Finding help and reporting issues

The QGIS community offers a variety of different community-based support options. These include the following:

- **GIS StackExchange**: One of the most popular support channels is `http://gis.stackexchange.com/`. This is a general-purpose GIS question-and-answer site. If you use the tag qgis, you will see all QGIS-related questions and answers at `http://gis.stackexchange.com/questions/tagged/qgis`.
- **Mailing lists**: The most important mailing list for user questions is qgis-user. For a full list of available mailing lists and links to sign up to, visit `http://www.qgis.org/en/site/getinvolved/mailinglists.html#qgis-mailinglists`. To comfortably search for existing mailing list threads, you can use Nabble (`http://osgeo-org.1560.x6.nabble.com/Quantum-GIS-User-f4125267.html`).
- **Chat**: A lot of developer communication runs through IRC. There is a #qgis channel on www.freenode.net. You can visit it using, for example, the web interface at `http://webchat.freenode.net/?channels=#qgis`.

> Before contacting community support, it's recommended to take a look at the documentation at `http://docs.qgis.org`.
> If you prefer commercial support, you can find a list of companies that provide support and custom development at `http://www.qgis.org/en/site/forusers/commercial_support.html#qgis-commercial-support`.
>
> If you find a bug, please report it because the QGIS developers can only fix the bugs that they are aware of. For details on how to report bugs, visit `http://www.qgis.org/en/site/getinvolved/development/bugreporting.html`.

Summary

In this chapter, we installed QGIS 3.4 and took a first look at the interface. We highlighted some of the exciting new features in the QGIS 3.4 release and looked at how they will impact us. We also explored the panels, toolbars, and menus that make up the QGIS user interface. At the end of the chapter, we interacted with the browser and the layers panel by dragging our data in and looking at how to save a project. Finally, we covered where to find help and report issues. In the next chapter, we will use QGIS to work with data.

2
Data Creation and Editing

At the core of any GIS is data. Without it, we cannot create maps or perform spatial analysis. In this chapter, we will load, edit, and create data. We will look at the large range of formats and types and how QGIS 3.4 helps us to work with them.

If you are familiar with GIS, then you will already know that we often work with vector data, namely points, lines, polygons, and raster data, as **pixels**. Vector and raster data are the core data types that we use in a GIS. However, text files, databases, and web services can also be integrated into GIS. We can use a location associated with a file (such as a coordinate), or we can perform joins to extend or spatially enable our existing data.

In this chapter, we will explore all of the data options in QGIS. The topics covered in this chapter are as follows:

- Data formats
- Loading data
- Interacting with data
- Vector data
- Attributes
- Editing and creating data
- Data joins
- Raster data
- Other data in spatial databases

Data formats

QGIS supports many GIS data formats. It makes use of the OGR library for vector data and the GDAL library for raster data.

> **TIP** To see the latest list of OGR vector formats, check out `http://www.gdal.org/ogr_formats.html`, and to see a list of GDAL raster formats, visit `http://www.gdal.org/formats_list.html`. If you can't find your format on the list, it is probably not supported in QGIS at this time.

Shapefiles are the most common form of vector data today, and QGIS supports this format. Similarly, GeoTIFF are probably the most common form of raster data, and QGIS also supports these. In this book, we will work with and use these formats, as well as the new GeoPackage format.

GeoPackage

QGIS 3 has chosen GeoPackage as its default format. This is an open format, unlike the Shapefile, which is proprietary. GeoPackage also supports rasters. It is built on a `SpatiaLite` database, has no file size limitations, and works as one file. The format was developed by the Open Geospatial Consortium and is increasingly being adopted by organizations around the world. A GeoPackage has a `.gpkg` extension, which unlike the Shapefile has several extensions.

Loading data

We are going to start by loading data into QGIS 3.4. You may have some of your own data that you wish to use, but all of the techniques are applicable to any GIS data you have. We will use the QGIS sample data for these examples. Over the following chapters, we will use the sample dataset to eventually build maps.

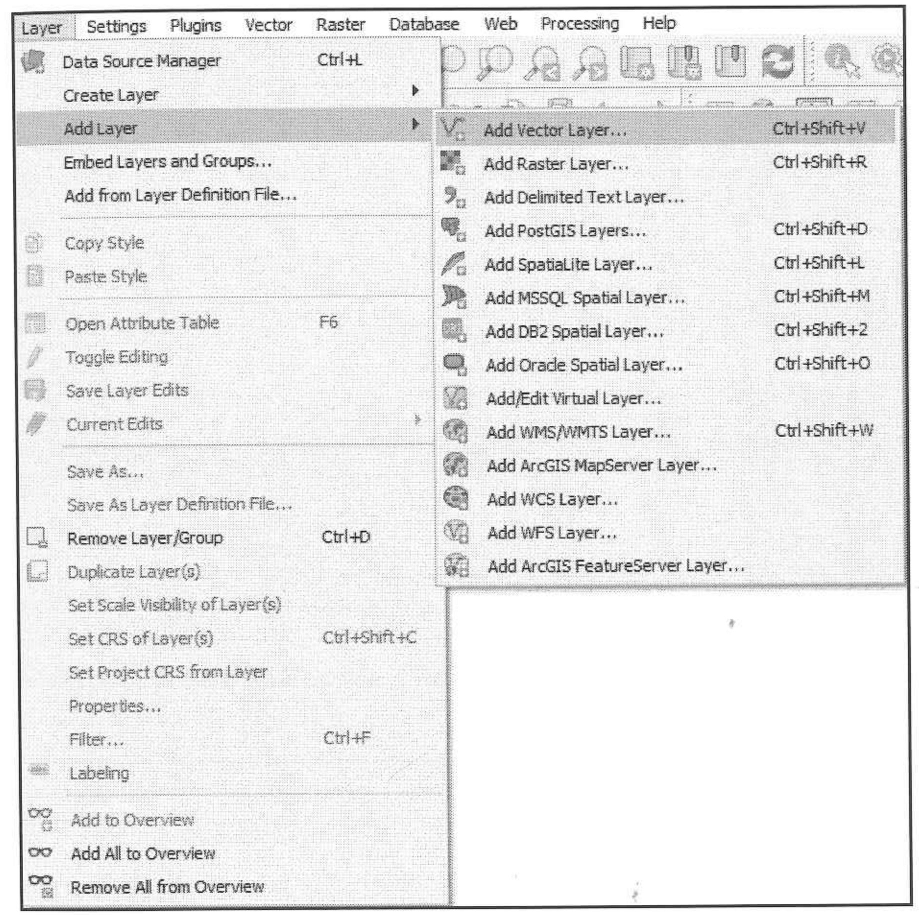
> **TIP** Download the QGIS sample data from `https://qgis.org/downloads/data/` and look for the `qgis_sample_data.zip` file. Download and extract this data to a folder on your computer.

Getting data into QGIS

Loading data into QGIS can be done in several ways. The three most common ways are as follows:

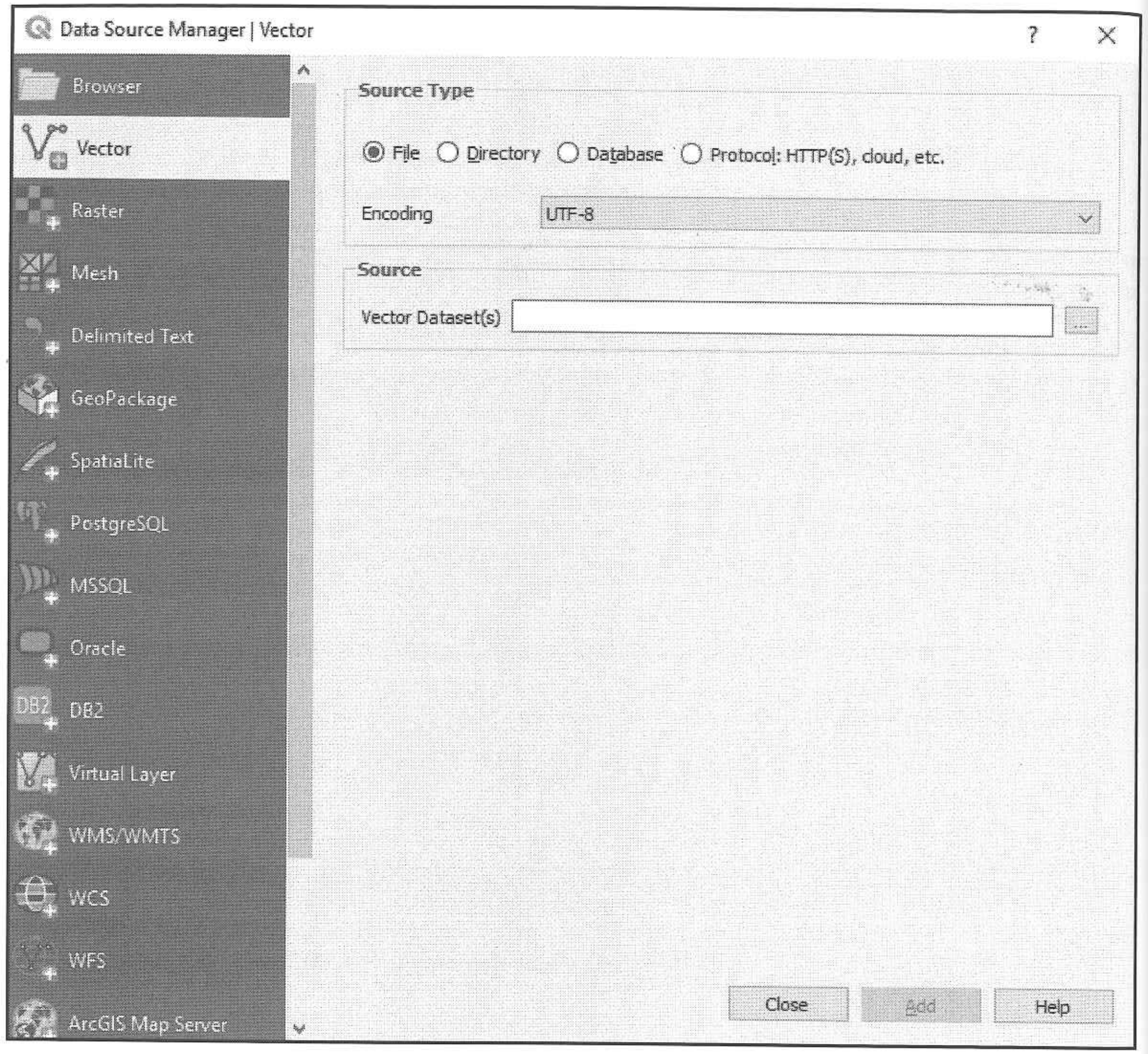
- You can drag data directly from a folder straight into the map.
- You can drag data from the browser panel (shown in `Chapter 1`, *Where Do I Start?*) onto the map.
- You can click **Layer** | **Add Layer** and choose what type of layer to add.

In the following screenshot, we are choosing to load a Vector Layer, which can also be done using the *Ctrl* + *Shift* + *V* shortcut:



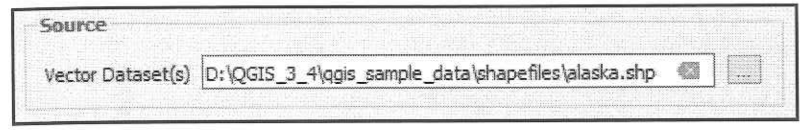
Adding a vector layer

By choosing this method, the **Data Source Manager** window will appear, with **Vector** highlighted, as follows:



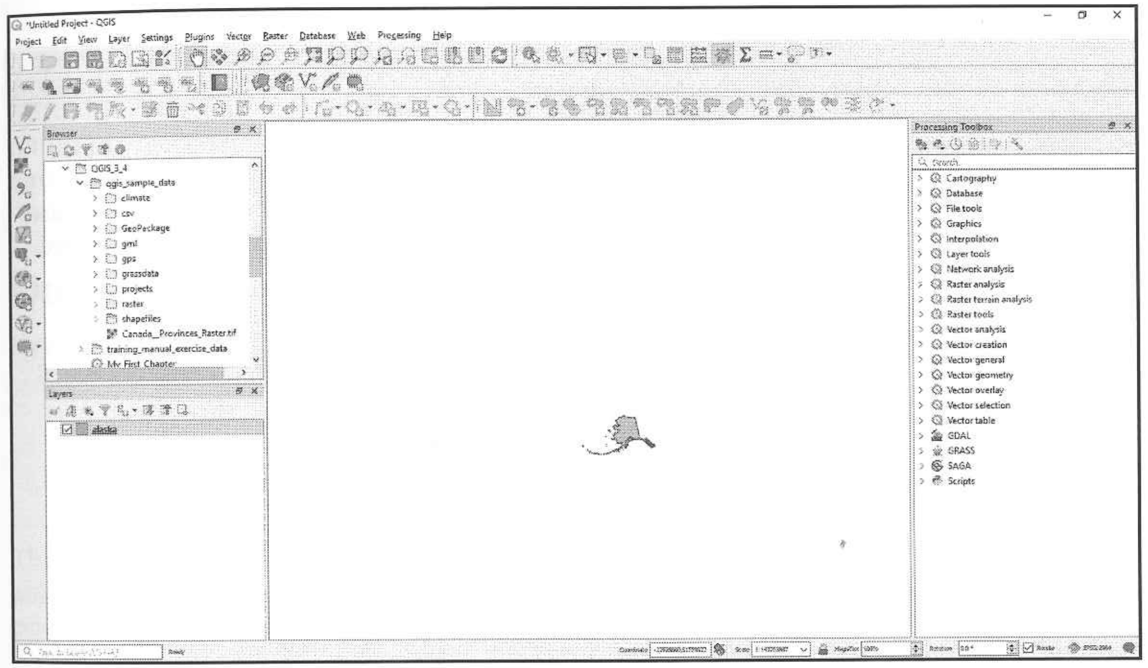
The QGIS Data Source Manager

Click on the ellipsis for **Vector Dataset(s)** and navigate to the layer you wish to load. In the following example, the `alaska` Shapefile is selected. You can load more than one layer at the same time by holding down the *Ctrl* button and clicking on multiple files:



Loading data in the data source manager

If you then click the **Add** button at the bottom of the **Data Source Manager** window, the data will be added to your **Map** window as well as the **Layers** window. The **Data Source Manager** window will remain open, and you can also load other layers. Close the window and your Shapefile will appear in the map, as follows:



Data loaded into the map panel in QGIS

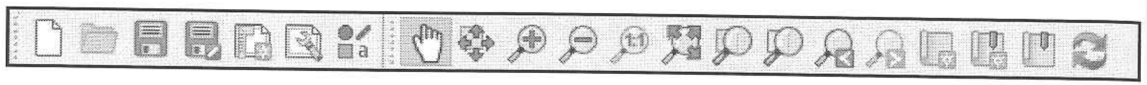
In the next section, we will look at how to interact with the data to zoom in and inspect its attributes.

Interacting with data

When you loaded the Alaska layer, you probably noticed that the data appeared as a relatively small object. In this section, we will cover some of the basic interactions we can perform with data. Let's start with zooming in.

Navigation

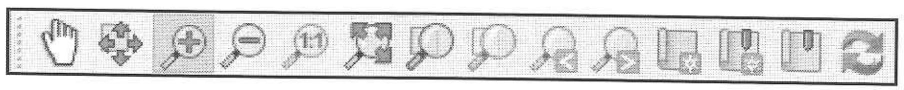
Later, we will show you how to zoom in on a layer to look at it in more detail. But first, let's remind ourselves of the project and navigation toolbar:



Project and Navigation toolbar

These tools are the workhorse of the GIS. They allow us to move around, inspect, measure, and select data. All are critical in any GIS, and if you have used mapping software before, you should be familiar with them.

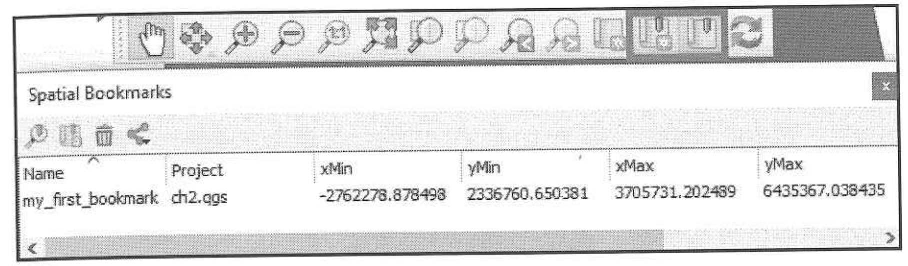
Start by clicking on the zoom-in icon (a magnifying glass icon with a + in it). When this option is selected, it will be highlighted in a different shade, just like any other interactive button in QGIS. This is demonstrated in the following screenshot:



Clicking on the zoom tool in the navigation toolbar

Left-click and draw around the Alaska shape. The map window will quickly refresh around the shape you have drawn. You can then click on the white hand to pan around the map to adjust to your preference. You can also use the scroll button on your mouse to zoom in and out of the map. Some of these navigational tools are grayed out; this is because a setting or event has not happened yet to enable them.

The bookmark tool is shown in the following screenshot and highlighted in the red box:

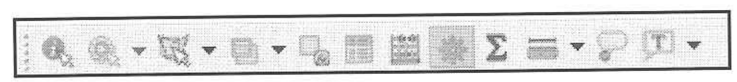


Spatial bookmarks

Spatial Bookmarks allow you to save your current map extent. This means that you can return to this bookmarked view later, or in another project. In the preceding example, I have created a bookmark called `my_first_bookmark`.

Data attributes toolbar

Here is the attributes toolbar:

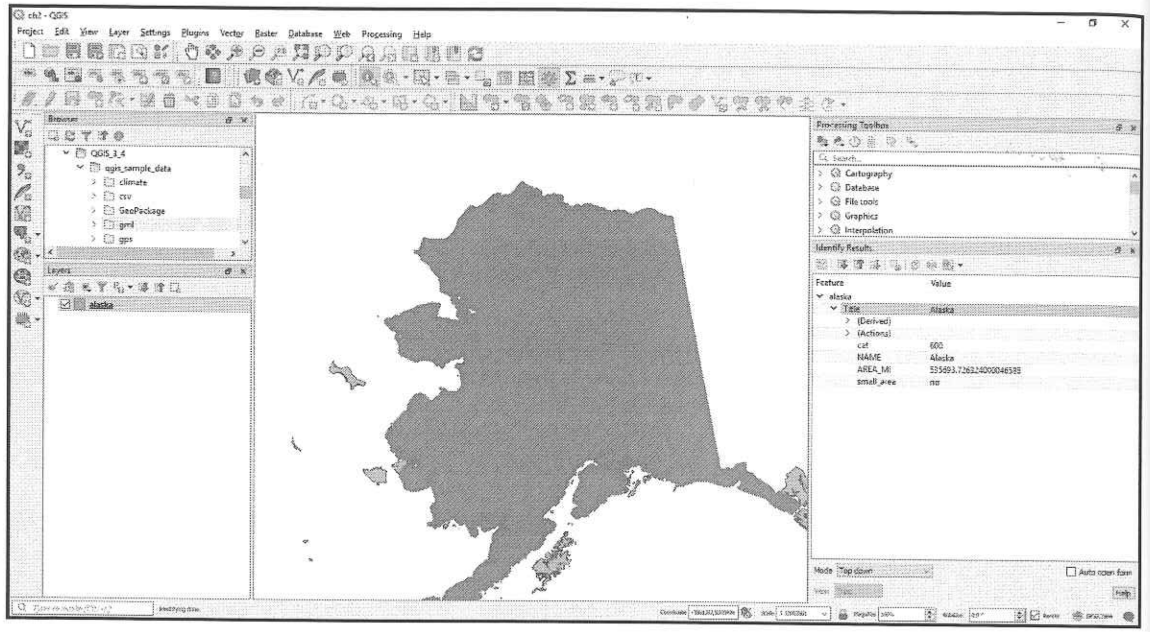


Attributes toolbar

This contains tools to inspect, measure, and select data, among other tools. In order to be able to use these tools, we need to have a layer selected in the layers window. Left-click once on the Alaska layer and this toolbar will have all of its options available.

Inspecting the data

Click on the blue **i** button in the attributes toolbar and then click on the layer in the map. This will display the feature attributes in a new panel, which will dock by default. Like all panels, you can move them around and resize as needed. Your QGIS project should look similar to the following diagram:



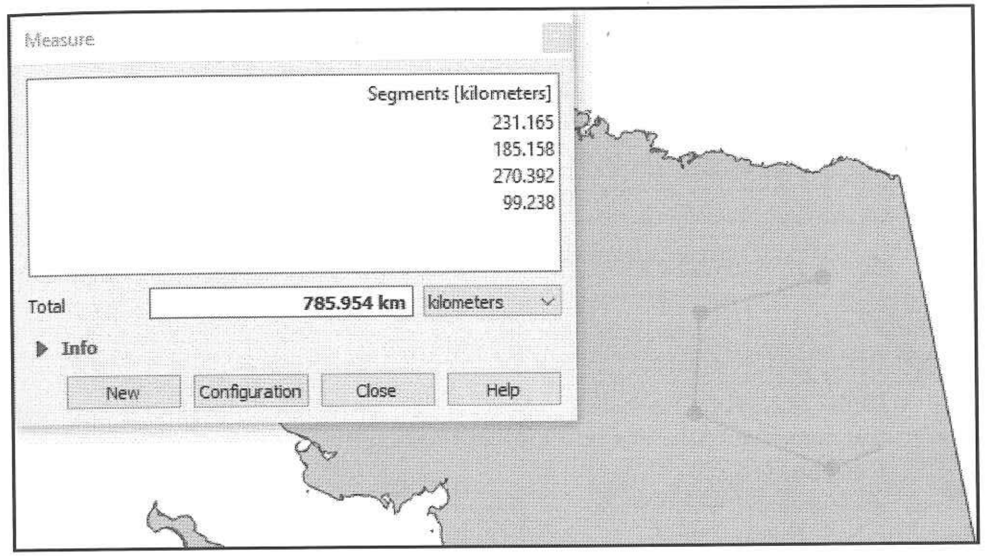
Inspecting data in QGIS

The new panel is called **identify results**. In this case, attributes include a **Category**, a **Name**, and an **Area value**. Later in this chapter, we will look at creating and editing data, as well as adding attributes.

Measuring data

The measure tool does not require a layer to be selected in the layer panel in order for it to work. This means that it is independent of a layer(s). The measure tool can measure length, area, or angle, and you can use the drop-down button next to the tool to select the required option. In this example, let's measure length. Select the measure line option and then left-click on the map to begin the measurement. After doing this, left-click again to measure the segment. Continue to left-click to measure segments of the line or right-click on the end point to finish measuring.

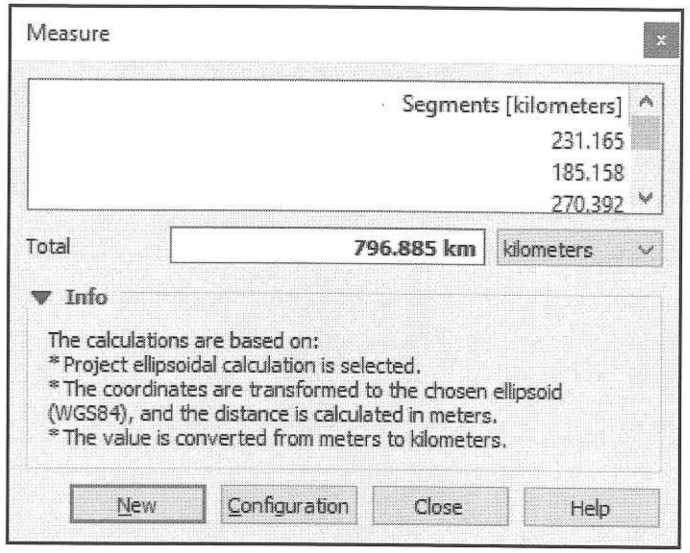
The following diagram shows a series of left clicks on the Alaska layer:



Measuring distance in QGIS

Use the drop-down boxes to adjust the units of measurement you require. The default is set to **meters**; we have changed ours to **kilometers** to obtain a more usable distance unit.

If you expand the arrow next to **Info** in the **Measure** toolbox, you will get more information about how the measure tool is calculating the distance/area. An example is shown in the following screenshot:



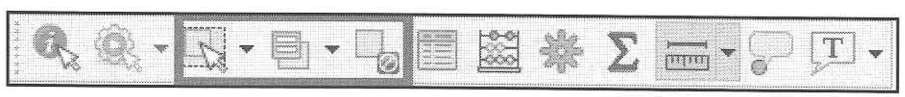
The measure tool info box

TIP

Make sure you check that you are in the expected projection before using the measure tools. We will look at Projections in the *Vector data* section of this chapter.

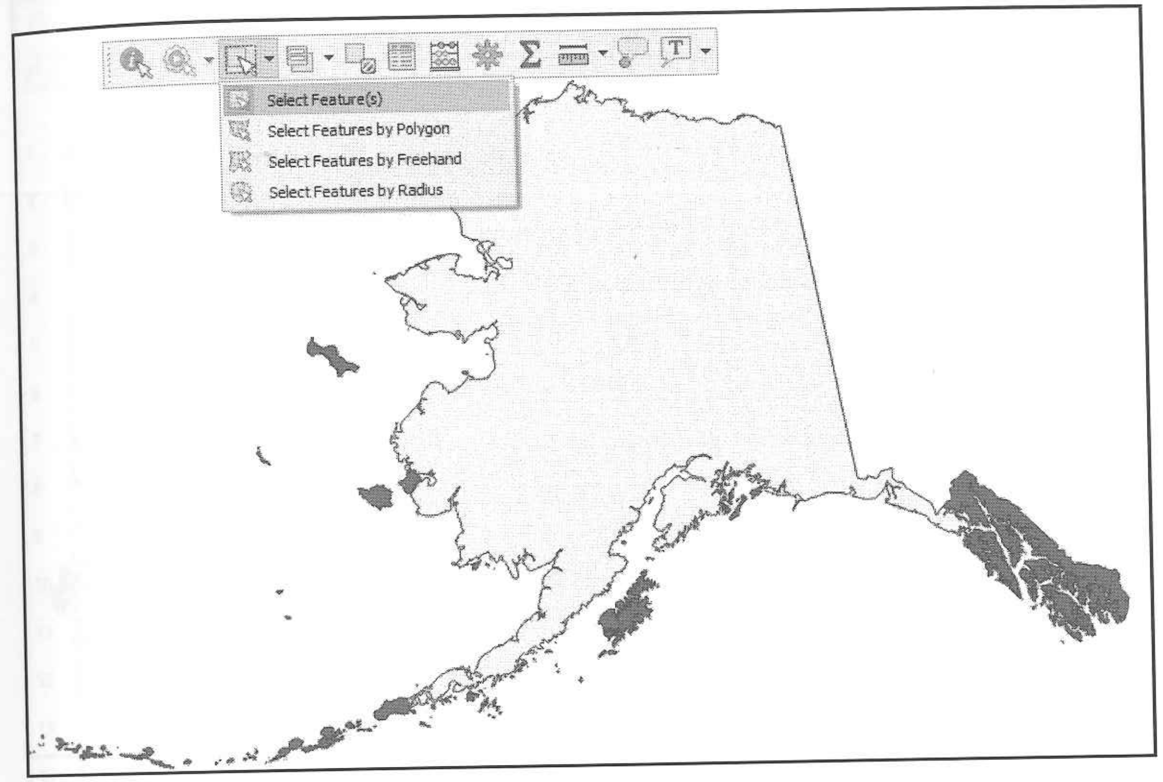
Selecting data

Selecting data is very useful in GIS. We can create a new layer from a selection, or we could add new data specific to that selection. QGIS allows multiple ways of selecting data, and we will cover some of these here. Later, in `Chapter 6`, *Extending QGIS with Python*, we will look at spatial selections. The data attributes toolbar allows selecting by area (this means drawing on the map by hand) and selecting by value (this means passing a query to select the data). The selection tools are highlighted in the following screenshot:



The select data tools

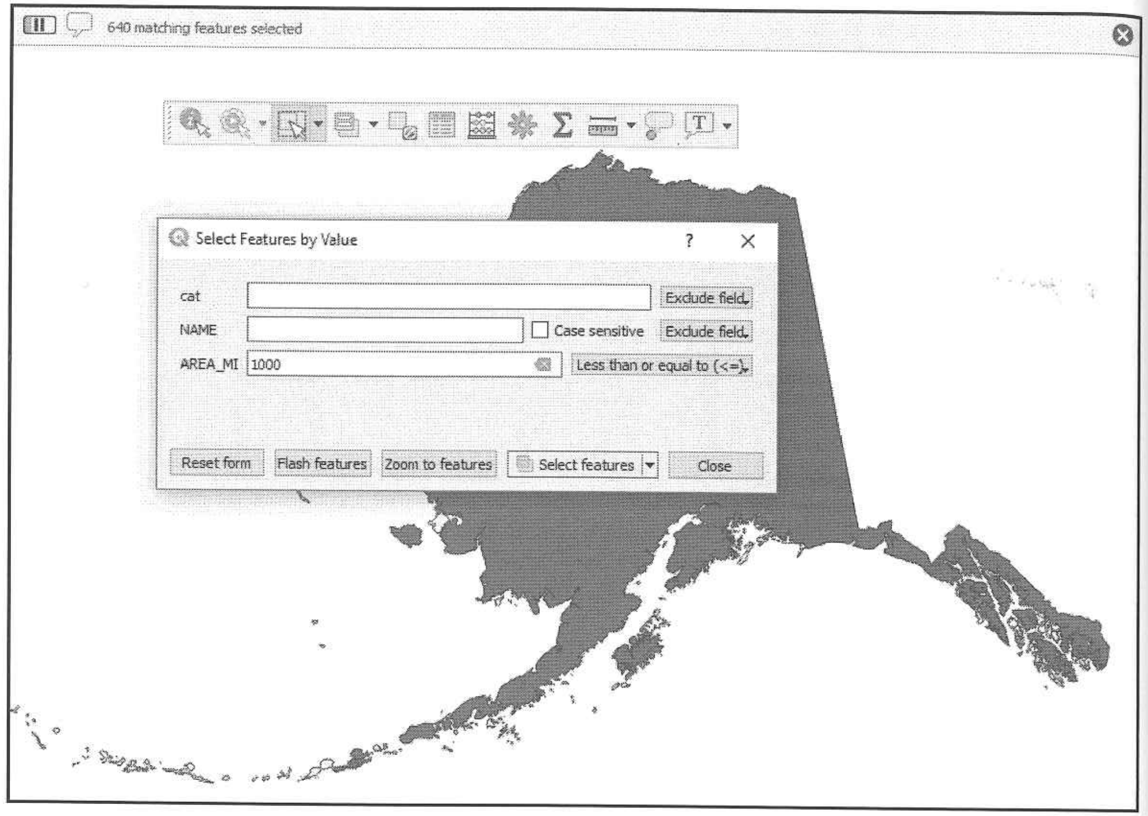
Working from left to right, the tools are Select by Area, Select by Attribute, and Clear Selection. Click on the Select by Location button and click on the largest area associated with the Alaska layer. It will change to yellow to highlight the selected layer. The following screenshot shows the other Select by Area options in the drop-down list:



Feature selected in QGIS

Click on **Clear Selection** to remove any selections from the layer.

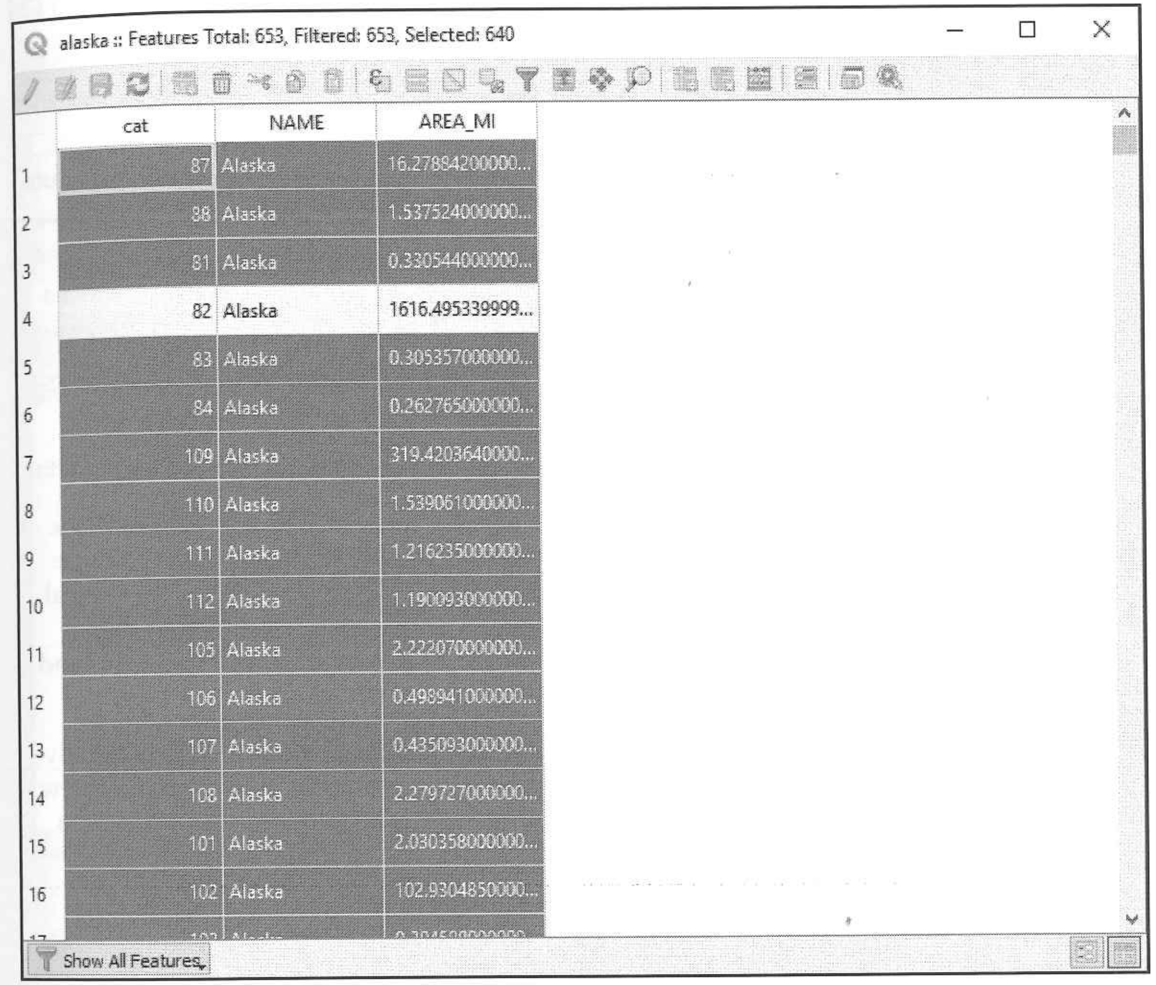
The **Select by Value** button is the second button on the toolbar described previously. Click on this button to open a new window:



Map showing the selected features by value option

In the **Select Features by Value** window, we have clicked on the **Exclude field** button associated with **AREA_MI** and selected **Less than or equal to (<=)**. As a parameter, we have typed `1000` into the box. We then clicked **Select features** and QGIS selected the features that match this expression. At the top of the screen, a notification tells us how many features have been selected. In this case, it is `640`.

If we click on close and then press *F6* or click on the button immediately to the right of the **Clear Selection** button, the attribute table will appear with the selected rows highlighted. This attribute table for the **Alaska** layer is shown as follows:



Attribute table with features selected

When you are finished inspecting this table, close it down by clicking the close button, on Windows this is an X symbol in the top-right corner. Keep this selection handy, as we will use it in the next section.

There is one more very useful tool worth mentioning, and that is the **Field Calculator** tool. This button looks like an abacus and is next to the **Attribute Table** button. We will use this tool next to begin creating our own data.

Vector data

Now that we have become more familiar with inspecting, selecting, and interacting with data within QGIS, it's time to create data. In this section, we are going to edit attribute data, create vector data, and join data.

Editing attribute data

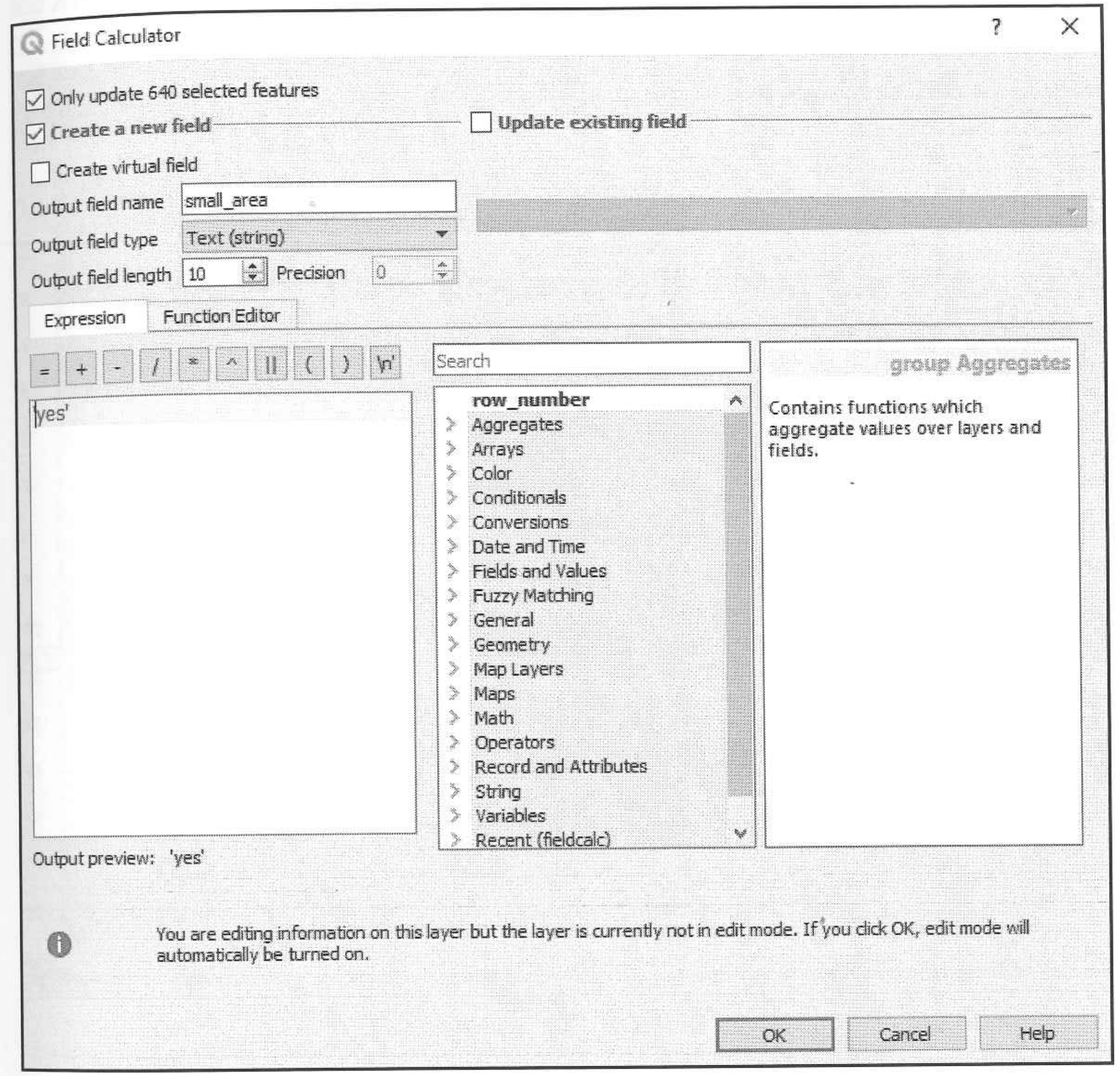
With the **AREA_MI** <= `1000` selection, let's add a field and populate it with a marker based on this selection.

Click on the **Field Calculator** button in the attributes toolbar to open up the field calculator.

This window contains considerable information; there is a great deal of power within the field calculator. In our case, we want to create a new field called `small_area` and assign `yes` if it is =< `1000`, or `no` if it is >`1000`. As with many of the tools in GIS, there are several ways of doing this. The method we will use is designed to help you to build confidence first. In `chapter 6`, *Extending QGIS with Python*, we will look at more complex queries and show how this particular query could be addressed in one step.

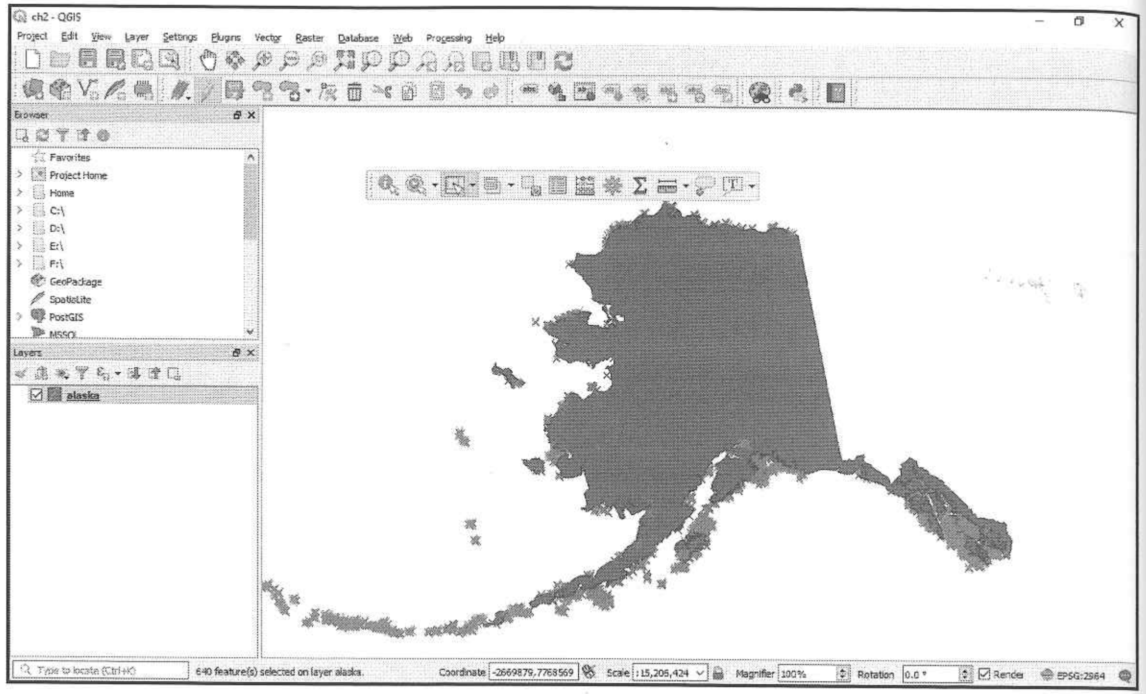
As shown in the following screenshot, the first check box is ticked. This tells QGIS to only apply the calculation to the selected data. The second check box tells QGIS to create a new field. We could update an existing one if we wanted, but at this point we want to build a new field and populate it. Set the field name to `small_area` and the type to **Text**.

Accept the defaults for field length and in the expression window type `yes` (including the single quotations). Your **Field Calculator** should now look like this:



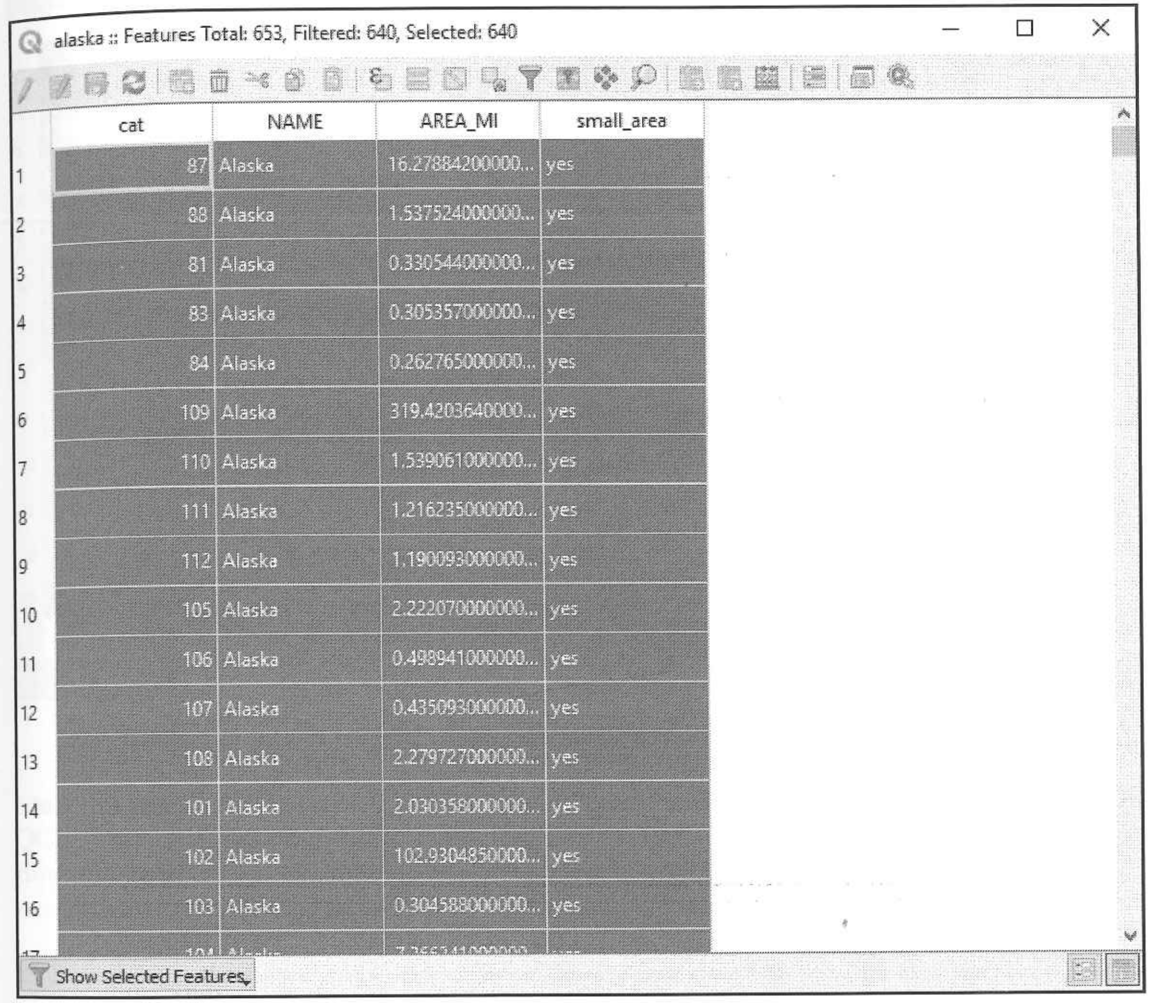
Field Calculator

By clicking on **OK**, you will have started an edit session and populated a field with `yes` based on your **Selection by Value** selection. Your screen will look similar to this:



An edit session in QGIS

In the **Layers** window, there is a pencil icon on the toolbar. This tells us that this layer is currently open in an edit session. Let's take a look at the attribute table; press *F6* or click on the **Attribute Table** button in the attribute toolbar. It should look like this:



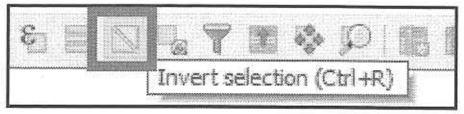
alaska :: Features Total: 653, Filtered: 640, Selected: 640

	cat	NAME	AREA_MI	small_area
1	87	Alaska	16.27884200000...	yes
2	88	Alaska	1.537524000000...	yes
3	81	Alaska	0.330544000000...	yes
4	83	Alaska	0.305357000000...	yes
5	84	Alaska	0.262765000000...	yes
6	109	Alaska	319.4203640000...	yes
7	110	Alaska	1.539061000000...	yes
8	111	Alaska	1.216235000000...	yes
9	112	Alaska	1.190093000000...	yes
10	105	Alaska	2.222070000000...	yes
11	106	Alaska	0.498941000000...	yes
12	107	Alaska	0.435093000000...	yes
13	108	Alaska	2.279727000000...	yes
14	101	Alaska	2.030358000000...	yes
15	102	Alaska	102.9304850000...	yes
16	103	Alaska	0.304588000000...	yes

Show Selected Features

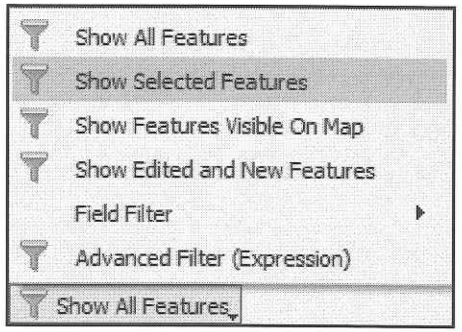
The attribute table with the selected features and the new field with the values populated

Congratulations! You have now created some data! Six-hundred and forty rows of data have now been updated with `yes` in the `small_area` field. But what about the remaining thirteen rows? They will now have no entry for the `small_area` field. Let's change that. Click on the **Invert selection (Ctrl +R)** button in the attribute table. It looks like this:



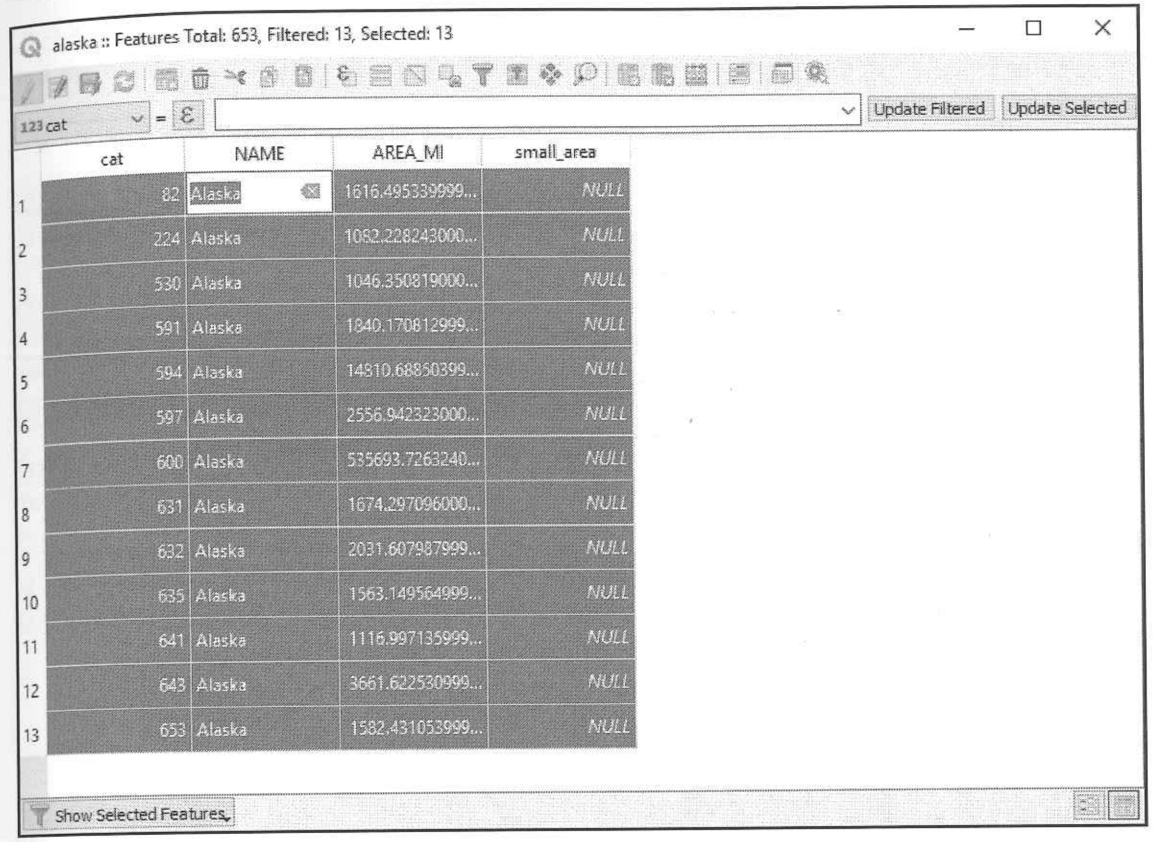
Inverting a selection

This will invert your selection. Once clicked, it may appear that you have lost everything. However, click on the button at the bottom left of the attribute table and click **Show Selected Features**. It will look like this:



Show all features option

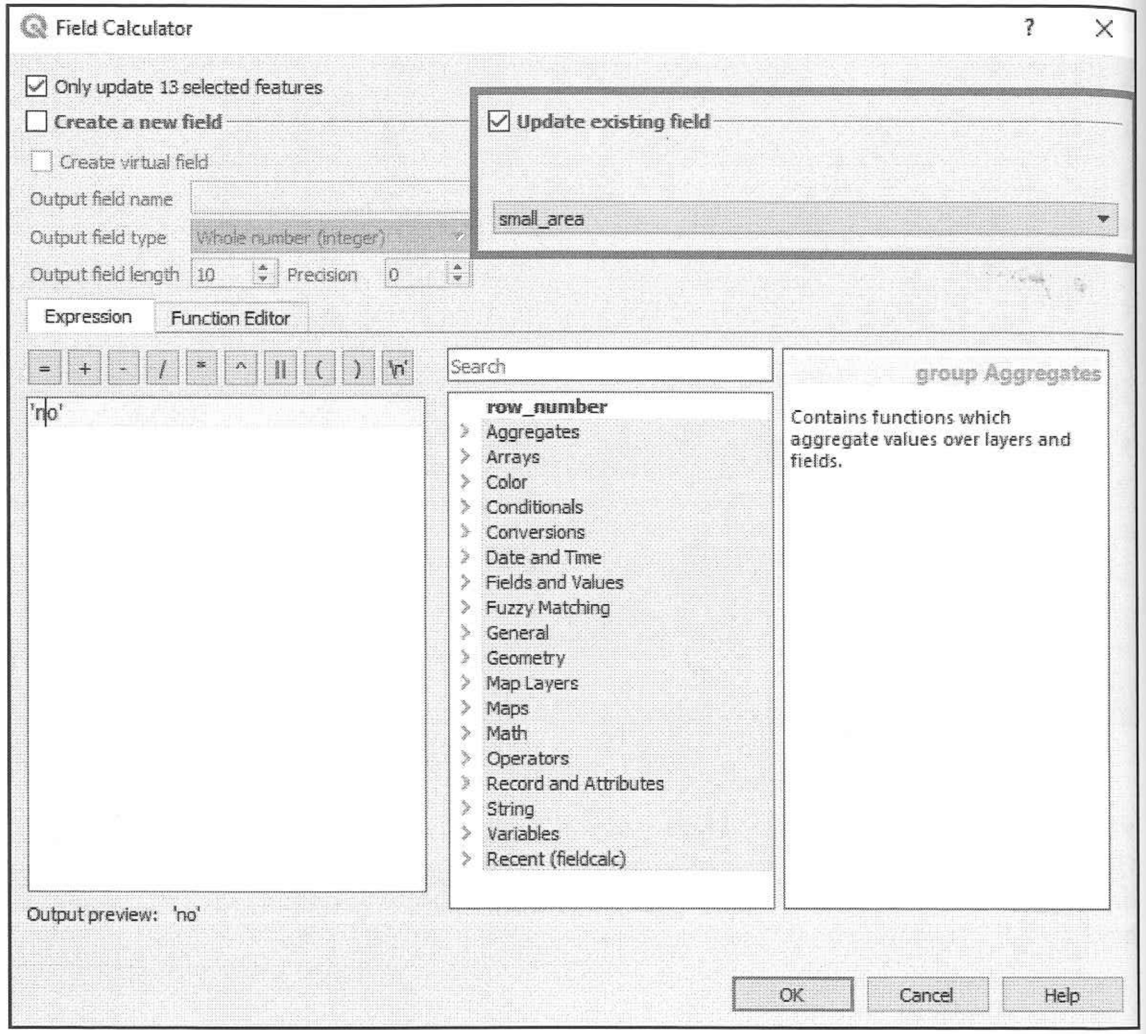
Now we have effectively selected all features with an **AREA_MA** > `1000`. You can see that the `small_area` values associated with this area are set to `NULL`:



Attribute table after inverted selection

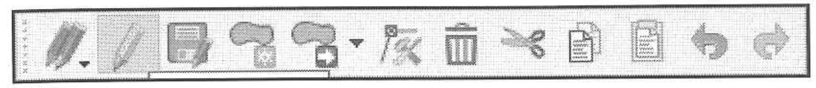
You may have already noticed that the attribute toolbar is not only in the main display of QGIS, but it is also in the attribute table. In this toolbar, select the **Field Calculator**. This time we are going to set the selected rows to `no`. Remember, this time we are not creating a new field, but updating one.

Your field calculator should look similar to the following screenshot. We have highlighted the important difference compared to when you are creating a new field:



Field calculator shows how to update an existing field, with selected features set to No

Click **OK** and then **Clear Selection**. Stop the editing session by clicking on the single pencil and then select `yes` to save the edits:



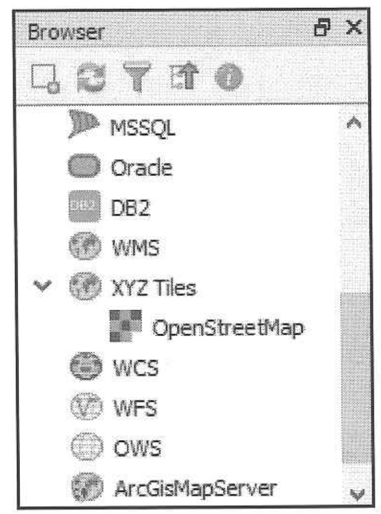
Editor toolbar

We have just edited attribute data in QGIS.

Building your own vector data

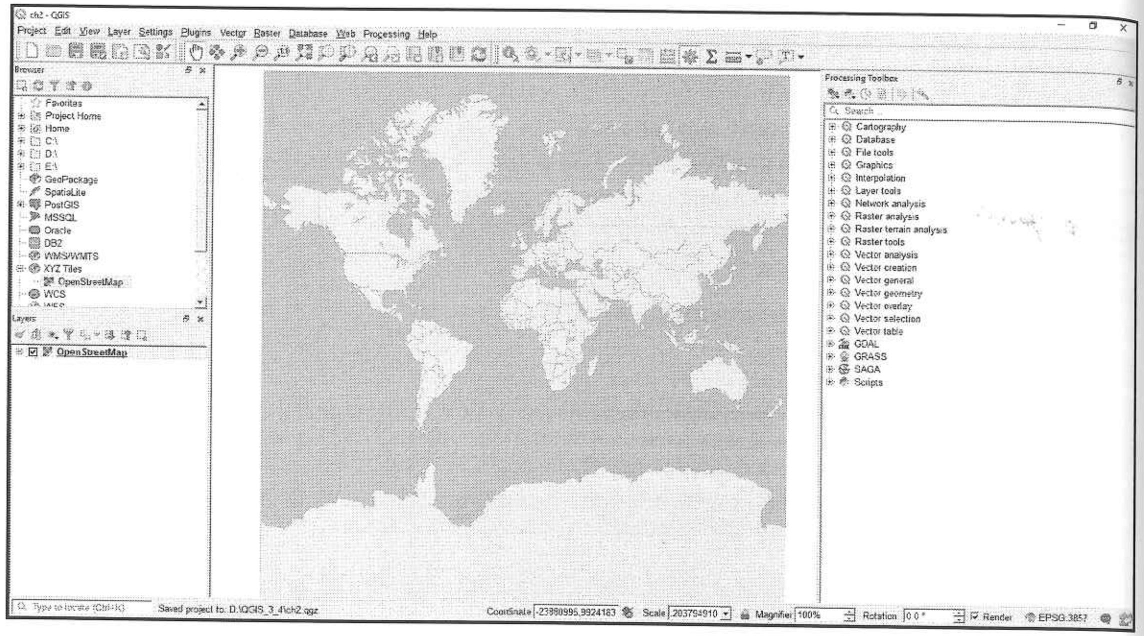
Start a new project in QGIS by either clicking **Project** | **New**, pressing *Crtl* + *N*, or clicking on the new project icon on the project toolbar. You will get a prompt if you wish to save the project; if you are following through this chapter, there is no need to save the project.

In a new project, scroll in the **Browser** window to the **XYZ Tiles** and expand the dropdown. In here, you will find **OpenStreetMap**:



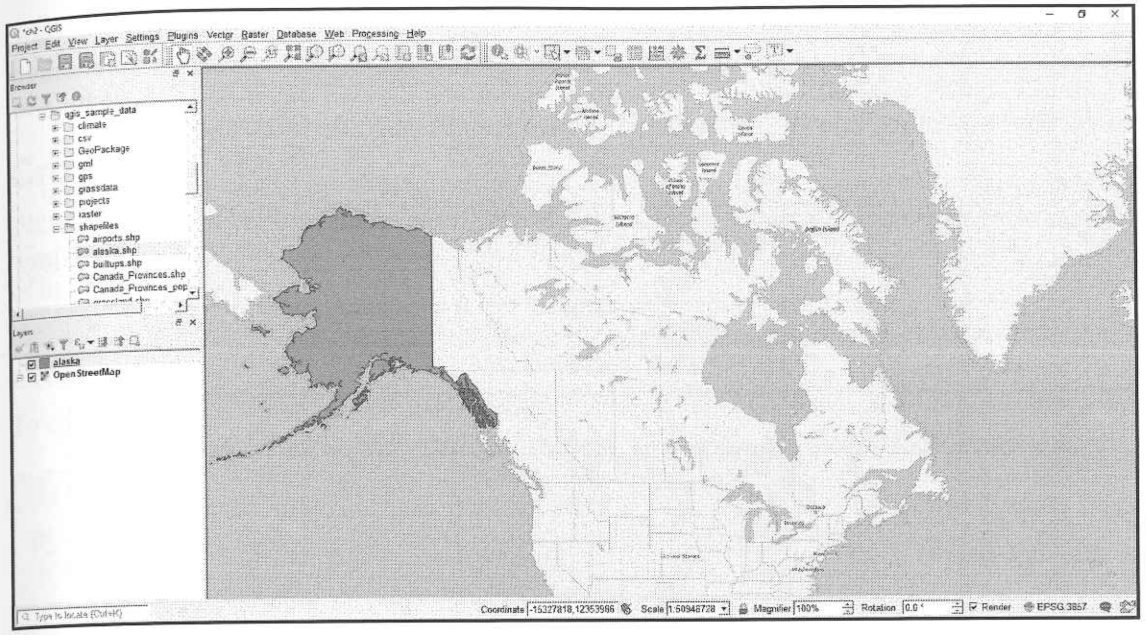
Browser window

Drag **OpenStreetMap** into the **Layers** window. The **OpenStreetMap** data will appear in the map view. Have a look in the bottom-right corner of the screen at the projection. This has changed to **EPSG 3857** (Web Mercator) as this is the native projection of the **OpenStreetMap** data you have just loaded. Your screen should look similar to this:



QGIS with OpenStreetMap layer loaded

Find the `Alaska.shp` that we used previously and load that into QGIS as well. Use the map navigation tools to zoom and pan to the data. The following diagram shows the data loaded together. The projection in the bottom-left corner has not changed, even though the projection of the **alaska** layer is different to the **OpenStreetMap Tile**. This is because QGIS 3.4 re-projects on the fly:



Alaska Shapefile loaded and re-projected on the fly into QGIS

TIP

You can right-click on any layer in the **Layers** window and look for the **Source** tab; it will contain information about the layer's projection.

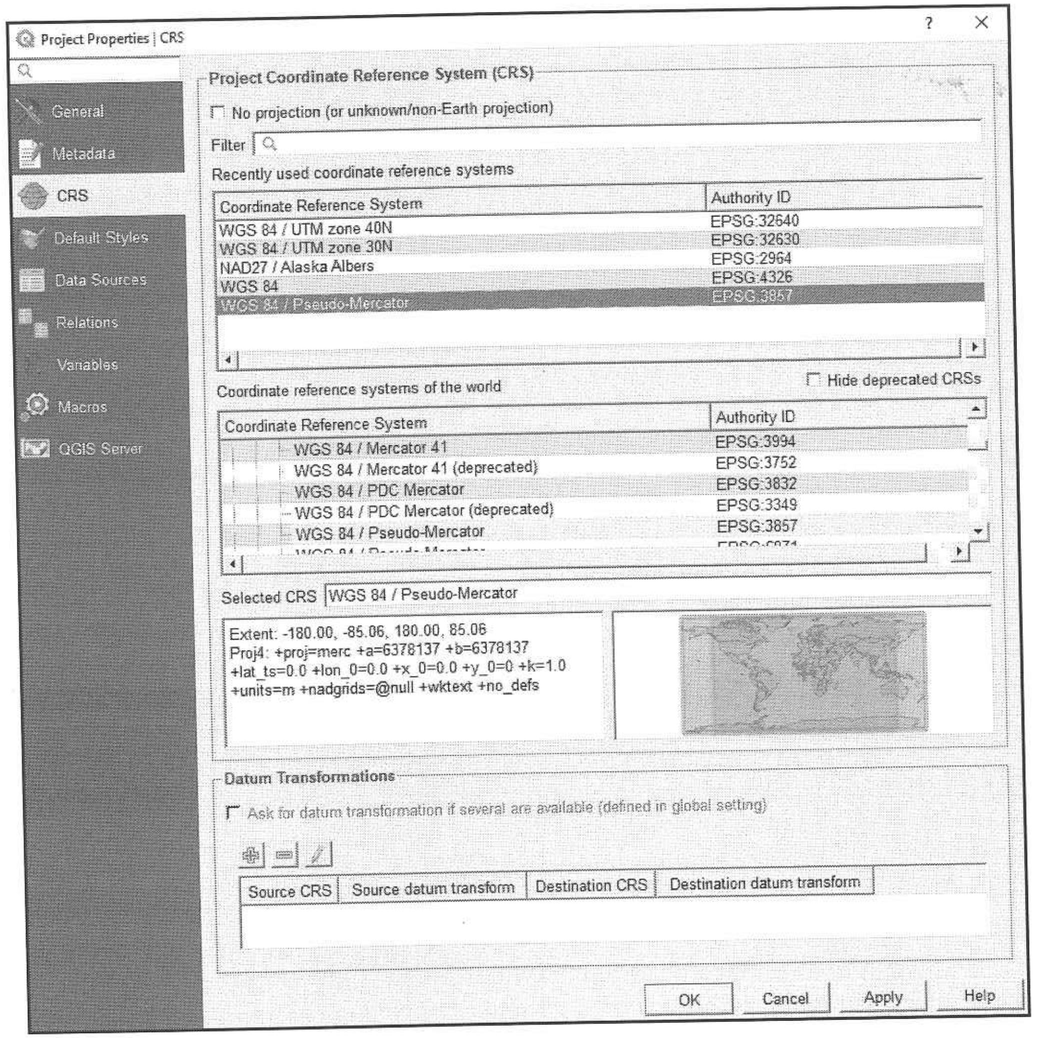
Projections

The default projection for QGIS (on load the first time) has the EPSG code 4326 (WGS 84). Unless changed, the default settings in QGIS allow the projection of the map to be set by the projection of the first layer loaded into the map, assuming that the data has a projection associated to it. If you are interested in learning more about this, a good starting point is the QGIS entry on Geographic Coordinate Systems:
`https://docs.qgis.org/testing/en/docs/gentle_gis_introduction/coordinate_reference_systems.html`

Suffice it to say that getting the projection correct in your GIS is fundamental to any analysis, mapping, or derivation of knowledge from your data. The world is not flat, and projecting your data is the process of taking the part of the Earth's surface you are working on and mathematically shrinking and distorting it so it fits on a plane. You can move from one projection system to another by transforming your data.

Let's use the Alaska layer as an example. The projection of the layer is **EPSG 2964**, and on load of this data, your QGIS project will be set to this projection. If you add data in another projection, QGIS will make re-projections of your data on the fly. This is very powerful and useful, but be careful that you know what projection system *all* your data is in.

To find out and change your map projection, look at the information bar. On the bottom-right corner, you will see that the projection is currently set to **EPSG 3857**, as we have loaded the **OpenStreetMap** data. Click on this to bring up the projection properties dialog:



Projection properties

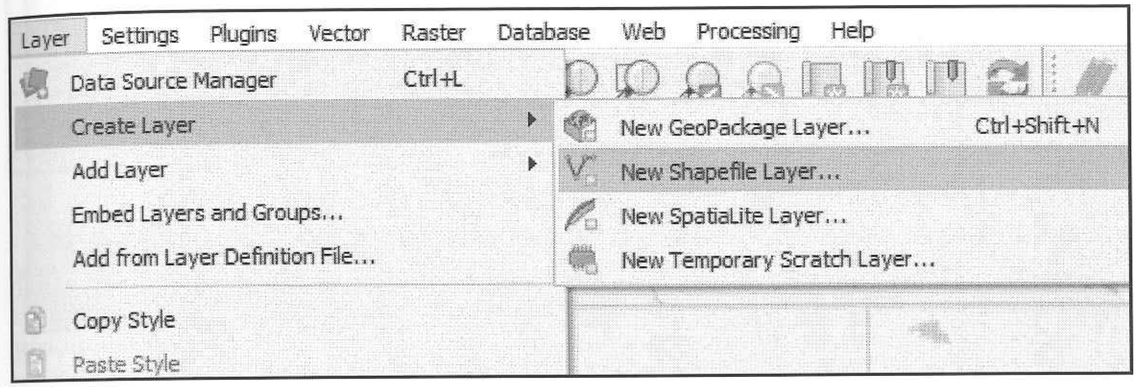
In the next section, we will create some data in this using this projection. Click on **OK** to close the coordinate properties dialog box.

Creating data

We will now create some data. During this exercise, we will be using digitizing tools to map two provinces in Canada. We will create a field in the attribute table and assign values in this table.

Create a ShapeFile

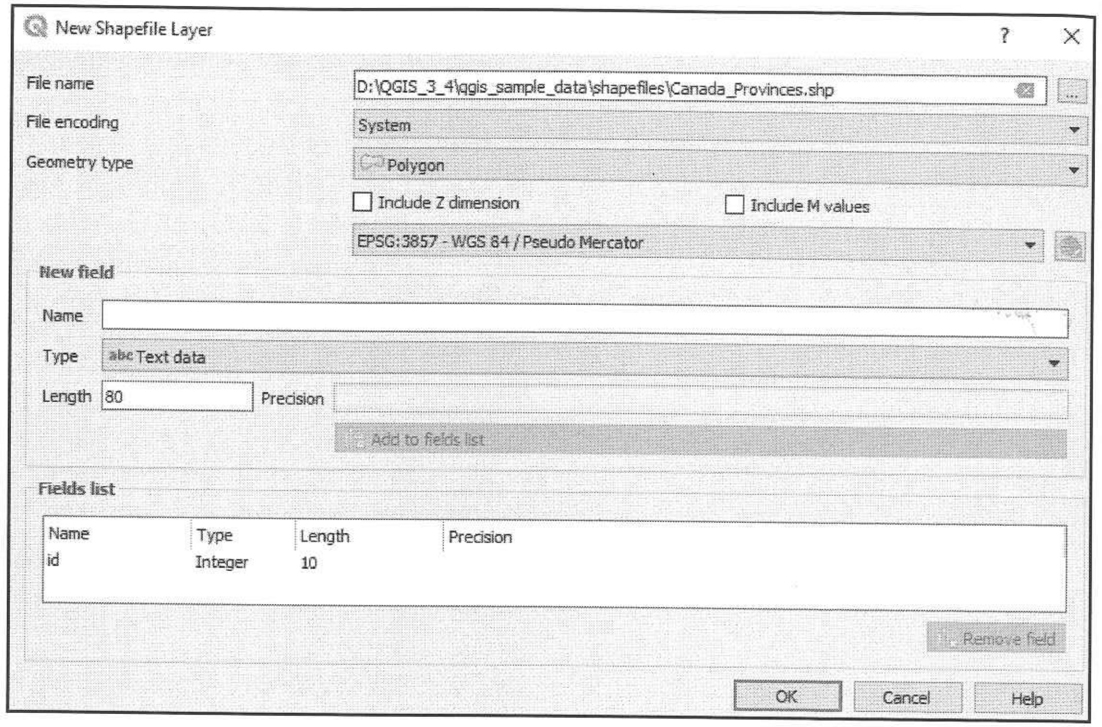
To create a Shapefile in QGIS 3.4, click on **Layer** | **Create Layer** | **New Shapefile Layer**:



Creating a New Shapefile Layer

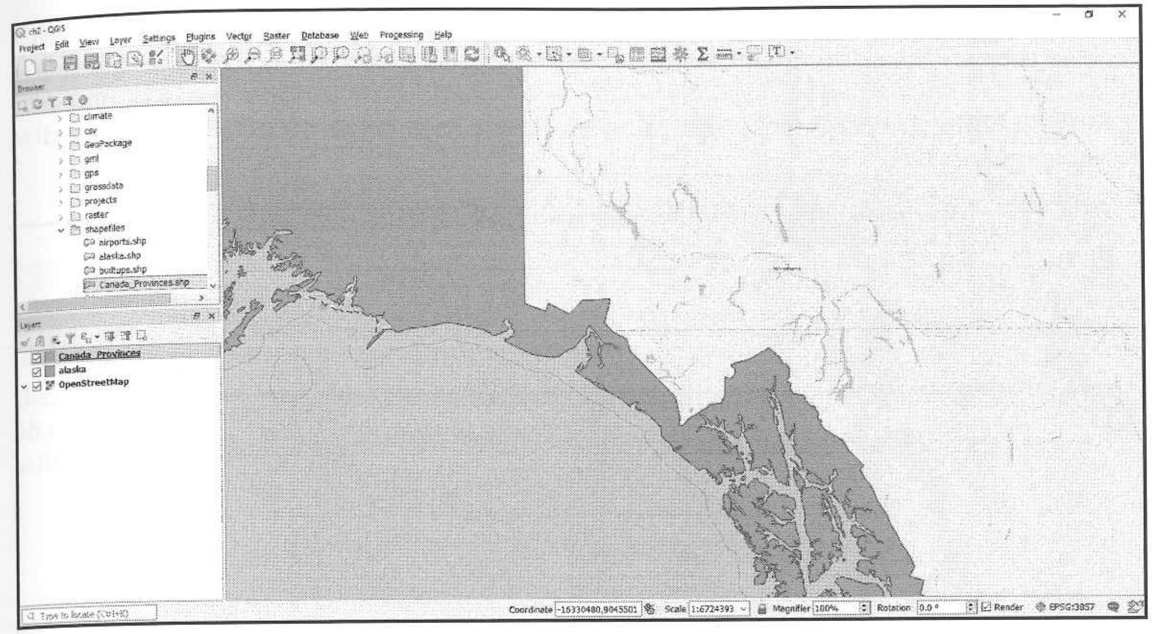
Again, we are going to be creating a new **Polygon** layer. We are assigning the name `Canada_Provinces.shp` in the `Shapefiles` folder or `qgis_sample_data` (where `Alaska.shp` is stored). The **Geometry** type is set to **Polygon,** and we have accepted the default project (as defined by the projection in the map window). We are not assigning any additional fields in the table at this point, but this is one way that you could define fields.

The dialog box should look similar to this:



Creating a New Shapefile Layer dialog box

Click on **OK** and the newly created Shapefile will be added to the map, the **Layers** window, and the **Browser** window. The layer also becomes the active selected layer in the **Layers** window, as shown in the following diagram:



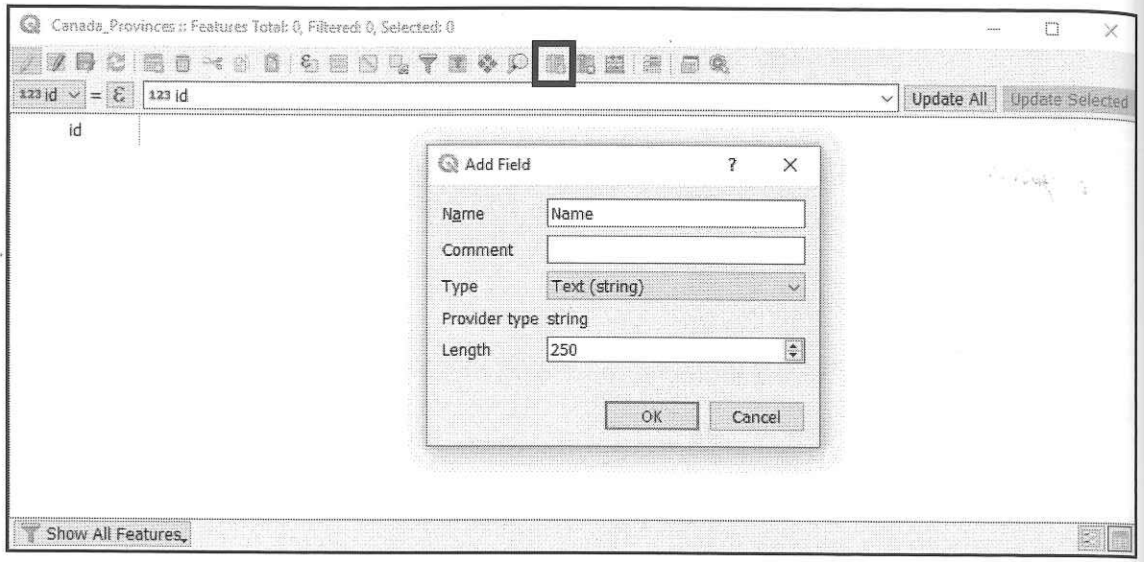
The Canada provinces ShapeFile added to the layers panel

If, like me, you didn't define any fields when you created this Shapefile, when you open the attribute table (right click | **open** attribute table, or click on the attribute table icon in the attribute toolbar), you will just have an **ID** field and an empty table. To change this, we will need to begin an editing session.

Editing tools

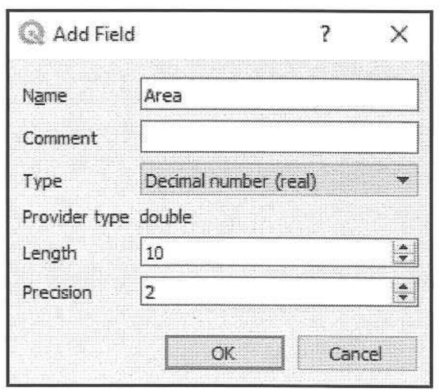
The editor toolbars we saw earlier in `Chapter 1`, *Where Do I Start?* If they are not displaying, select **View** | **Toolbars** | **Digitizing** for the Digitizing toolbar. Repeat for the advanced digitizing and shape digitizing toolbars. The following editing example also applies to any vector format, including the GeoPackage format. For this example, we will work with the Shapefile we previously created.

Make sure your newly created Shapefile is the selected layer in the **Layers** windows, then click on the pencil, and you will start your editing session. Before we start drawing, let's add a couple of fields that we can assign data to. Open the attribute table and click on the **Add Field** button (highlighted in the following screenshot), create a new **Text** field called `Name`, and select **OK**:



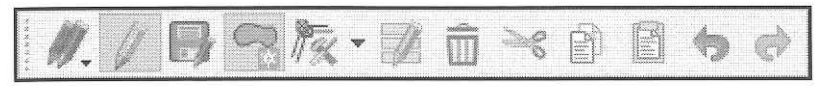
Adding a field

Add another field called `Area`, then set the **Length** to `10` and the **Precision** to `2`:



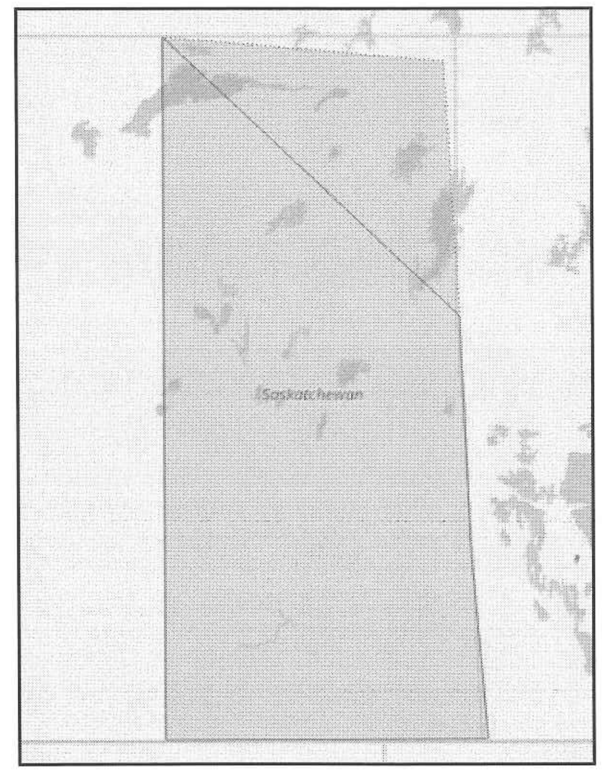
Defining the new field

Click on **OK**, and you should now have three attribute fields, **id**, **Name**, **Area**, which we will populate. Close the attribute table. We will now digitize a Canadian province. In the Digitizing toolbar, click the Add Polygon Feature button, shown here as a green polygon with a yellow star underneath:



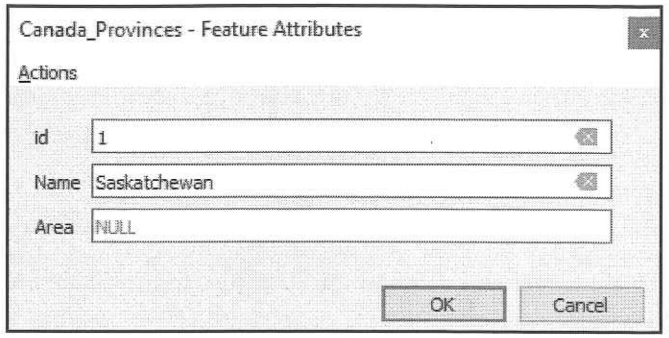
The editing toolbar with the add polygon feature button selected

As your cursor hovers over the map, it will look like a target with crosshairs. To begin digitizing, left-click with the mouse. When you have finished, right-click to complete the shape. The shape you are drawing, in our case the approximate province of **Saskatchewan**, will be shown in a faint red:



Drawing a new feature

After we have right-clicked and completed the polygon, the feature-attribute form appears. We are going to populate the **id** and the **Name** field:

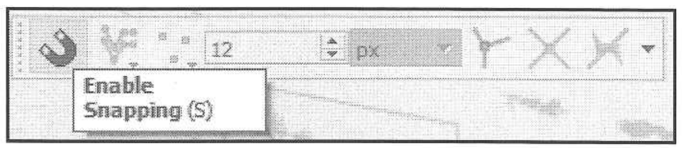


Creating new attributes

Click on **OK** to complete the entry, and then click on **Save** in the digitizing toolbar. Let's create another province. This time, we will use snapping tools to make sure we don't leave any gaps between the two polygons.

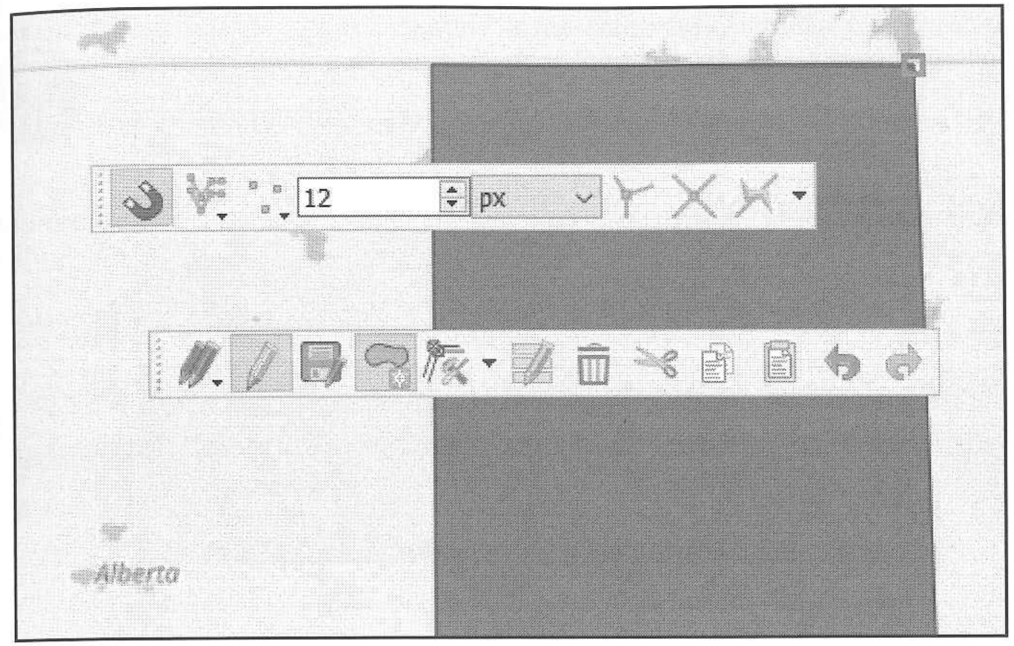
Snapping

Enable the snapping toolbar from the **View** menu and click on the magnet icon in the toolbar:



Enable snapping tool

Click on the Add Polygon Feature button on the digitizing toolbar, and as you move your mouse to begin digitizing, you will notice a pink square. This indicates that when you left-click, it will snap to the existing node. It should look similar to the following screenshot:

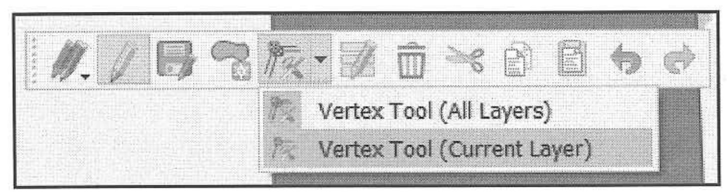


Snapping node highlighted

Continue editing as you would normally and assign the **ID** as `2` and the **Name** as `Manitoba` (assuming you are digitizing the province of Manitoba). Go ahead and capture Alberta using the same method and assign the **ID** to `3`.

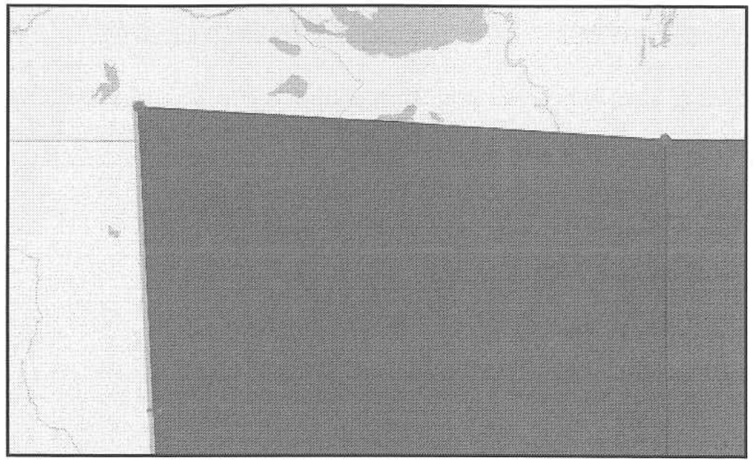
Mistakes and correcting with editing

When editing and creating geometries, mistakes often occur. Mistakes such as a misplaced click or a missed snapped point are common errors that you could introduce. A simple way to correct any poorly placed nodes is to use the vertex tool when editing has been toggled on. Click on the Vertex Tool and select the **Vertex Tool (Current Layer)** option:



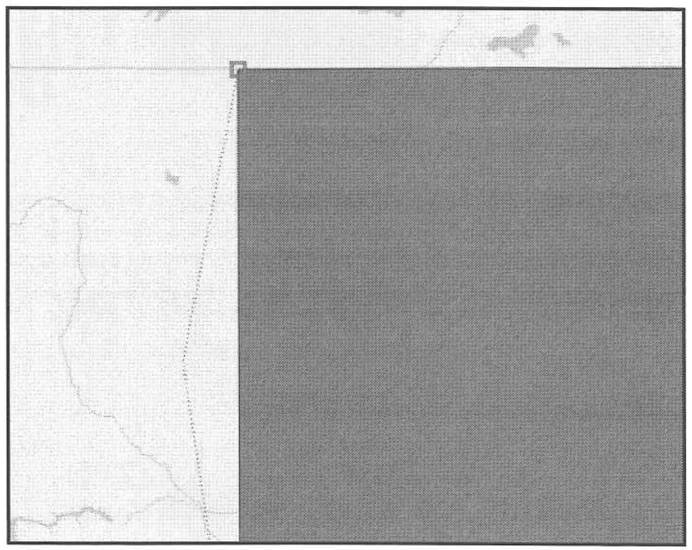
The editing toolbar with the vertex tool selected

As you move your mouse over the feature you want to edit, the nodes will show up as red dots. Left-click once on the red dot to select the node and then reposition it as required:



Editing a segment

You can add a node by double-clicking on the segment you wish to add to the node, and then dragging this to the desired location. A dashed red line will appear on the screen as you move your mouse. This is shown in the following diagram:

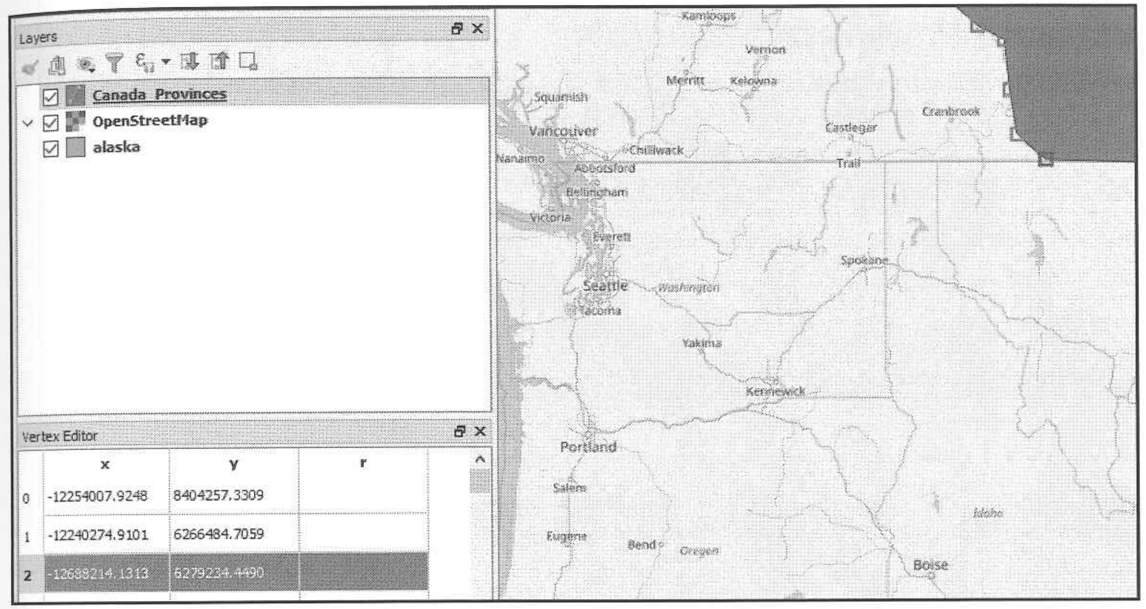


New node added to the feature

TIP

Alternatively, when the mouse cursor hovers over a line segment, the editing tools also display a potential new node candidate at the center of the line. This candidate node can be clicked and moved to a new location. This should be less error-prone than adding new nodes by double-clicking.

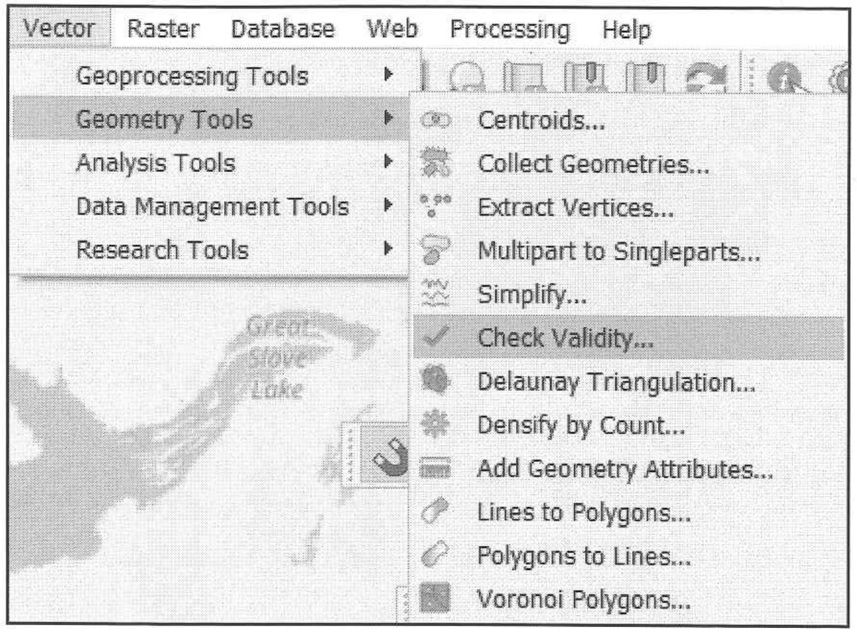
Finally, you can edit the coordinates of each node individually if necessary. Left-click on the feature and select **Vertex editor** from the pop-up menu. This will open another panel below the layers browser, which lists the coordinates; navigate to the point you wish to edit and make the adjustments in the corresponding table. This is shown here:



Vertex editor panel opened

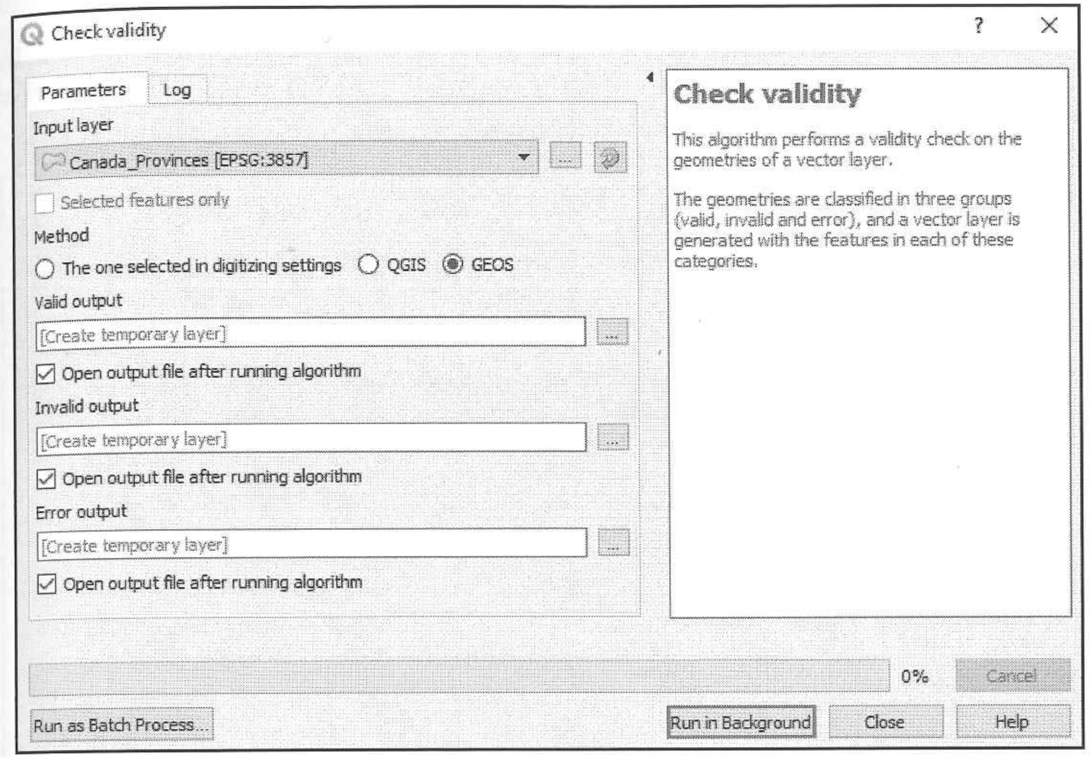
This is the most time-consuming way of editing nodes, and is not recommended. Use the snapping tool alongside the **Vertex** tool for best results.

Finally for this section, if you wish, you can use the **Check Validity** tool via the **Vector | Geometry Tools** menu:



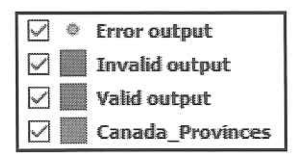
Checking the validity of the geometry

This will bring up an interface that reports any errors topologically:



Check the validity tool dialog box

Running the **Check validity** tool (shown in the preceding screenshot) will create temporary files that will be displayed as new layers in QGIS (shown in the following screenshot). These can guide your corrections:

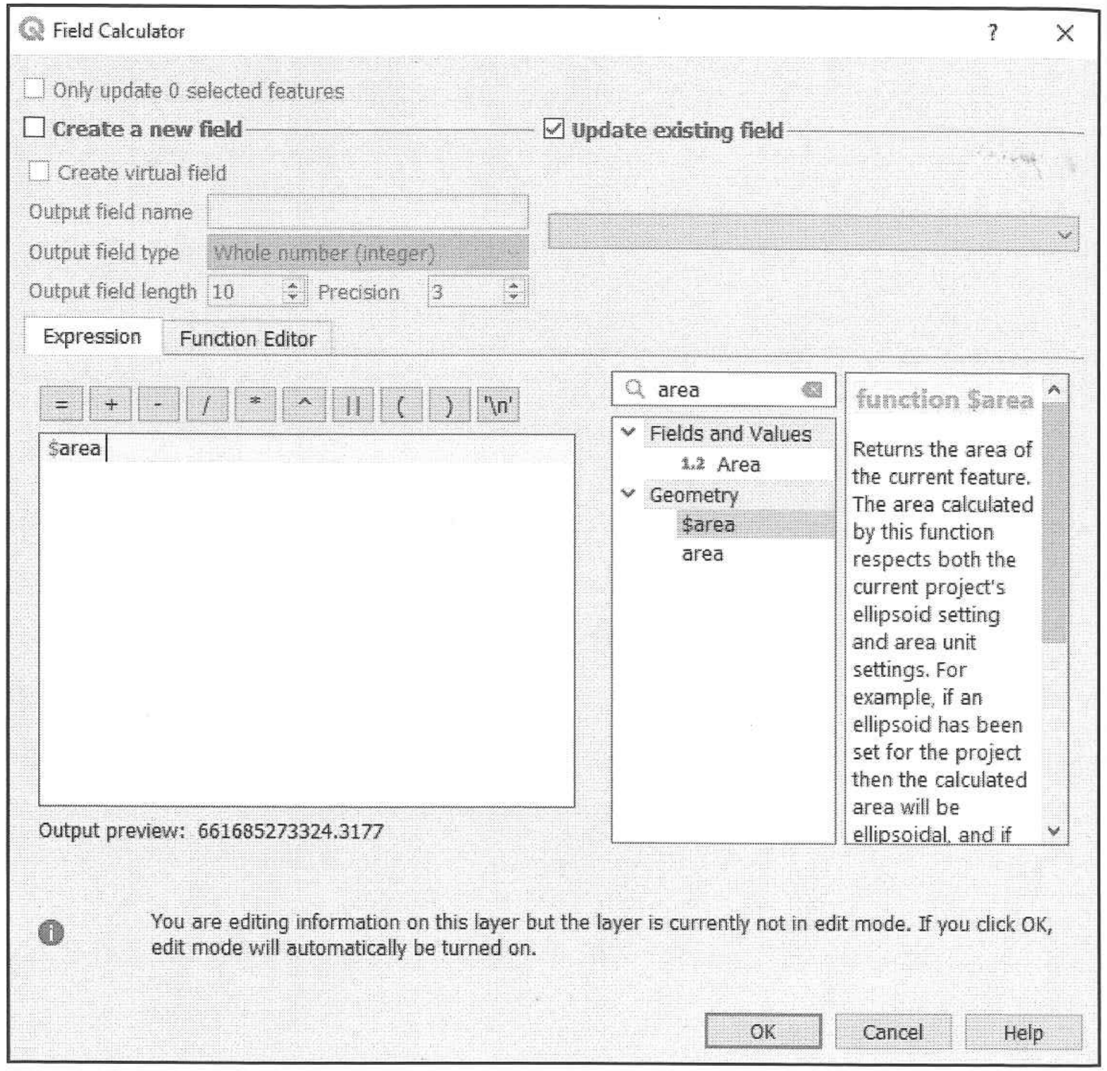


Output in the layer panel

There are further tools to help clean up editing errors that we will explore in `Chapter 6`, *Extending QGIS with Python*.

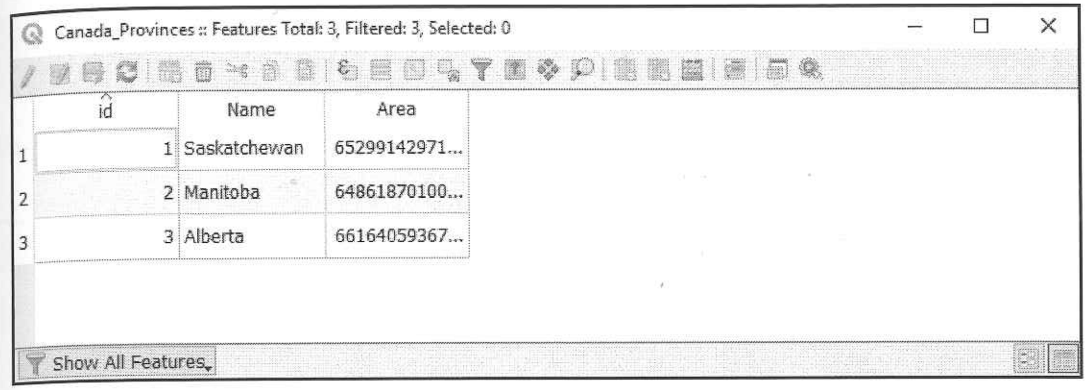
Populating attribute data

Finally, let's populate the **Area** field. Open the attribute table and click on the **Field Calculator**. We will update the **Area** field and use the **$area** from the **Geometry** section. In the following screenshot, we have searched for an area and doubled-clicked **$area** to move it to the **Expression** window:



Field calculator to calculate the area

When done, click on **OK** and look at the attribute table. It will not be exactly the same as mine, as you have manually digitized these areas. You should have values populated, and this will look similar to the following screenshot:



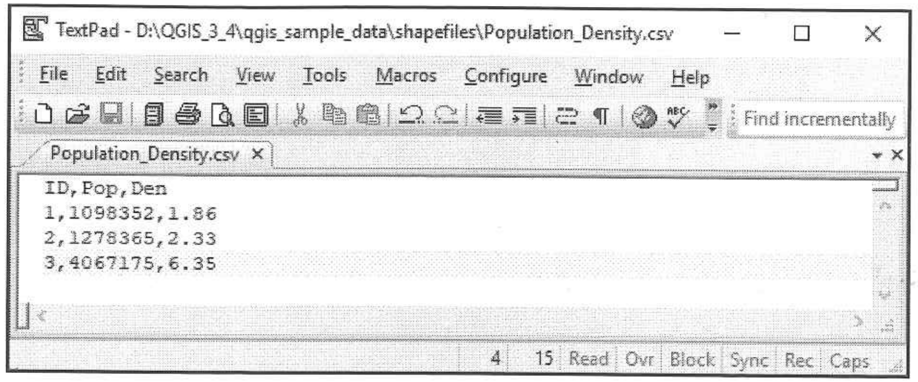
Attribute table with updated area values

We will discuss how to use these and other attributes to symbolize our data in `Chapter 4`, *Creating Great Maps*. We will then explore how to use these symbolized layers to create maps in `Chapter 5`, *Spatial Analysis*.

Data joins

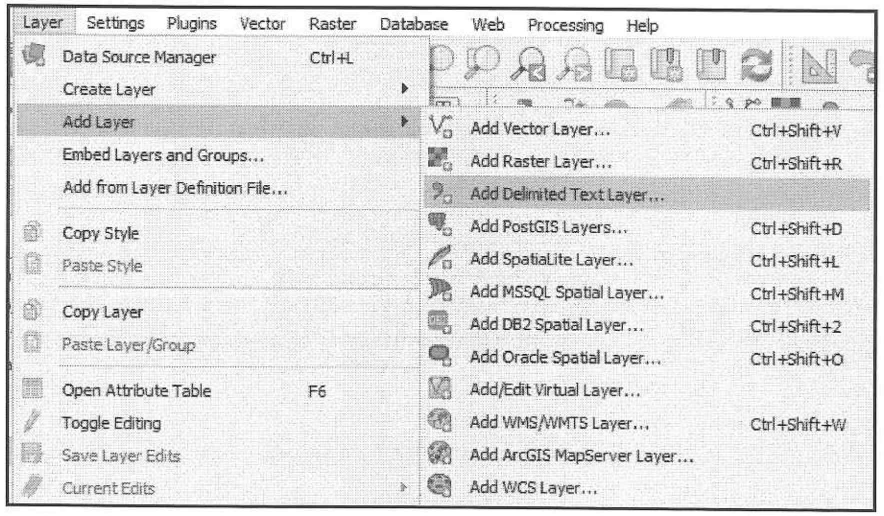
Data joins serve an important function in any database. It often means that we can join data with a common identifier, extending the original data. In this section, we will create a simple table and join it to the `Canada_Provinces` Shapefile that we have just created.

Using the data on the corresponding Wikipedia pages for each of the three provinces that we have captured, I have created the following `.csv` file:



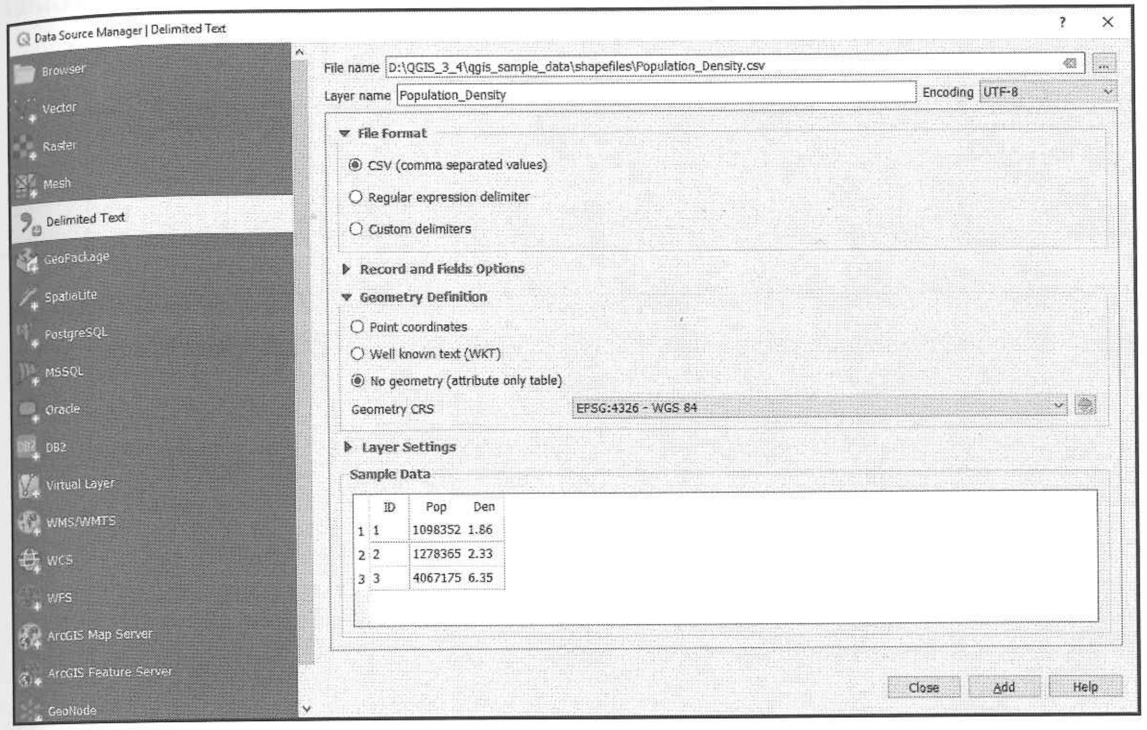
Create a simple .csv file to join to our Canadian provinces data

Here, I have kept the **ID** the same as the one we created in our Shapefile. I have saved this `.csv` file in the same place as my `Canada_Provinces.shp`. Create this layer in a text editor of your choice. Then add this text layer by clicking **Layer** | **Add Layer** | **Add Delimited Text Layer**:



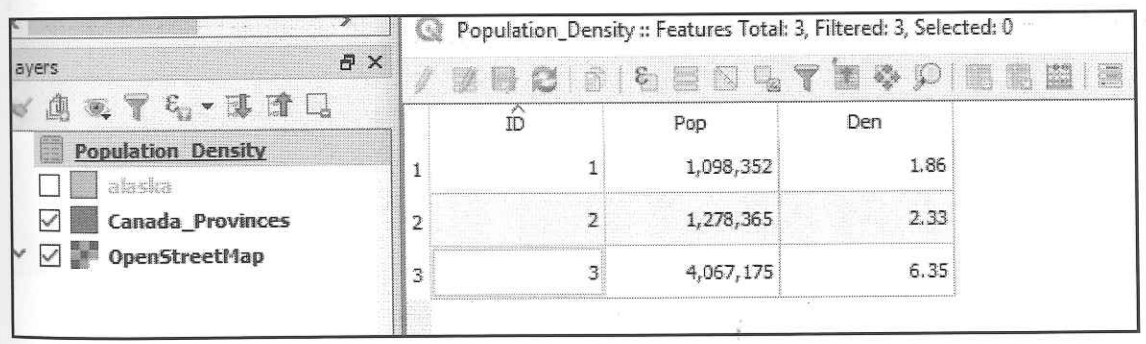
Add Delimited Text Layer

In the resulting dialog box, make sure in the **Geometry** definition you select **No Geometry (attribute only table)**:



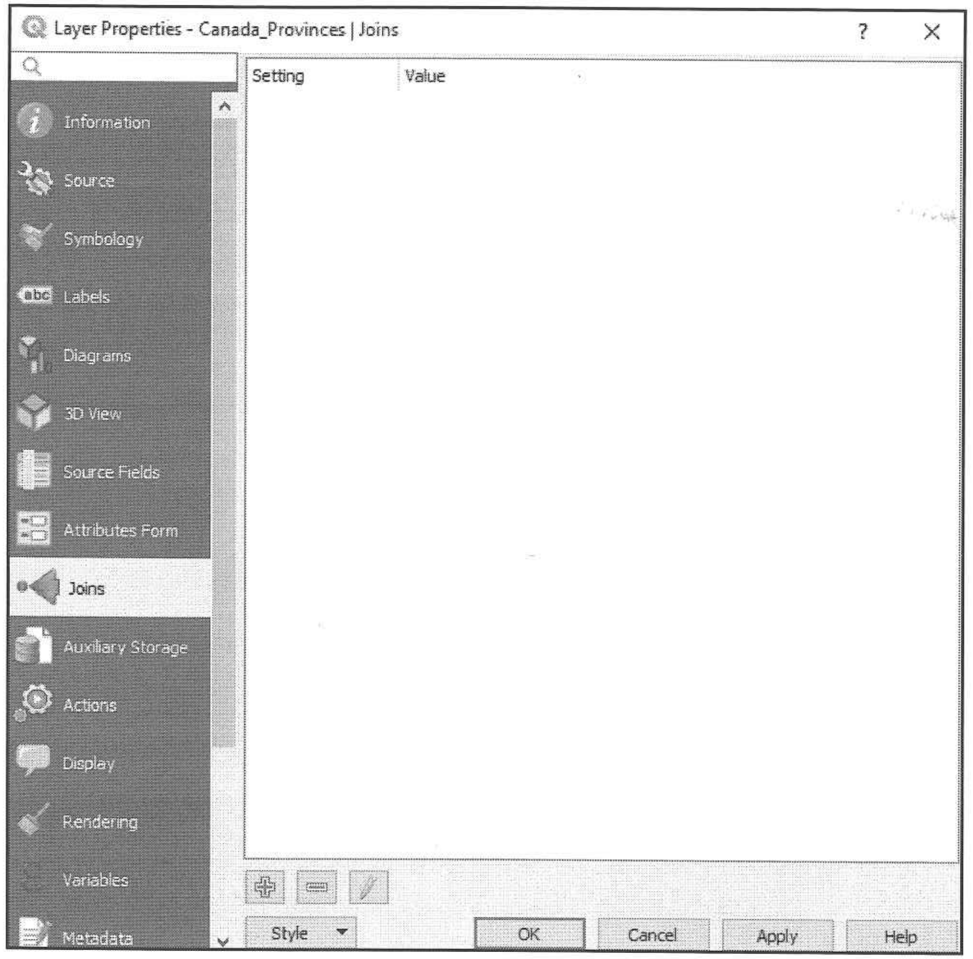
Data source manager to add non-spatial data

Click **Add**, and the csv will be added to the layers panel. If you open the attribute table, your screen should look similar to the following screenshot:



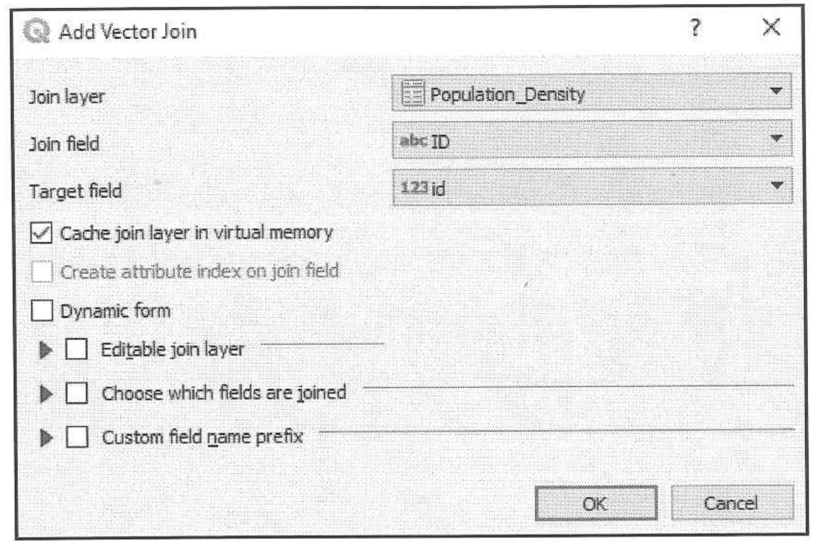
The newly added csv file in the layers panel

To join this data to our **Canada_Provinces** layer, close the **Population_Density** table, right-click on the **Canada_Provinces** layer, and select **Properties** | **Joins**. You will have the **Layer Properties** dialog box open (which we will explore extensively in `Chapter 4`, *Creating Great Maps*), which should look similar to the following screenshot:



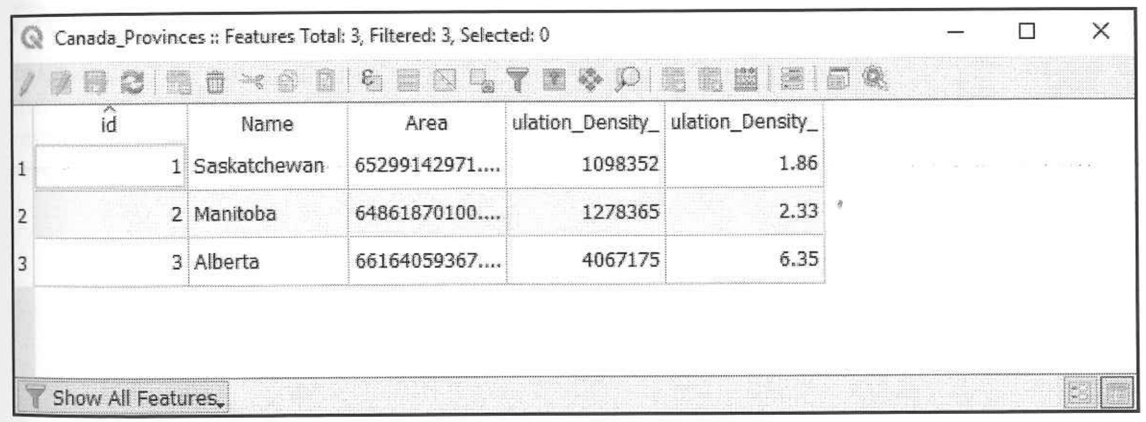
Layer properties Joins option

Click on the green **+** button in the bottom left and set up the join layer as the **Population_Density** layer, the **Join field** as **ID**, and the **Target field** as **ID**, and then click **OK**. The join dialog box is shown here:



Add Vector Join dialog box

Close the **Layer Properties** dialog box and reopen the attribute table for the **Canada_Provinces** layer. The population data will now be joined:



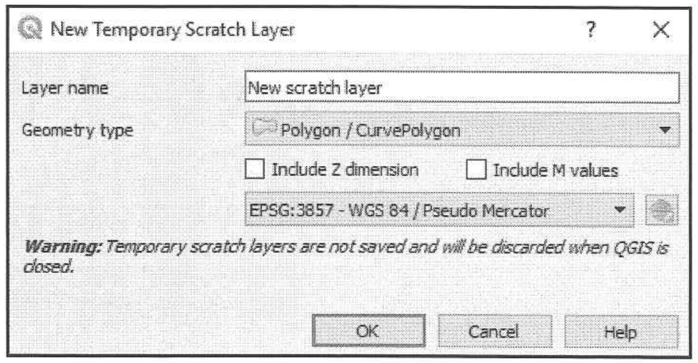
The attribute table with the joined data

When joining data, care is needed to ensure that there is a common join. This process of data wrangling is often the slowest part. However, the reward is an expanded dataset that we can then use to symbolize and build maps.

Finally, if you want to make this join permanent, you need to save the **Canada_Provinces** layer as a new Shapefile. Right click on the **Canada_Provinces** layer, select **Save As**, set the new name as `Canada_Provinces_population.shp`, and click **OK**. The newly created layer will be added to the **Layers Panel** and the map. When you right-click to inspect the attribute table, the field names will have been abbreviated to fit in with the field name restrictions of the Shapefile format.

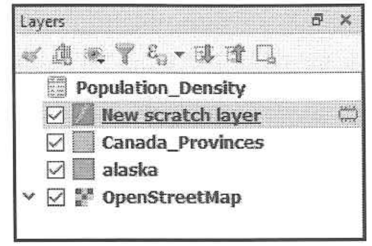
Using temporary scratch layers

When you just want to quickly draw some features on the map, temporary scratch layers are a great way of doing that. The benefit is that you don't need to think about file formats and locations for your temporary data. Go to **Layer** | **Create Layer** | **New Temporary Scratch Layer** to create a new temporary scratch layer. As you can see in the following screenshot, all we need to do to configure this temporary layer is pick a **Geometry type**, a **Layer name**, and a CRS. Once the layer is created, we can add features and attributes as we would with any other vector layer:



Creating a New Temporary Scratch Layer

As the name suggests, temporary scratch layers are temporary. This means that they will vanish when you close the project, so use them with care. A temporary layer will appear with a circular icon on the right next to it in the Layer Panel. This will remind you which layers will be lost when you close the project. This is shown here:



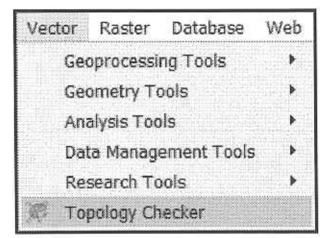
You can convert a scratch layer to a permanent layer by right-clicking and saving it as a new layer, just as we did with the joins. Alternatively, you could install the memory-saver plugin, which will save all the temporary layers with the project.

Checking for topological errors and fixing them

Sometimes the data that we receive from different sources or data that results from a chain of spatial processing steps can have problems. Topological errors can be particularly annoying since they can lead to a multitude of different problems when using the data for analysis and further spatial processing. Therefore, it is important to have tools that can check data for topological errors and that know ways to fix discovered errors. Care must always be taken when changing topological errors on batch, so make sure any corrections work on a few geometries before scaling across all your data.

Finding errors with the Topology Checker

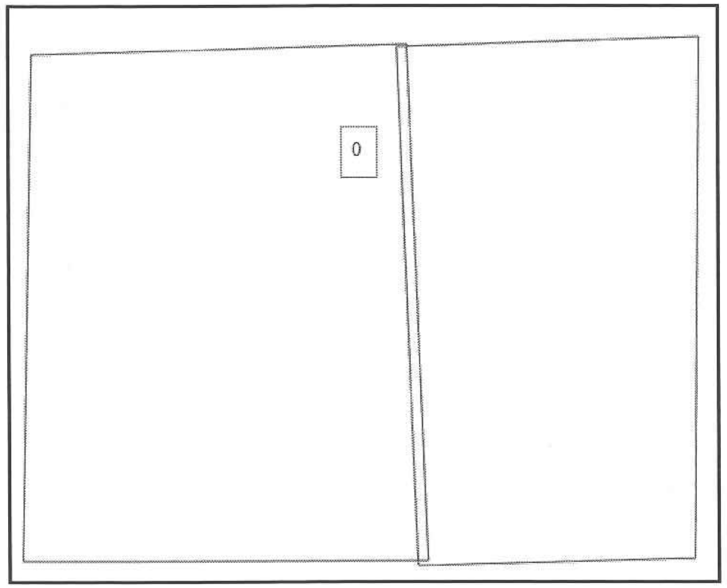
In QGIS, we can use the Topology Checker plugin; it is installed by default and is accessible through the **Vector** menu **Topology Checker** entry (if you cannot find the menu entry, you might have to enable the plugin in **Plugin Manager**). This shown in the following screenshot:



Topology Checker in the Vector menu

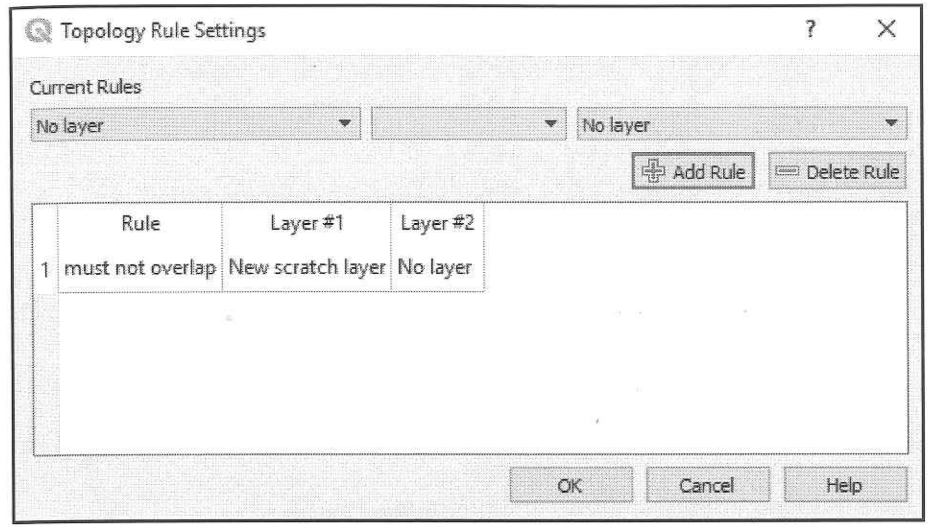
When the plugin is activated, it adds a **Topology Checker** panel to the QGIS window. This panel can be used to configure and run different topology checks and will list the detected errors.

To see the **Topology Checker** in action, create a temporary scratch layer with polygon geometries and digitize some small polygons that overlap. As you can see from the following screenshot, we have made the fill transparent. Right-click on the layer, select **Properites** | **Symbology**, and then set the Fill Color to transparent (`Chapter 3`, *Visualizing Data,* will help if you are unsure about how to do this). This overlap is highlighted in the following screenshot:



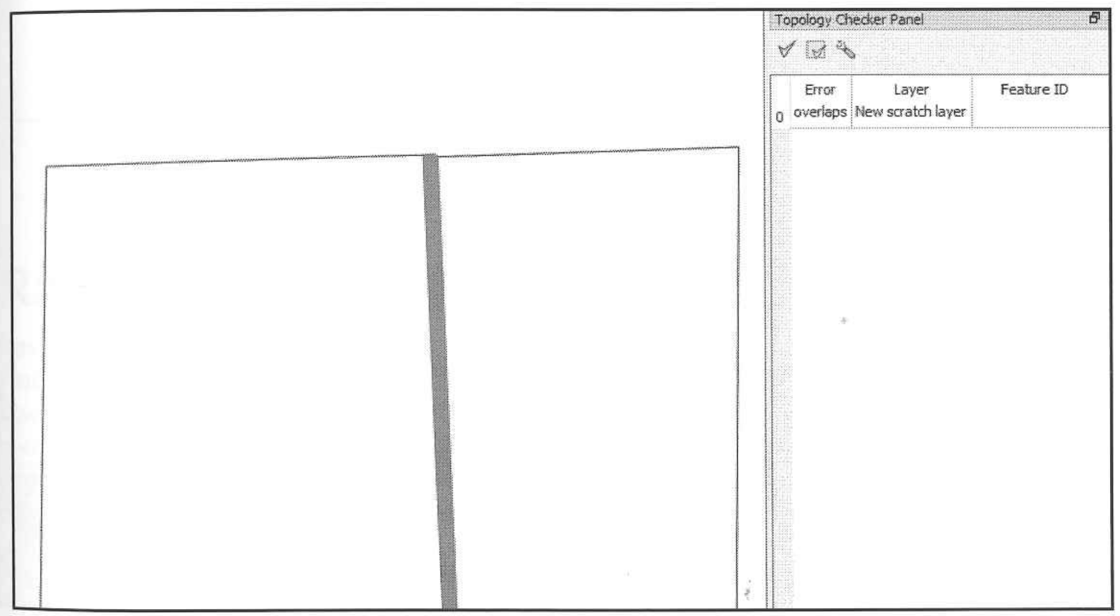
Overlapping topology

In the topological checker, create a new rule by clicking on the configure button which (looks like a wrench tool button). Set a rule to say that the **New scratch layer must not overlap** each other. This is shown in the following screenshot:



Setting the Topology rules

Click on **OK**. In the **Topology Checker** panel, click on the yellow check mark to validate the rule. The overlap we have created will return one error highlighted in red. This is shown in the following screenshot:

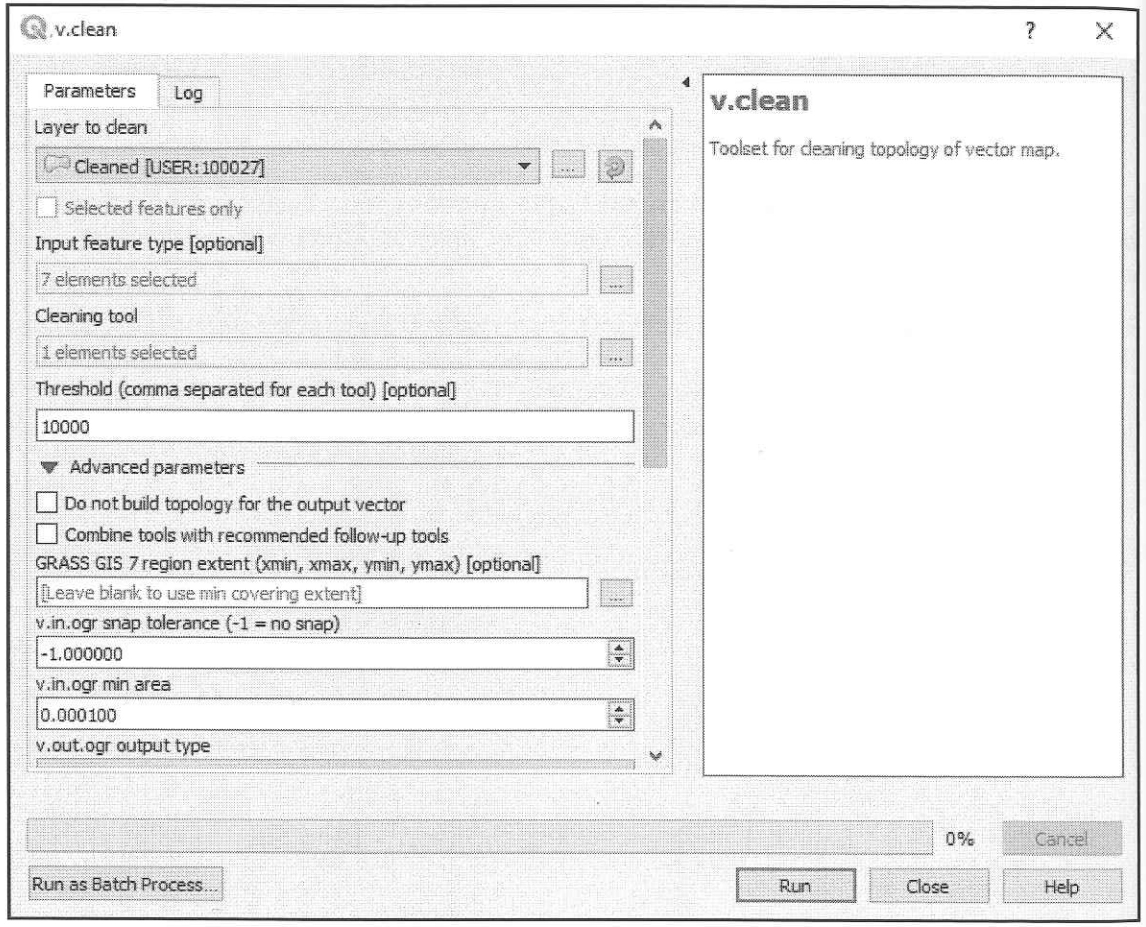


Highlighted error based on the rule set

Now that we have highlighted an error, we need to fix it. In the next section, we will use the v.clean tool to do this.

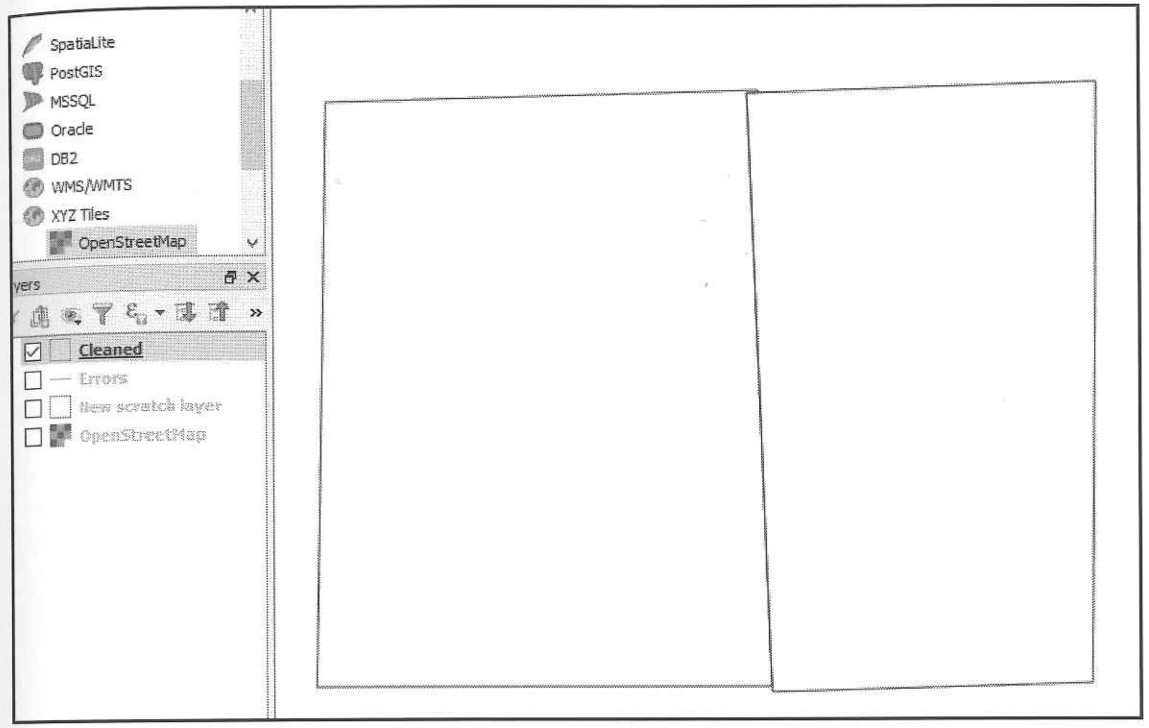
Fixing invalid geometry errors

To get rid of these overlapping polygons, we can use the **v.clean** tool. Search for **v.clean** in the **Processing Toolbox** | **Cleaning tool** option and set to **rmarea** (meaning remove area), which is also available through the **Processing Toolbox**. In the example shown in this screenshot, the **Threshold (comma separated for each tool) [optional]** value of `10000` tells the tool to remove all polygons with an area less than 10,000 square meters by merging them with the neighboring polygon with the longest common boundary:



Fixing the errors with v.clean

Check the size of your overlap area and adjust this value accordingly. Click on **Run** when you are ready to run the **v.clean**. The resulting layer is called **Cleaned**. My new data looks like the following screenshot:

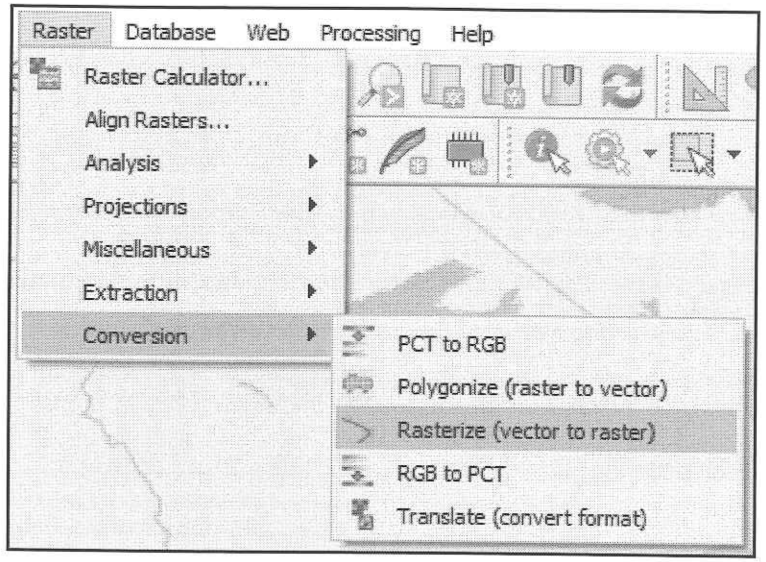


The resulting cleaned layer

Raster data

Raster data is not created like vector data. Quite often, raster data is created from other raster data, such as creating a slope raster from a topographic raster. Raster data can also be created from vector data, which can be done by gridding a series of coordinates/points. QGIS has the ability to perform all of these functions. We will further explore raster visualization in `Chapter 4`, *Creating Great Maps*, and processing rasters in `Chapter 6`, *Extending QGIS with Python*.

In this section, we will convert our `Canada_Provinces.shp` to a raster. In the menu, click **Raster** | **Conversion** | **Rasterize (vector to raster),** as shown in the following screenshot:

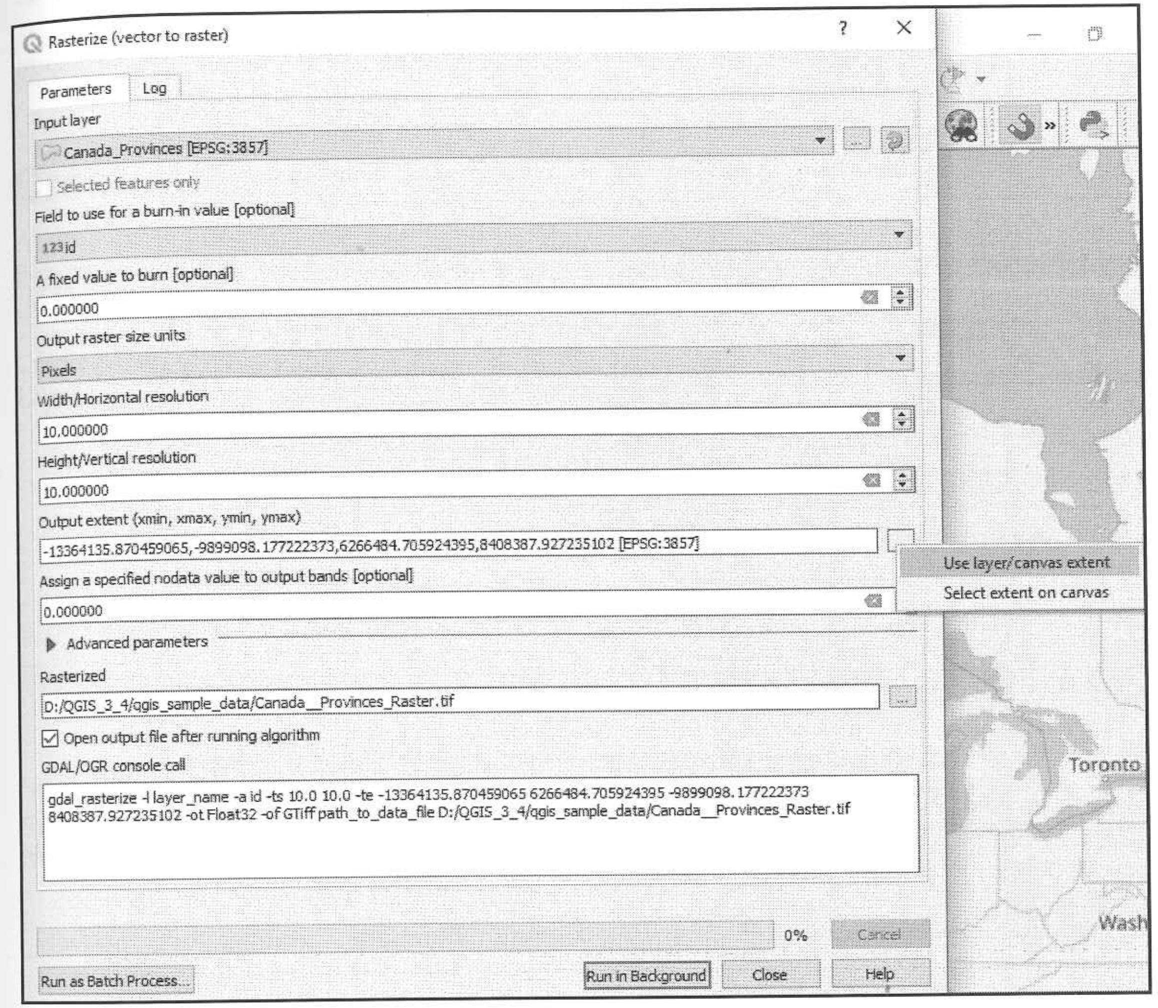


Vector to raster

In the following dialog box, enter these parameters:

- **Input Layer**: `Canada_Provinces`.
- **Burn-in value**: **id.**
- **Output raster size units**: **Pixels.**
- **Width**: `10` (this will create a raster with a width of 10 pixels).
- **Height**: `10` (this will create a raster with a height of 10 pixels).
- **Output extent**: Select the **Use layer/canvas** and then click on `Canada_Provinces` to auto-populate.
- **Rasterized**: Save the `Canada_Provinces_Raster.tif` filename.

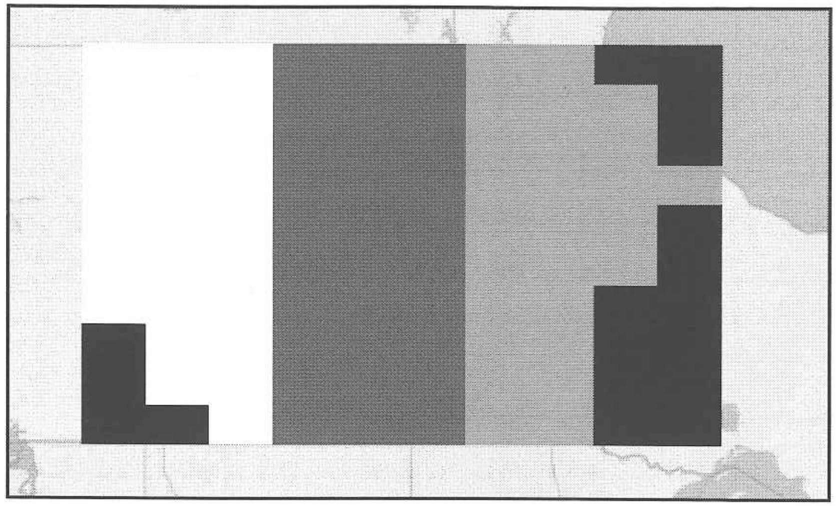
Your screen should look similar to this:



Rasterize the vector layer dialog box

Now select **Run in Background** (a new feature of QGIS 3). Once this has been done, the dialog box will remain open. You can close this down if no errors are reported.

The result will be shown in the map, and this should be similar to the following screenshot:



Raster result

We will look at shading and displaying Raster datasets in `Chapter 3`, *Visualizing Data*.

> TIP: Did you notice that the bottom of the Rasterize dialog box contained a GDAL command? You could copy and paste this into the `OSGeo4W` shell and execute the command there: `gdal_rasterize -l layer_name -a id -ts 10.0 10.0 -te -13364135.870459065 6266484.705924395 -9899098.177222373 8408387.927235102 -ot Float32 -of GTiff path_to_data_file D:/QGIS_3_4/qgis_sample_data/Canada__Provinces_Raster.tif`. Look out for this command in later chapters. As you grow familiar with QGIS, you will begin to notice the `gdal` commands that are running in the background.

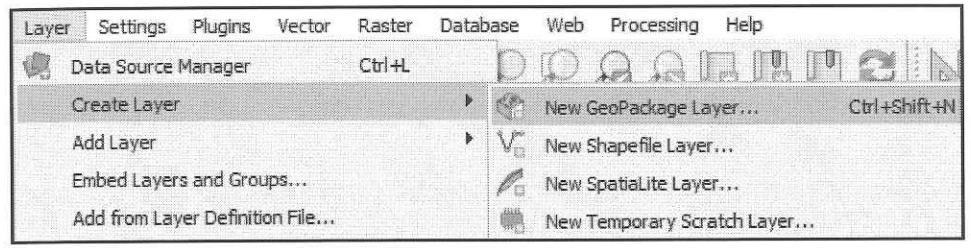
Other data

Finally, we will take a look at databases and the process of creating data in the new GeoPackage format.

Creating a GeoPackage

At the start of this chapter, we looked at data formats and the GeoPackage format. Remember that, in QGIS 3, the GeoPackage format is the default format. So, let's finish this chapter by creating data in the GeoPackage format:

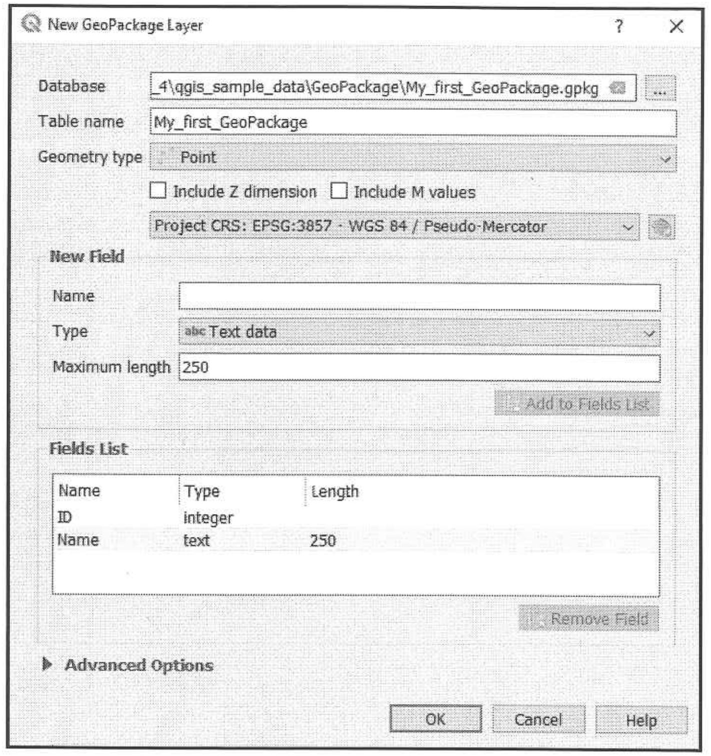
Create a new GeoPackage layer by clicking **Layer** | **Create Layer** | **New GeoPackage Layer...**:



Create a New GeoPackage Layer

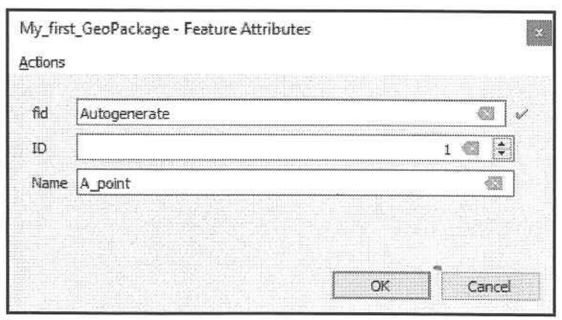
Now fill in the resulting dialog box. Let's create a GeoPackage called `My_first_GeoPackage`, set the **Geometry type** as a **Point**, and add the **ID** and **Name** fields.

It should look similar to the following screenshot:



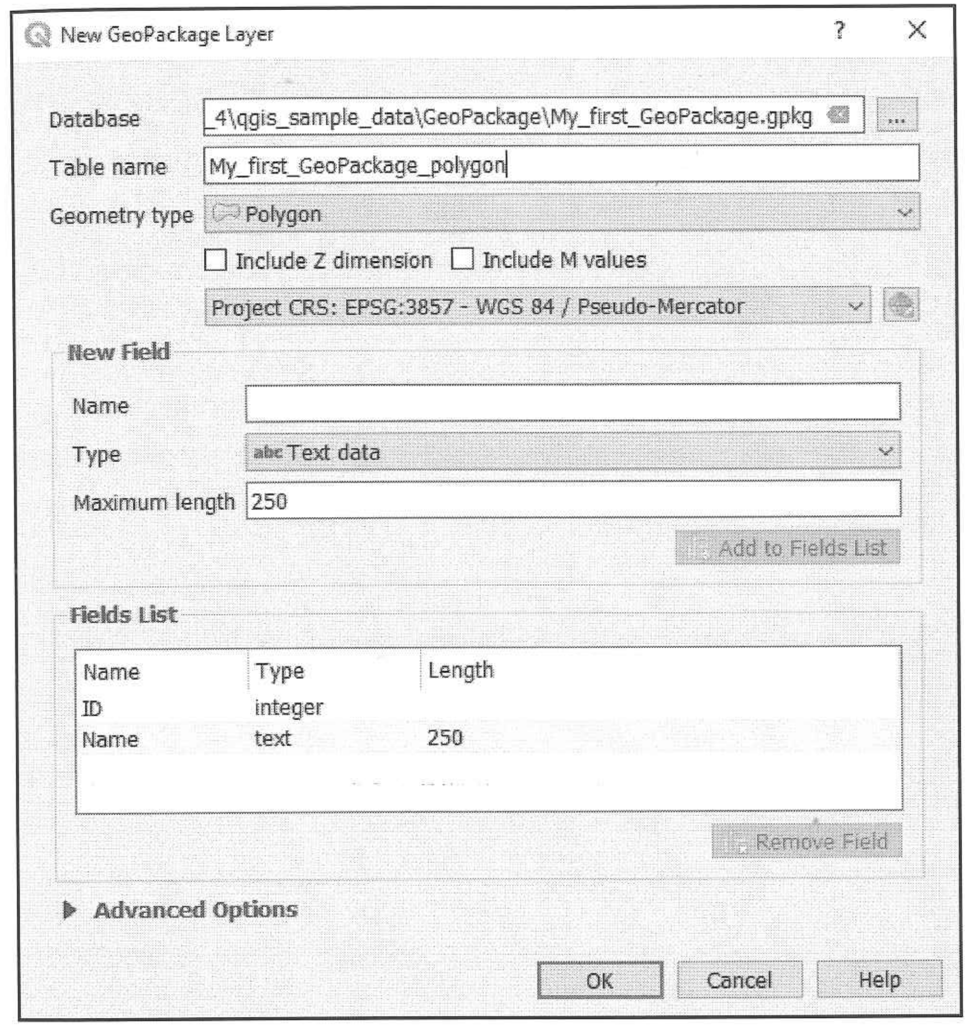
New GeoPackage Layer dialog box

Use what you have already learned in this chapter to select this layer, toggle editing on, and then capture a point. Fill in the associated attributes, which are similar to what we can see in the following screenshot:



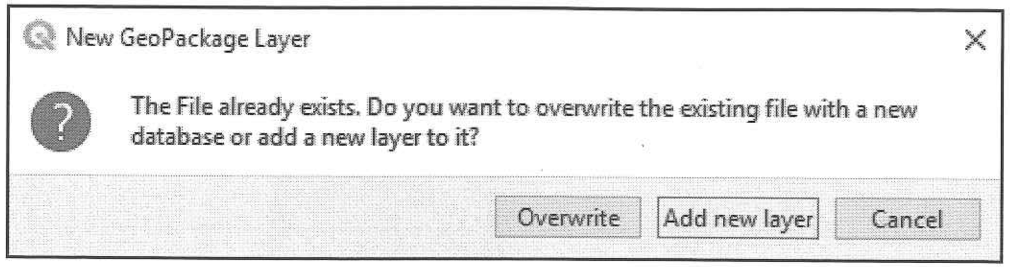
Adding feature attributes

Click **OK** to confirm this, and you have officially created your first point layer in a GeoPackage. You can add another layer with a different geometry type within the same GeoPackage. This is a change from working with a Shapefile. You can create a new GeoPackage layer by clicking **Layer** | **Create Layer** | **New GeoPackage Layer**, but this time, point it at the existing GeoPackage. Create a new **Table name**, select **Geometry type** as **Polygon**, and add fields:



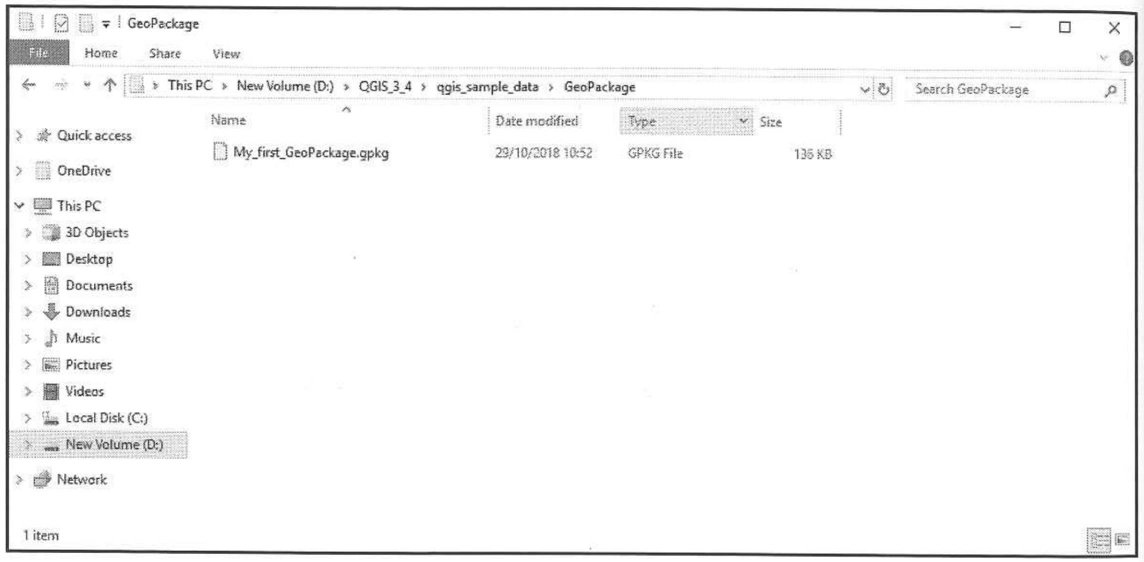
Adding a new layer to the GeoPackage

You will get an overwrite message, as shown in the following screenshot. Here, click on **Add new layer**:



Add new layer dialog box

Edit your new layer as we have previously with a Shapefile. This can be repeated for different layers with different feature types. After saving all of your associated files and projects, close down QGIS and then navigate to the folder where you created your GeoPackage. Here, you will see just one file, irrespective of the number of layers you have added to the GeoPackage. Compare that to the folder you saved your Shapefile in. The following screenshot shows one file for many layers with different geometries:

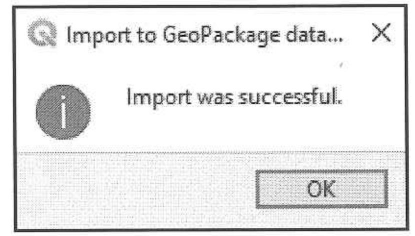


The GeoPackage in the folder directory in Windows

Exporting to a different format

Reopen QGIS and load in a layer from your GeoPackage. Right-click on the layer in the **Layers** Panel and select **Export** | **Save Features As.** In the resulting dialog box, you can choose the format you wish to export to. This process is the same for any Vector or Raster layer.

If you wish to export a Shapefile into a GeoPackage, you could, in the **Browser** Panel, click and drag the Shapefile into your GeoPackage. This is a super simple layer to import data into a GeoPackage:

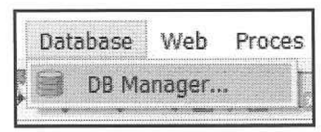


Importing into GeoPackage

Your original Shapefile will still exist. You have just imported a copy.

Spatial Databases

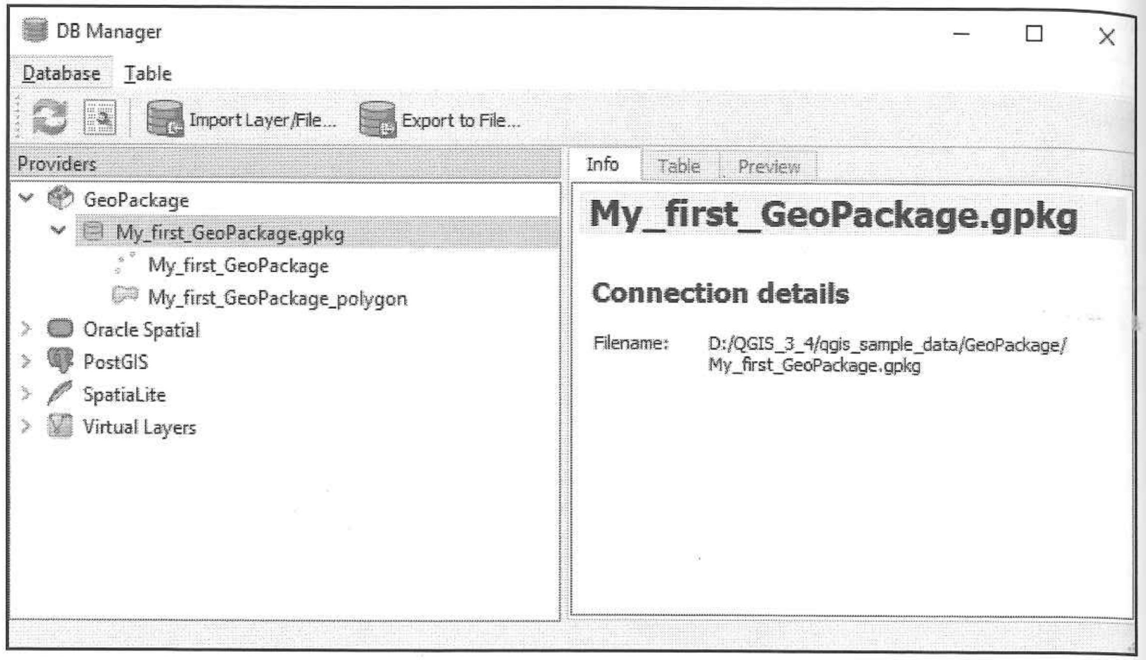
Let's conclude this chapter by briefly mentioning Spatial Databases. DB Manager is a great way of handling connections with Spatial Databases, including PostGIS, Oracle Spatial, SpatiaLite, and GeoPackage. To open the **DB Manager**, click **Database** | **DB Manager**:



Opening the DB Manager

You will be presented with the ability to connect to all of the databases listed above. To connect, right-click on the database. A GeoPackage is a database container, which means it supports direct use. In the DB Manager, let's connect to our newly created GeoPackage layer. Right-click on GeoPackage and select **New Connection**. Navigate to the location in which you stored it, and then click **OK**.

You should see the connection displayed in the following screenshot:



The DB Manager

You can drag and drop these layers into QGIS. You can then upload new data, export existing data, and run SQL on the data. The DB manager performs the same function for all the GIS database types previously listed. If you have access to a database, such as PostGIS, then the DB manager is a very convenient place in QGIS to interact with the data.

Summary

In this chapter, you learned the basics of using data in a GIS. QGIS 3.4 supports all the formats in OGR and GDAL. The default format for QGIS 3.4 is the GeoPackage. We also looked at creating and editing Vector data, as well as attribute tables and geometries. Furthermore, we joined data and used snapping tools to preserve topology. Don't ignore Raster data though; we will use it many times throughout this book. In this chapter, we briefly reviewed its creation and hinted at the powerful GDAL tools built into QGIS. Finally, we looked at spatial databases in QGIS 3.4 and connected to them in DB Manager.

In the next chapter, we will look at styling and visualizing this data.

3
Visualizing Data

In this chapter, we will look at visualizing GIS data. We will build on the knowledge gained in `Chapter 2`, *Data Creation and Editing*, in which we learned how to load, create, and edit GIS data. QGIS automatically styles data when added to the map. This is useful for a quick inspection, but to convey more meaning, we need to style our data so that the information presented becomes more intuitive. That is what this chapter is all about.

The topics covered in this chapter are as follows:

- Styling data
- Interactive styling
- Styling Rasters
    - Styling Terrain data
    - Styling Satellite imagery
    - Raster toolbar
    - Styling landcover maps
- Saving styles
- Styling Vectors
    - Points
    - Simple markers and SVG
    - Lines
    - Polygons