

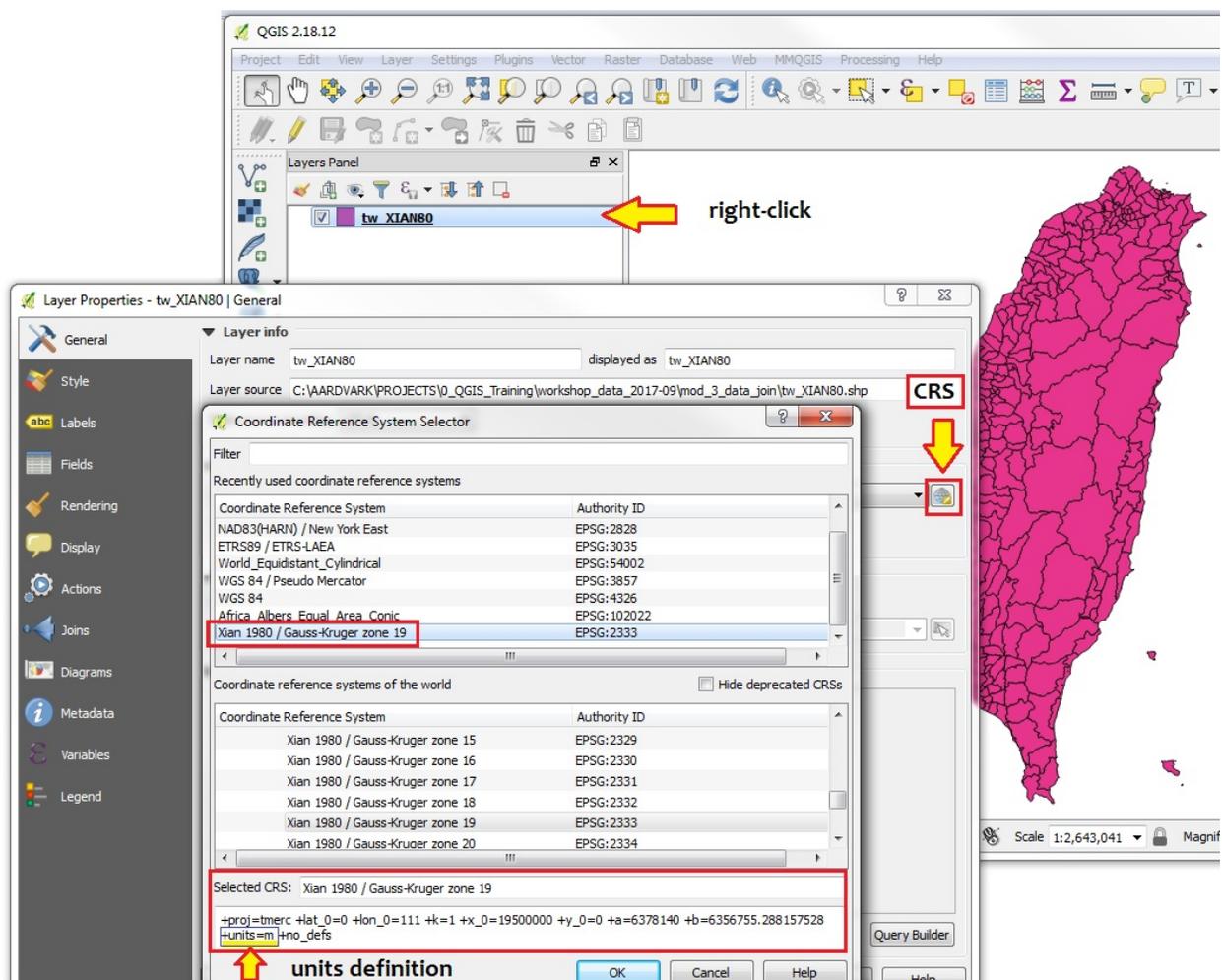
Module 3: Data Join

Conceptual Overview: The Data Join is used to related tabular data to an existing spatial data layer.

Exercise: In this example we take a polygon layer, containing all the ADM2 districts of Taiwan, and join to those polygons a table containing population data for those districts. This enables thematic mapping, as we saw in Module 2.

1. Open Example Data Layers

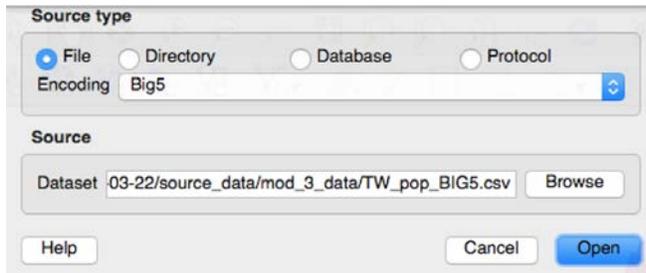
1.1 from the folder *mod_3_data_join* add the vector shapefile: *tw_XIAN80* (a polygon file for Taiwan districts). Right-click to open CRS settings, and note that EPSG:2333 (Xi'an 1980 Projection) is In **meters**



2. Check .csv file contents

2.1 Now we want to open tabular data file (that does NOT contain X, Y coordinates). Open the tabular data file called *TW_pop_BIG5.csv* using the ADD VECTOR DATA button (& set BIG5 encoding).

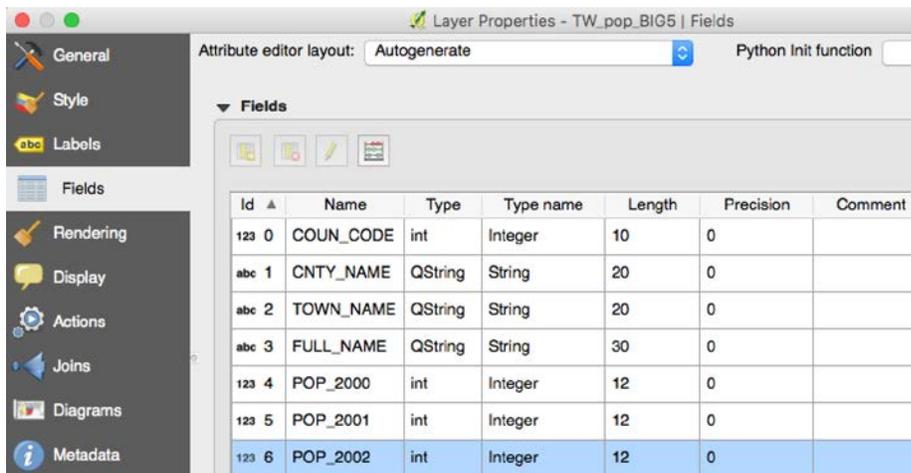




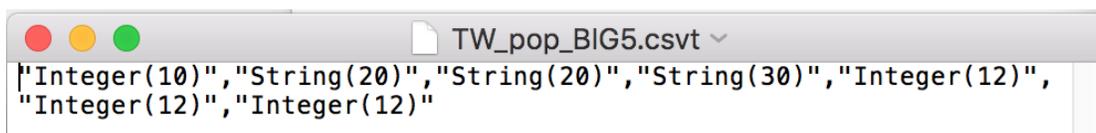
2.2 *right-click* to open the attribute table and inspect the content

	COUN_CODE	CNTY_NAME	TOWN_NAME	FULL_NAME	POP_2000
0	1000101	台北縣	板橋市	台北縣板橋市	529059
1	1000102	台北縣	三重市	台北縣三重市	382266
2	1000103	台北縣	中和市	台北縣中和市	398123
3	1000104	台北縣	永和市	台北縣永和市	228099
4	1000105	台北縣	新莊市	台北縣新莊市	372175

2.3 *right-click* the tabular data layer to inspect Properties | Fields. You want to make note of the field that will serve as the Join Key, in this case COUN_CODE, which is **Type = int Length = 10**

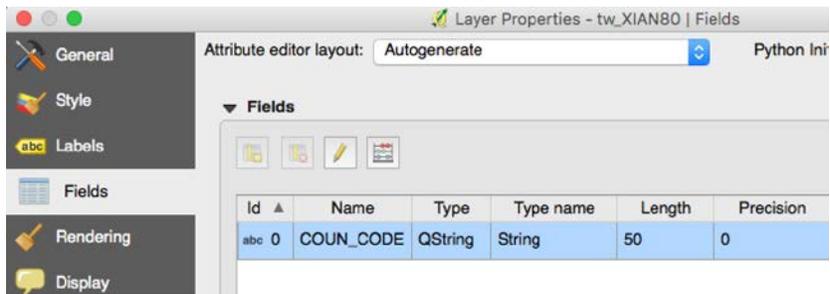


2.4 this Field format was set in the .csvt file. If you open that file in a text editor, you will see that you can control exactly how the fields are interpreted when QGIS opens them. This is extremely useful, for example, if you want to prevent integers from being ingested as String format.



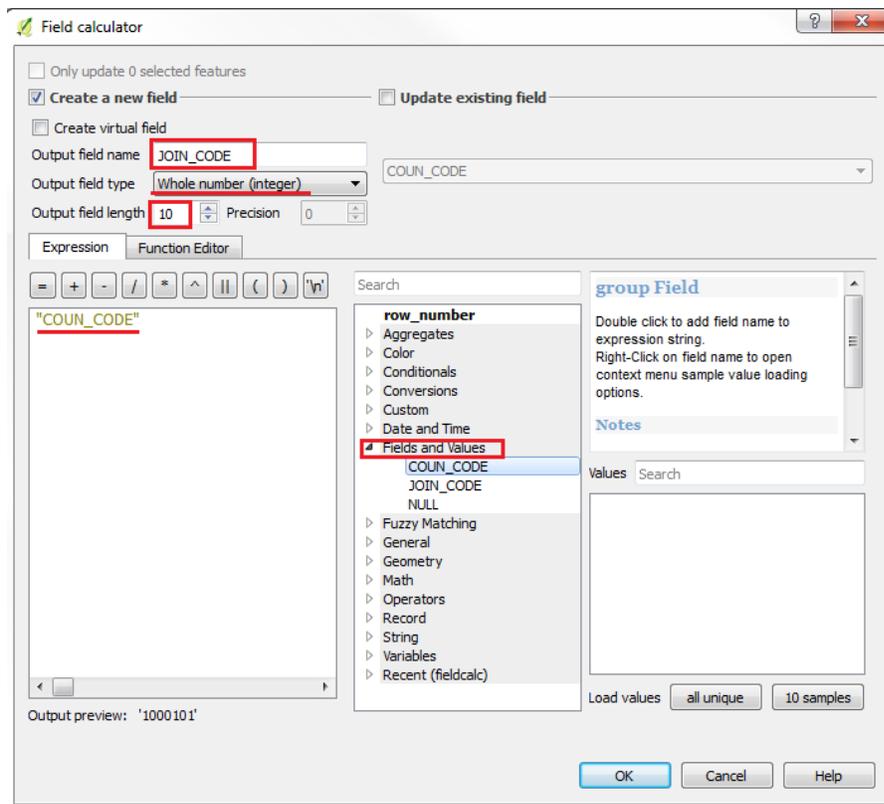
TIP, let's *right-click* on *TW_pop_BIG5.csv* and save as *z.csv*. Then remove *TW_pop_BIG5.csv*.

2.5 right click on `tw_XIAN80` layer and inspect the Properties | Fields



Now we have a problem! The format of the `COUN_CODE` in the polygon layer is **Type = String**. We should have an integer field of the `COUN_CODE` values.

3. We will right-click on `tw_XIAN80` layer to view the Attribute Table, and we want to TOGGLE EDITING, the pencil icon button. With the EDITING toggled on, the FIELD CALCULATOR (which looks like an abacus) will no longer be grayed out. Click on FIELD CALCULATOR to open the dialog box



3.1 first we accept default, Create a new field, and type a name in the Field Name: `JOIN_CODE`

3.2 next we set the OUTPUT FIELD TYPE, making sure it is set to Whole Number (Integer)

3.3 the Output Field Length can be set to 10

3.4 now we will insert the value from the COUN_CODE field by expanding the drop-down menu for FIELDS AND VALUES and double-clicking on the field name COUN_CODE. What this does is to enter the Field Name that we double-clicked into the Expression form. (If you know SQL you can write your own expressions here!)

3.5 Note that the Output Preview is already giving us a view that looks correct

3.6 Click OK to run the Expression Query, which in this case will create a new Field, JOIN_CODE and populate it with the values from COUN_CODE converted to integers.

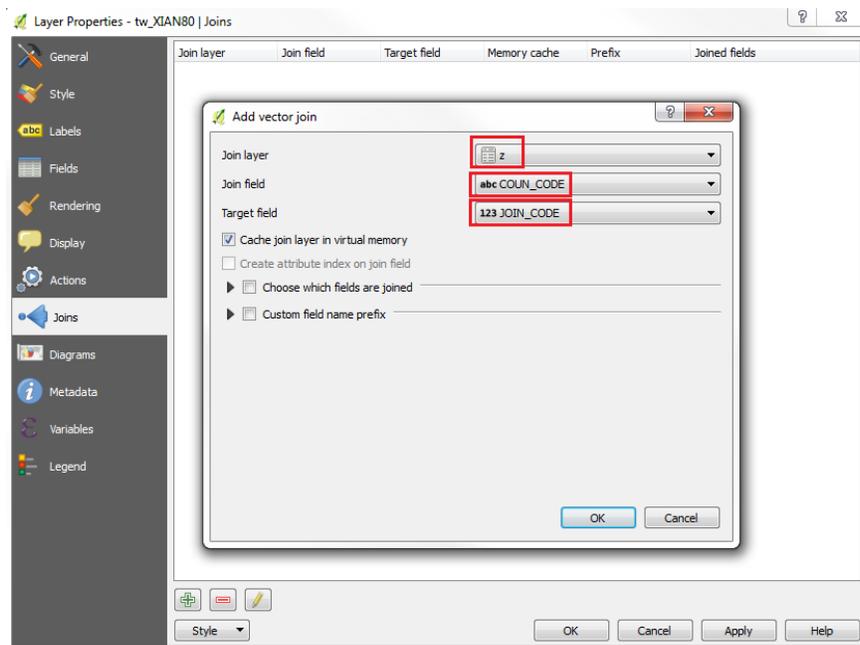
3.7 The Attribute Table should be updated with the query results. Notice that the TEXT (String) values align to the left side of the cells, while the NUMERIC (Integer) values align to the right edge. This is a good indicator that the query worked as expected.

	COUN_CODE	JOIN_CODE
0	1000101	1000101
1	1000102	1000102

3.8 click the TOGGLE EDIT button and hit SAVE to save the updated FIELD. Right click on the updated layer tw_XIAN80, and check Properties | Fields to double-check **JOIN_CODE: Type = int Length = 10**

4. Having confirmed that the TARGET LAYER (the Polygon Shapefile) and the JOIN LAYER (the tabular data) both contain a JOIN field in the SAME format, we can go ahead with the JOIN itself.

4.1 Right-click on the TARGET LAYER (tw_XIAN80) and go to Properties | Joins. Then click on the green PLUS sign to launch the JOIN form



4.2 the JOIN LAYER is the tabular data file which we renamed to z.csv

4.3 the JOIN FIELD is in the JOIN LAYER, in this case the COUN_CODE field (Integer) in z.csv

4.4 the TARGET FIELD is the TARGET LAYER itself, in this case the field that we created JOIN_CODE (Integer) in tw_XIAN80.shp

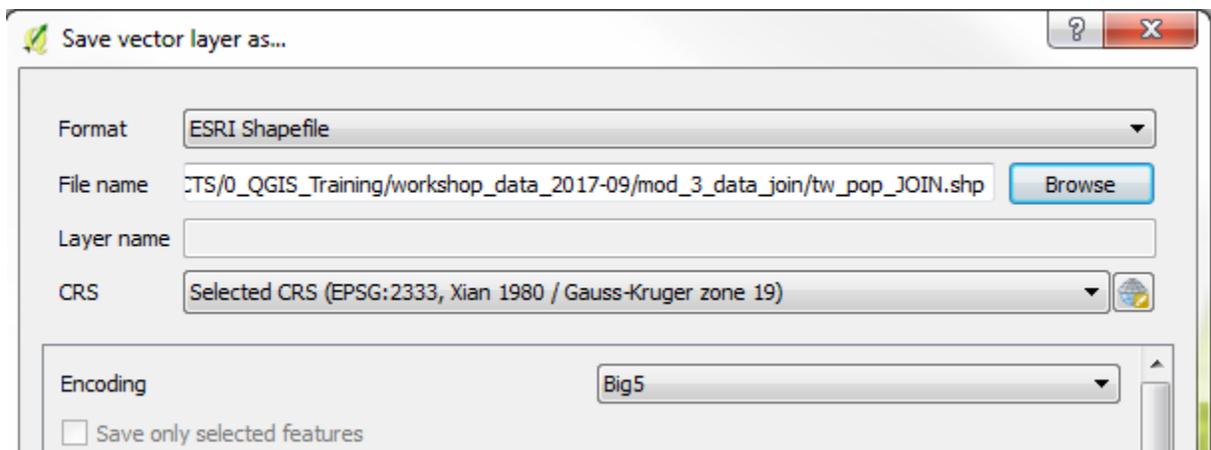
4.5 click OK to run the JOIN which is held in the local Memory Buffer

4.6 right click on the tw_XIAN80 layer to open the Attribute Table and note that the tabular data now APPEARS to be joined to the target layer. It is RELATED in memory, but not actually JOINED yet.

COUN_CODE	JOIN_CODE	z_CNTY_NAME	z_TOWN_NAME	z_FULL_NAME	z_POP_2000	z_POP_2001	z_POP_2002
1000101	1000101	台北縣	板橋市	台北縣板橋市	529059	532694	535476
1000102	1000102	台北縣	三重市	台北縣三重市	382266	384051	384217

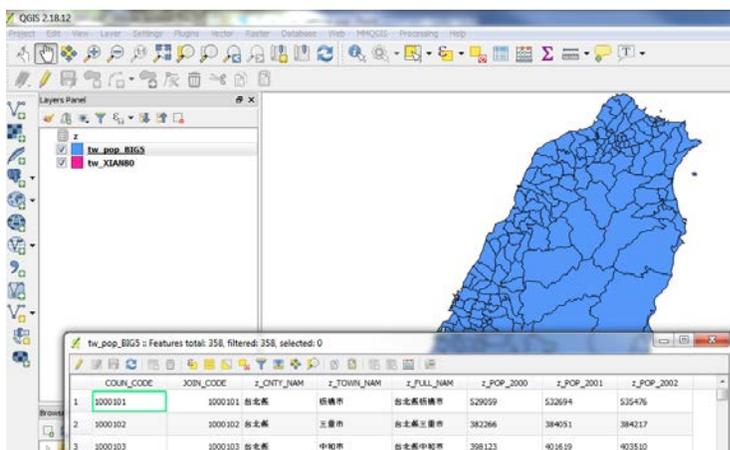
4.7 note that some functions will work with the RELATED tabular data, but geoprocessing will NOT work when the JOIN is held in memory. In order to permanently join the data to the Target file, you will want to right-click on the Target Layer and **SAVE AS a new Shapefile**.

4.8 We will save to our WORK folder as tw_pop_JOIN.shp

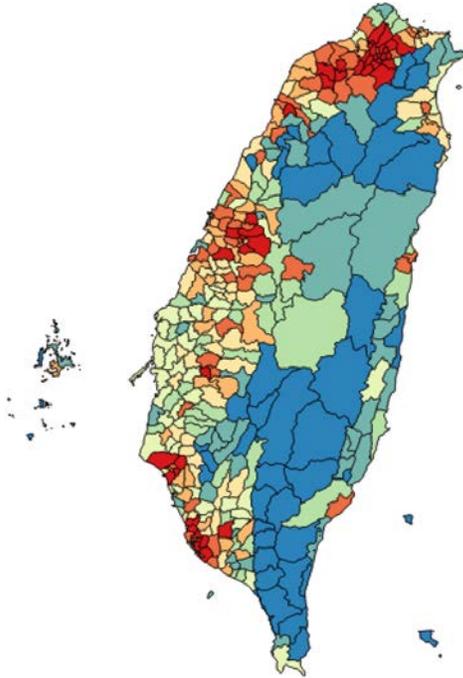


4.9 Note the Format = ESRI Shapefile, the CRS = EPSG:2333, Xian 1980, and the Encoding = Big5.

4.10 With these settings click OK, then the new Layer will be created and opened in the Project layer list



4.11 Now we can right-click on *tw_pop_BIG5* , go to the Properties | Style settings and create Thematic Maps of Population values from the data that we have successfully joined to the polygon layer.



TIP: Note that the field names in any JOIN operation include the first 8 characters of the join filename as a prefix, like: TW_pop_B_5. To help identify the fields of the JOIN layer, first save it to a new file with only one character filename, such as z.csv. Then the JOIN field will look like: z_POP_2002, instead of:

COUN_CODE	JOIN_CODE	TW_pop_BIG	TW_pop_B_1	TW_pop_B_2	TW_pop_B_3	TW_pop_B_4	TW_pop_B_5
1000101	1000101	台北縣	板橋市	台北縣板橋市	529059	532694	535476

Data Sources

Taiwan Ministry of Education

https://worldmap.harvard.edu/data/geonode:taiwan_pop_big5_wgs84_AOM