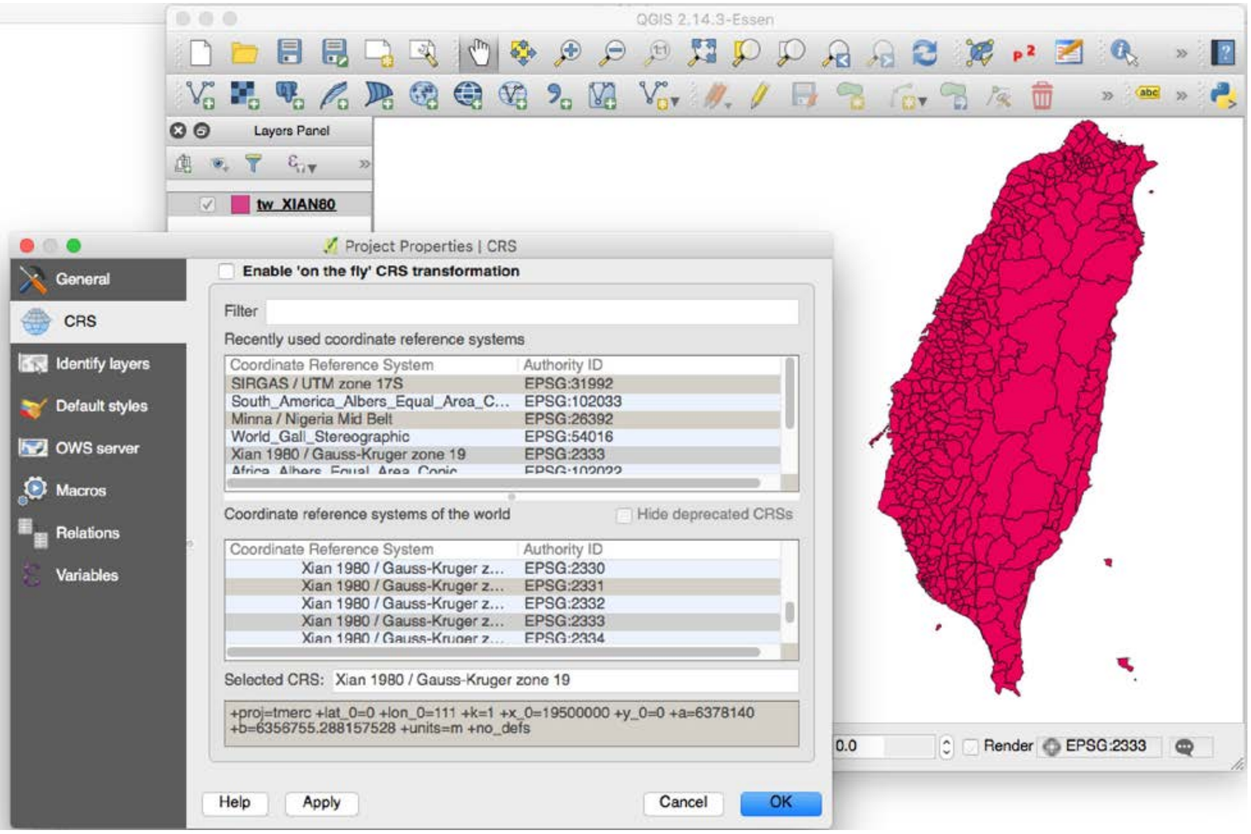
Module 3: Data Join

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 tw_XIAN80 (polygon file for Taiwan districts) Note Xi'an 1980 Projection In **meters**



☐ Enable 'on the fly' CRS transformation

Filter

Recently used coordinate reference systems

Coordinate Reference System	Authority ID
SIRGAS / UTM zone 17S	EPSG:31992
South_America_Albers_Equal_Area_C...	EPSG:102033
Minna / Nigeria Mid Belt	EPSG:26392
World_Gall_Stereographic	EPSG:54016
Xian 1980 / Gauss-Kruger zone 19	EPSG:2333
Africa_Albers_Equal_Area_Conic	EPSG:102022

Coordinate reference systems of the world ☐ Hide deprecated CRSs

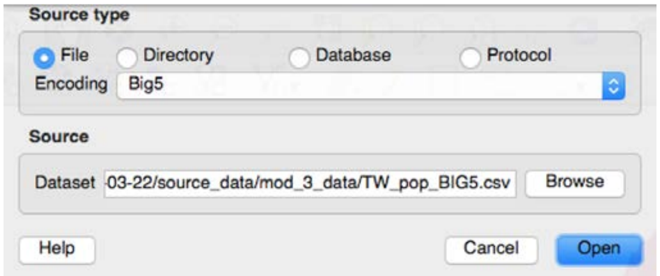
Coordinate Reference System	Authority ID
Xian 1980 / Gauss-Kruger z...	EPSG:2330
Xian 1980 / Gauss-Kruger z...	EPSG:2331
Xian 1980 / Gauss-Kruger z...	EPSG:2332
Xian 1980 / Gauss-Kruger z...	EPSG:2333
Xian 1980 / Gauss-Kruger z...	EPSG:2334

Selected CRS: Xian 1980 / Gauss-Kruger zone 19

```
+proj=tmerc +lat_0=0 +lon_0=111 +k=1 +x_0=19500000 +y_0=0 +a=6378140
+b=6356755.288157528 +units=m +no_defs
```

2. Check .csv file contents

2.1 open the tabular data file using the same ADD VECTOR DATA method (select BIG5 encoding)



Source type

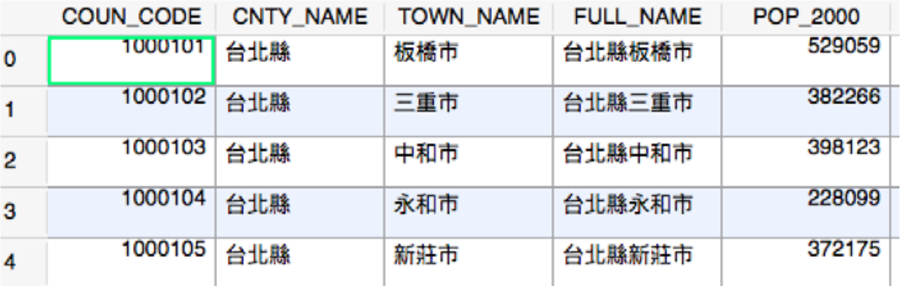
● File ○ Directory ○ Database ○ Protocol

Encoding Big5

Source

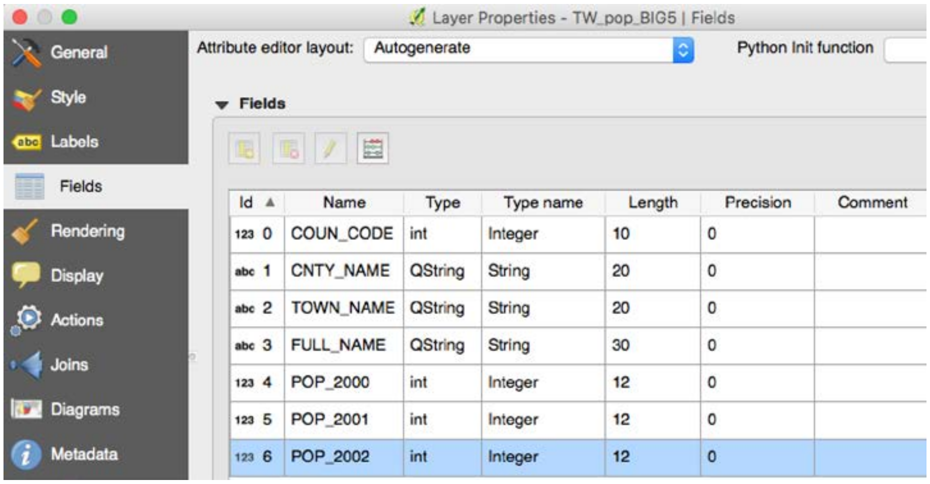
Dataset 03-22/source_data/mod_3_data/TW_pop_BIG5.csv

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

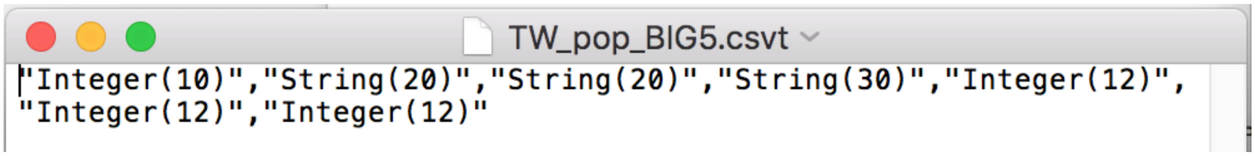
2.3 *right-click* the tabular data layer to inspect Properties | Fields. You want to make note of the field that will serve as the Join Key, in this case COUN_CODE, which is Type Int (10)



Attribute editor layout: Autogenerate

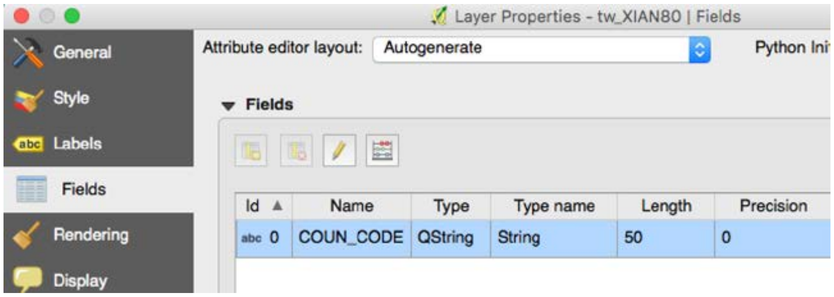
Id	Name	Type	Type name	Length	Precision	Comment
123 0	COUN_CODE	int	Integer	10	0	
abc 1	CNTY_NAME	QString	String	20	0	
abc 2	TOWN_NAME	QString	String	20	0	
abc 3	FULL_NAME	QString	String	30	0	
123 4	POP_2000	int	Integer	12	0	
123 5	POP_2001	int	Integer	12	0	
123 6	POP_2002	int	Integer	12	0	

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.



```
"Integer(10)","String(20)","String(20)","String(30)","Integer(12)",
"Integer(12)","Integer(12)"
```

2.5 *right click* on tw_XIAN80 layer and inspect the Poperties | Fields

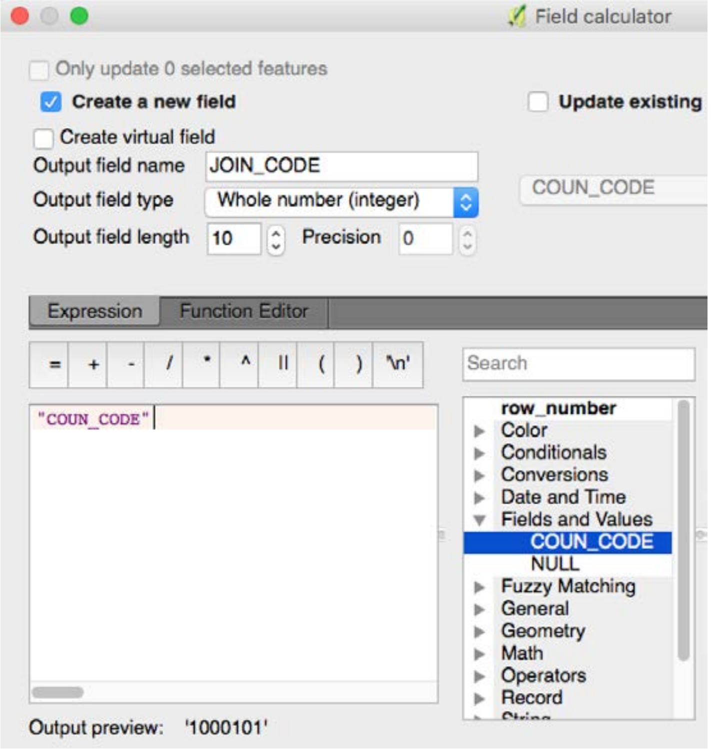


Attribute editor layout: Autogenerate

Id	Name	Type	Type name	Length	Precision
abc 0	COUN_CODE	QString	String	50	0

Now we have a problem. The format of the COUN_CODE in the polygon layer is 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



☐ Only update 0 selected features

☑ Create a new field

☐ Update existing

☐ Create virtual field

Output field name: JOIN_CODE

Output field type: Whole number (integer)

Output field length: 10

Precision: 0

```
"COUN_CODE"
```

Output preview: '1000101'

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)

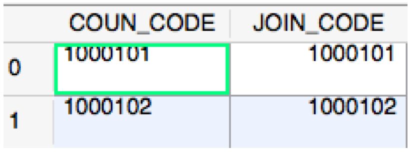
3.3 the Output Field Length can be set to 10

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.

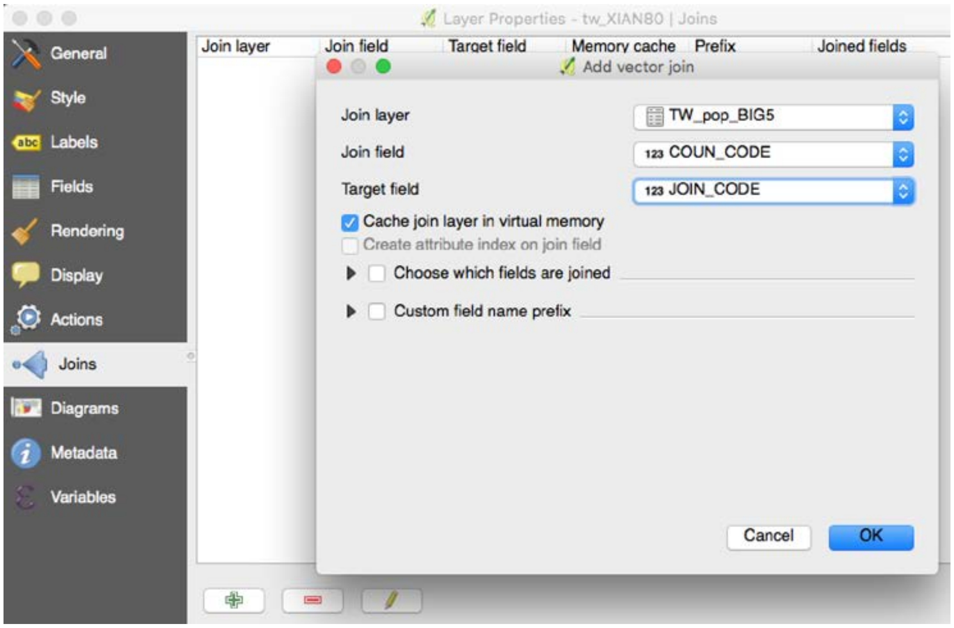


	COUN_CODE	JOIN_CODE
0	1000101	1000101
1	1000102	1000102

3.8 click the TOGGLE EDIT button and hit SAVE to save the updated FIELD

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



Join layer: TW_pop_BIG5

Join field: 123 COUN_CODE

Target field: 123 JOIN_CODE

☑ Cache join layer in virtual memory

☐ Create attribute index on join field

☐ Choose which fields are joined

☐ Custom field name prefix

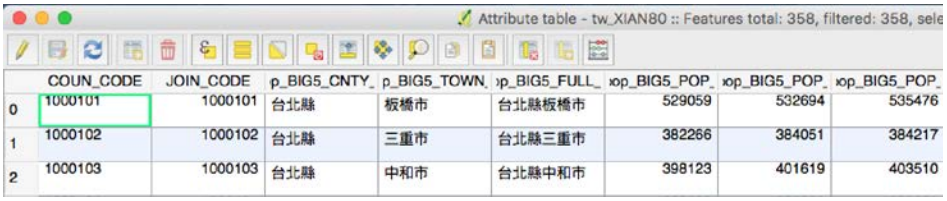
4.2 the JOIN LAYER is the tabular data file, tw_pop_BIG5

4.3 the JOIN FIELD is the join field in the JOIN LAYER, in this case the COUN_CODE field (Integer)

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

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 see the Attribute Table and that the tabular data now APPEARS to be joined to the target layer. It is RELATED in memory, but not actually JOINED yet.

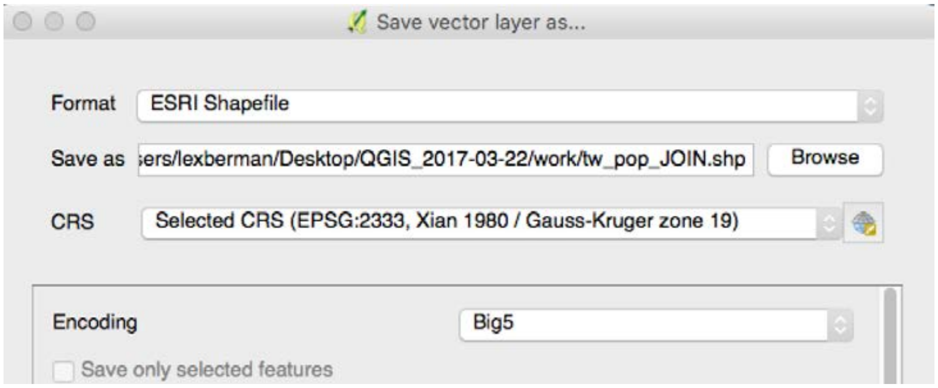


Attribute table - tw_XIAN80 :: Features total: 358, filtered: 358, sele

	COUN_CODE	JOIN_CODE	p_BIG5_CNTY_	p_BIG5_TOWN_	p_BIG5_FULL_	op_BIG5_POP_	op_BIG5_POP_	op_BIG5_POP_
0	1000101	1000101	台北縣	板橋市	台北縣板橋市	529059	532694	535476
1	1000102	1000102	台北縣	三重市	台北縣三重市	382266	384051	384217
2	1000103	1000103	台北縣	中和市	台北縣中和市	398123	401619	403510

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



Save vector layer as...

Format: ESRI Shapefile

Save as: ›ers/lexberman/Desktop/QGIS_2017-03-22/work/tw_pop_JOIN.shp

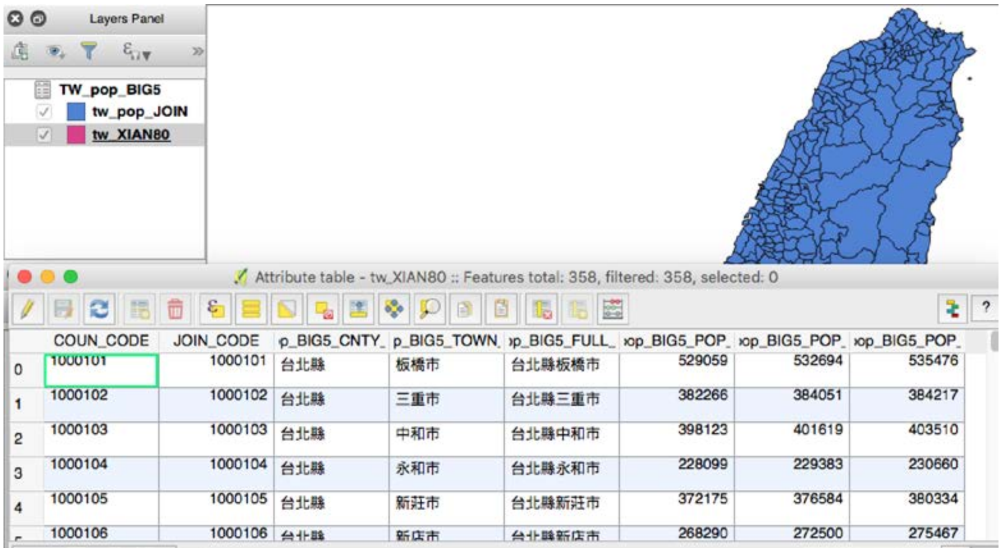
CRS: Selected CRS (EPSG:2333, Xian 1980 / Gauss-Kruger zone 19)

Encoding: Big5

☐ Save only selected features

4.9 Note the Format is ESRI Shapefile, the CRS is Xian 1980, and the Encoding is Big5.

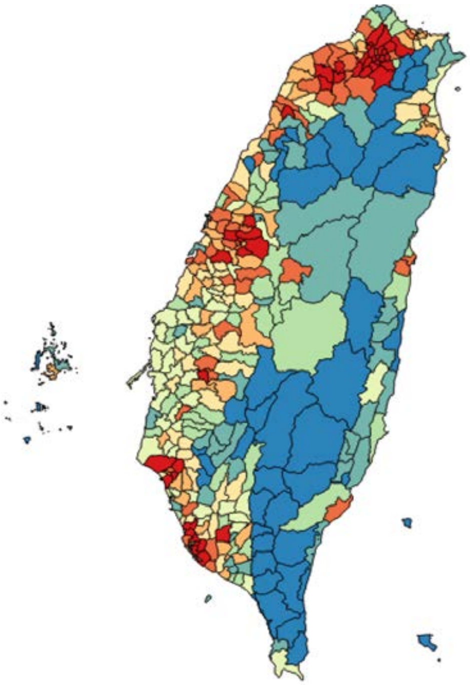
4.10 With these settings click OK, the new Layer will be created and open in the Project



Attribute table - tw_XIAN80 :: Features total: 358, filtered: 358, selected: 0

	COUN_CODE	JOIN_CODE	p_BIG5_CNTY_	p_BIG5_TOWN_	p_BIG5_FULL_	op_BIG5_POP_	op_BIG5_POP_	op_BIG5_POP_
0	1000101	1000101	台北縣	板橋市	台北縣板橋市	529059	532694	535476
1	1000102	1000102	台北縣	三重市	台北縣三重市	382266	384051	384217
2	1000103	1000103	台北縣	中和市	台北縣中和市	398123	401619	403510
3	1000104	1000104	台北縣	永和市	台北縣永和市	228099	229383	230660
4	1000105	1000105	台北縣	新莊市	台北縣新莊市	372175	376584	380334
5	1000106	1000106	台北縣	新店市	台北縣新店市	268290	272500	275467

4.11 Now we can go to the Properties | Style settings to create Thematic Maps of Population values from the data that we have successfully joined to the polygon layer.



Raw counts Population per district

Data Sources

Taiwan Ministry of Education

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