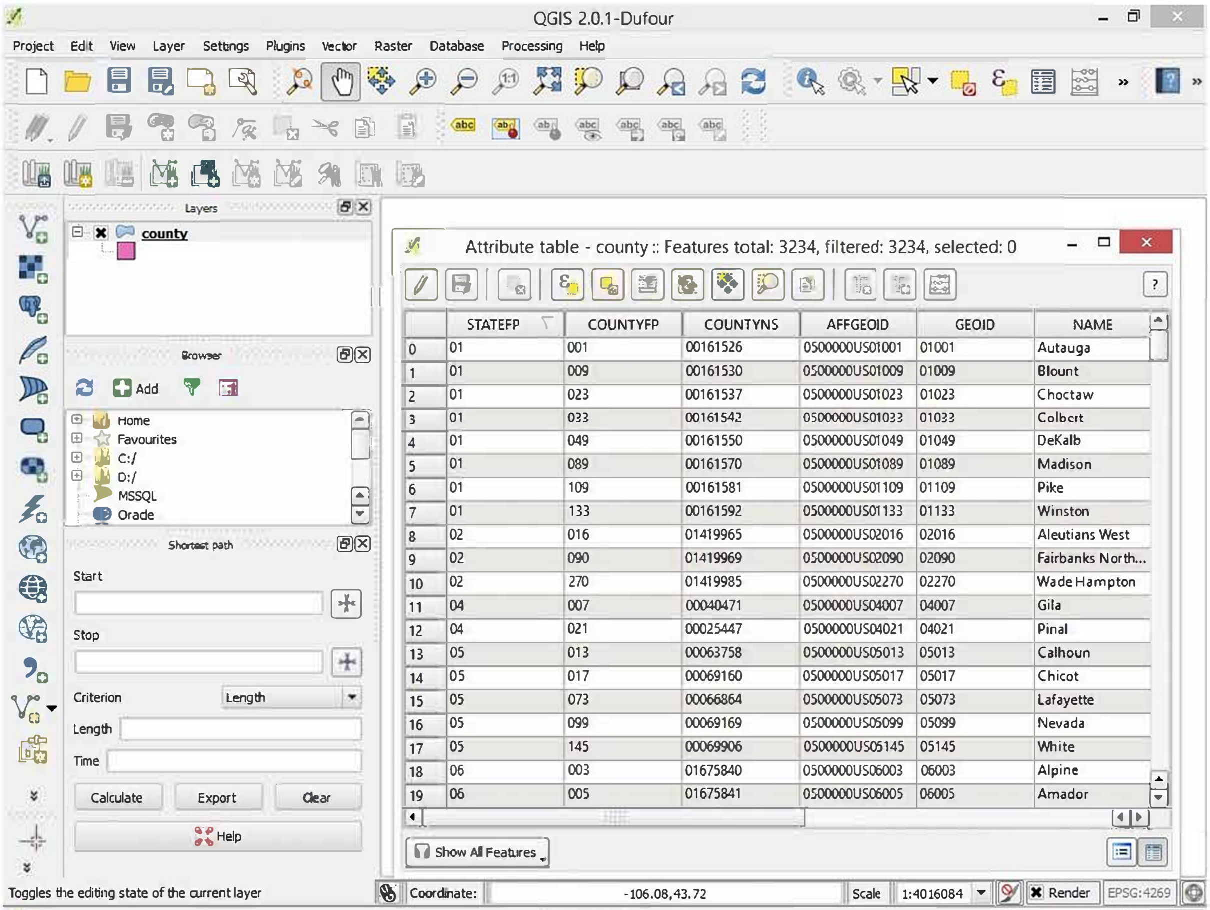
   The main Layer Properties window reappears.

8. **Right-click the County layer in the Layers window and choose Zoom to Layer from the pop-up menu that appears.**

<remember>

You can make these kinds of queries as complicated as you need. You can choose to display only polygons for which the value in a specific column is larger or smaller than a given value, or you can combine different arguments for different fields. QGIS relies on SQL queries.

The map shown in Figure 13-17 displays only the counties that are in Colorado.

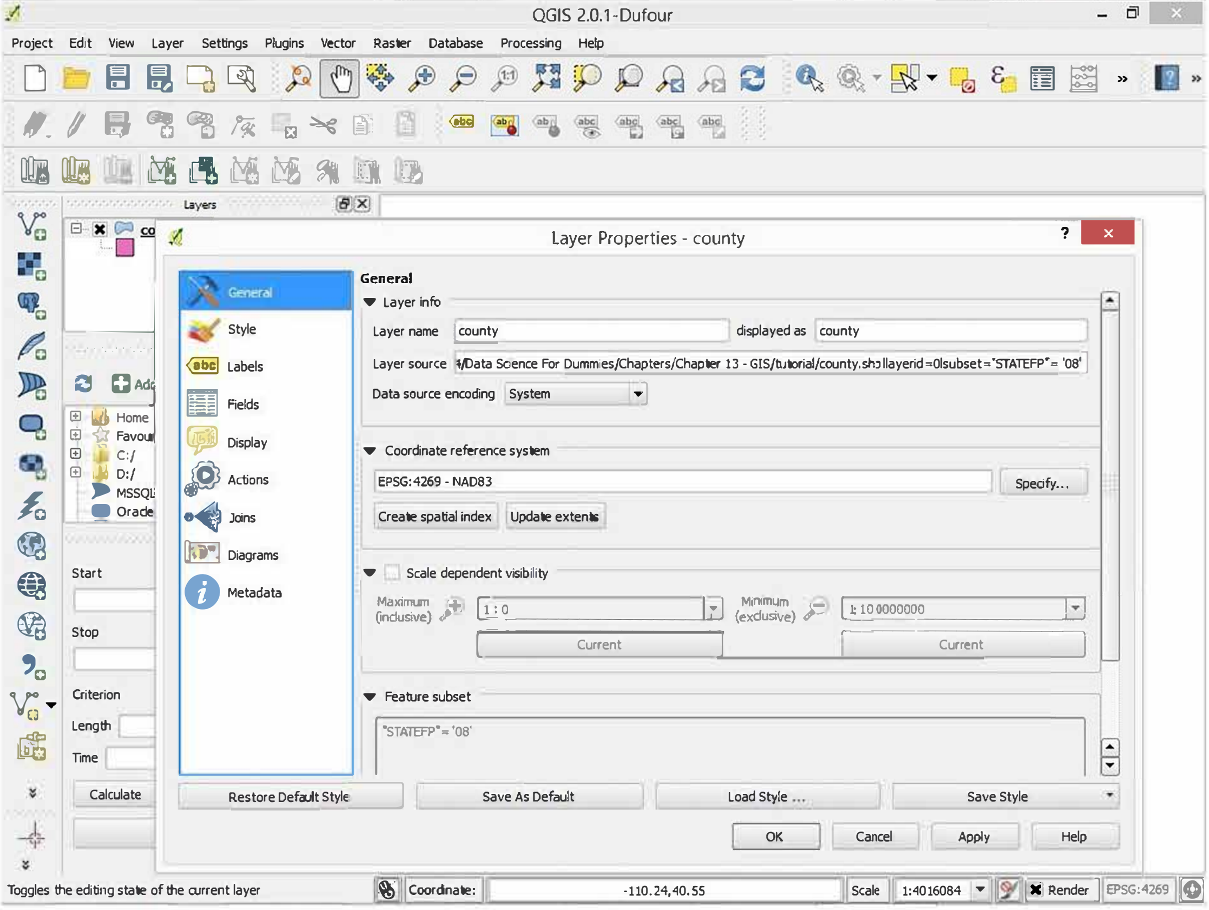


Attribute table - county :: Features total: 3234, filtered: 3234, selected: 0

	STATEFP	COUNTYFP	COUNTYNS	AFFGEOID	GEOID	NAME
0	01	001	00161526	0500000US01001	01001	Autauga
1	01	009	00161530	0500000US01009	01009	Blount
2	01	023	00161537	0500000US01023	01023	Choctaw
3	01	033	00161542	0500000US01033	01033	Colbert
4	01	049	00161550	0500000US01049	01049	DeKalb
5	01	089	00161570	0500000US01089	01089	Madison
6	01	109	00161581	0500000US01109	01109	Pike
7	01	133	00161592	0500000US01133	01133	Winston
8	02	016	01419965	0500000US02016	02016	Aleutians West
9	02	090	01419969	0500000US02090	02090	Fairbanks North...
10	02	270	01419985	0500000US02270	02270	Wade Hampton
11	04	007	00040471	0500000US04007	04007	Gila
12	04	021	00025447	0500000US04021	04021	Pinal
13	05	013	00063758	0500000US05013	05013	Calhoun
14	05	017	00069160	0500000US05017	05017	Chicot
15	05	073	00066864	0500000US05073	05073	Lafayette
16	05	099	00069169	0500000US05099	05099	Nevada
17	05	145	00069906	0500000US05145	05145	White
18	06	003	01675840	0500000US06003	06003	Alpine
19	06	005	01675841	0500000US06005	06005	Amador

Toggles the editing state of the current layer

Coordinate: -106.08,43.72 Scale 1:4016084 EPSG:4269



Layer Properties - county

General

Layer info

Layer name: county displayed as: county

Layer source: ‡/Data Science For Dummies/Chapters/Chapter 13 - GIS/tutorial/county.sh|layerid=0|subset="STATEFP"= '08'

Data source encoding: System

Coordinate reference system

EPSG:4269 - NAD83

□ Scale dependent visibility

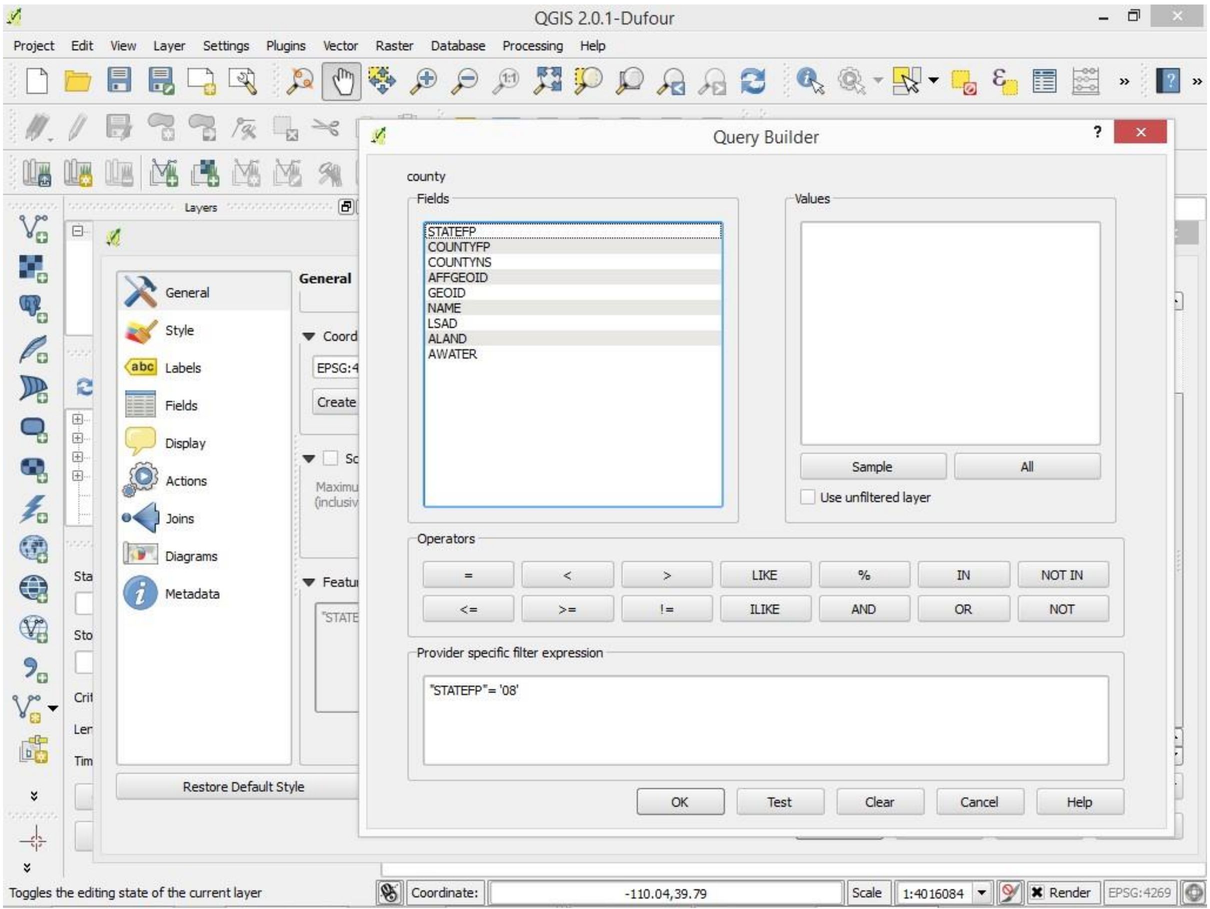
Maximum (inclusive) 1 : 0 Minimum (exclusive) 1: 10 0000000

Feature subset

"STATEFP"= '08'

Toggles the editing state of the current layer

Coordinate: -110.24,40.55 Scale 1:4016084 EPSG:4269



county

Fields

STATEFP
COUNTYFP
COUNTYNS
AFFGEOID
GEOID
NAME
LSAD
ALAND
AWATER

Values

□ Use unfiltered layer

Operators

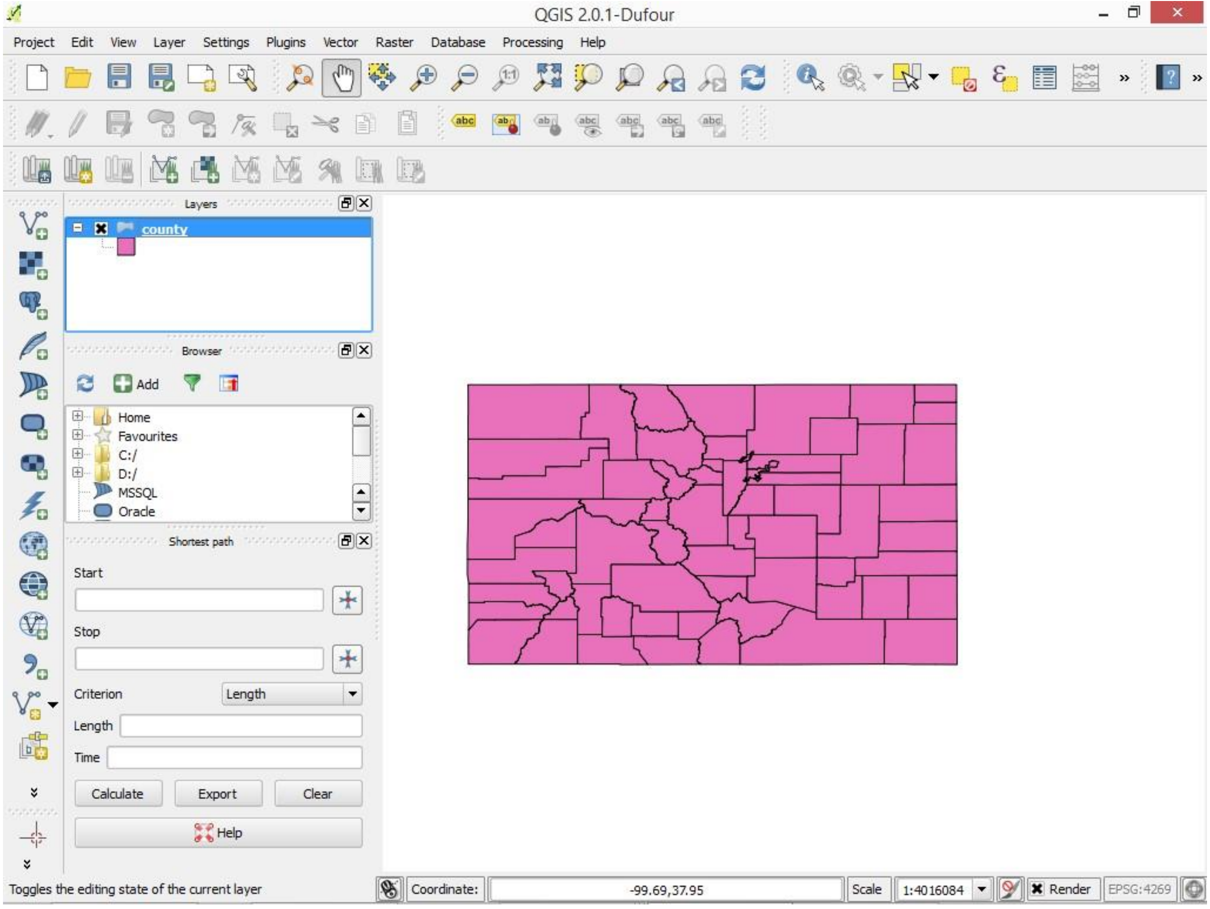
=	<	>	LIKE	%	IN	NOT IN
<=	>=	!=	ILIKE	AND	OR	NOT

Provider specific filter expression

```
"STATEFP"= '08'
```

Toggles the editing state of the current layer

Coordinate: -110.04,39.79 Scale 1:4016084 ☒ Render EPSG:4269



Start

Stop

Criterion: Length

Length

Time

Toggles the editing state of the current layer

Coordinate: -99.69,37.95 Scale 1:4016084 ☒ Render EPSG:4269