

Introduction to GIS with QGIS

1 Introduction

This practical provides an introduction to GIS using the open source QGIS software. We will focus on vector data, in particular point and polygon data.

2 Datasets used

Note that for many of the datasets used, the data has been clipped to keep files sizes manageable. To access the full datasets, follow the links referred to in the *Source* column.

filename	Description	Source
accidents.shp	Road accidents in central Manchester	data.gov.uk
SJ89NW.tif	OS Open Map - Local	Ordnance Survey ¹
SJ89.tif	OS VectorMap District	
OSMlinks.shp	Road links in central Manchester	OpenStreetMap
OSMnodes.shp	Road nodes in central Manchester	
LA_boundaries.shp	UK Local authority boundaries (subset)	ONS ²
LSOA.shp	Census Lower Super Output Areas (subset)	ONS ³
WY_crime_JF_2019.csv	Crimes in West Yorkshire January and February 2019	data.police.uk
leeds_city.shp	Boundary of the city of Leeds	ONS ⁴
leeds_generalized.shp	Generalised boundary of Leeds district	ONS ⁵
iod_2019.csv	English indices of deprivation at LSOA level (subset)	gov.uk ⁶
names.csv	Open Names (subset)	Ordnance Survey ⁷

¹ <https://www.ordnancesurvey.co.uk/opendatadownload/products.html>

² <https://geoportal.statistics.gov.uk/datasets/local-authority-districts-december-2016-full-clipped-boundaries-in-the-uk>

³ <http://geoportal.statistics.gov.uk/datasets/lower-layer-super-output-areas-december-2011-full-clipped-boundaries-in-england-and-wales>

⁴ <http://geoportal.statistics.gov.uk/datasets/major-towns-and-cities-december-2015-boundaries>

⁵ <https://geoportal.statistics.gov.uk/datasets/local-authority-districts-april-2019-boundaries-uk-buc>

⁶ <https://www.gov.uk/government/statistics/english-indices-of-deprivation-2019> (File 7)

⁷ <https://www.ordnancesurvey.co.uk/opendatadownload/products.html>

3 Introduction to point pattern analysis

1. Open QGIS and create a new blank project (*Project > New*)
2. Save the project in your working folder as *accidents* (*Projects > Save As...*)
3. Open the *Shapefile* format dataset *accidents.shp* (*Layer > Add Layer > Add Vector Layer > Source*)
4. Look at the locations of the road accidents, can you spot any patterns?
5. Add the vector layer, *OSMlinks.shp* (road network)
6. Does this layer provide any useful context?
7. Open the attribute table for the accidents layer (Right-click on layer, *Open Attribute Table*)
8. Note the different attributes for the accidents
9. How many accidents are there?
10. What years does the layer represent?

3.1 Backdrop mapping

Although this practical focusses on the use of vector data, we can use raster datasets as backdrop mapping. Here we are going to use mapping provided as open data by the Ordnance Survey.

1. Add the raster layer *SJ89NW.tif* (*Layer > Add Layer > Add Raster Layer*)
2. Note that there is no attribute table for this layer, as it is a raster
3. Zoom in and out and determine the scale range (maximum and minimum scale) at which this layer is appropriate
4. Add the raster layer *SJ89.tif*
5. Zoom in and out and determine the scale range (maximum and minimum scale) at which this layer is appropriate. How do the two layers vary?

We can also use backdrop map tiles provided by web services.

6. From the *Browser* window, drag the *OpenStreetMap* layer (under XYZ tiles) into the *Map* view
7. Does this layer provide any useful context?

We can add other base layers provided by web services.

8. Right-click *XYZ Tiles* in the *Browser* window and select *New Connection*
9. Under *Name* enter *Google Roads* or something similar
10. In *URL* enter:

`https://mt1.google.com/vt/lyrs=m&x={x}&y={y}&z={z}`

This link connects through to the Google Maps tile server⁸.

11. Click *OK*
12. You should now have a new entry under XYZ tiles. Drag it into the map window.
13. What scale ranges do the OSM and Google Maps layers work best at?

3.2 Symbolising data

1. Hide all layers apart from the *accidents*
2. Select the *Symbology* properties for the *accidents* layer (right-click, *Properties* > *Symbology*)
3. Change from *Single Symbol* to *Categorized*
4. For the column select *SEVERITY*, leave the *Color Ramp* as *Random Colors* and click *Classify* then *Apply*
5. With the Symbology window still open, click on each of the three symbols and experiment with *Color* and *Size*
6. Select *Labels* from the *Properties* window
7. Change from *No labels* to *Single labels*
8. *Label with YEAR*

3.3 Identifying patterns

A quick visual analysis of the accident data will probably conclude that the accidents are clustered, mostly around road junctions. We can quantify the level of clustering.

3.3.1 Nearest neighbour analysis

1. Remove the labels for the accidents
2. Open the *Toolbox* (*Processing* > *Toolbox*)
3. In the *Search* box, type *neighbour*
4. Double-click on *nearest neighbour analysis*
5. Ensure that *accidents* is the *Input Layer* then click *Run*
6. A web link will appear in the *Results Viewer*, click it:

```
Algorithm: Nearest neighbour  
File path: C:  
\Users\nickg\AppData\Local\Temp\processing\_d6b1f6065295456d9b3f2cc12c0ccb31\862a136eac1d4497a6ef78a9997fc78d\OUTPUT\_HTML\_FILE.html
```

A browser will open, containing the results:

Observed mean distance: 16.526014291166195

Expected mean distance: 36.29858682507632

Nearest neighbour index: 0.45527982592835964

⁸ This address and others can be found here: <https://gis.stackexchange.com/a/217670>

Number of points: 266

Z-Score: -16.995938899861503

This tool calculates the mean nearest neighbour (NN) (*observed*) distance for the dataset and compares it to the *expected* mean NN distance if the points were randomly distributed. In this case the observed is less than the expected, yielding an index of less than 1, meaning that the *accidents are clustered*. If the data was randomly located then the index would equal 1. The lower the index, the higher the degree of clustering. The z-score represents a standard deviation⁹.

We can explore the nearest neighbour analysis further by generating a random set of data to emulate the accident data.

7. From the *Vector* menu select *Research Tools > Random Points in Extent*
8. For *Input Extent* select *Use Layer Extent* and choose the *accidents* layers
9. For *Number of points* select the same number as there are in the *accidents* layer
10. Click *Run*
11. We now have a point layer with the same number and extent as the *accidents* layer. Repeat the nearest neighbour analysis using the new layer. **What is the value for the index? Is this as expected?**

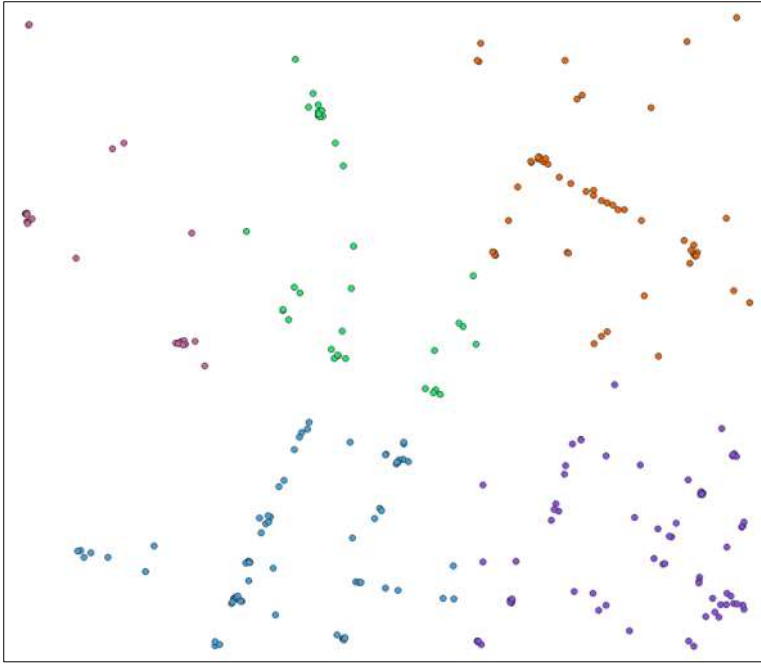
The ability to generate random datasets is a useful feature of QGIS; it helps us to validate our results.

3.3.2 Identifying Clusters

We can also identify the actual clusters in a dataset, using the K-means algorithm, which identifies n clusters in a dataset.

1. In the *Toolbox* search for *k-means*
2. Double-click on the *K-means clustering* tool (Under *Vector analysis*)
3. Ensure that *accidents* is the *Input Layer*, select 5 as the *Number of clusters* then click *Run*
4. Note the new layer named *Clusters*
5. Open the attribute table for *Clusters* and note the new column, *CLUSTER_ID*
6. Open the *Symbology* tab for the new layer and *Categorize* the data using the *CLUSTER_ID* column. The five clusters should be identifiable:

⁹ For more on z-scores see <http://desktop.arcgis.com/en/arcmap/10.3/tools/spatial-statistics-toolbox/what-is-a-z-score-what-is-a-p-value.htm>



We can also extract the mean centres for the clusters.

1. In the *Toolbox* search for *mean*
2. Double-click on the *Mean coordinate(s)* tool
3. Select *Clusters* as the *Input Layer*
4. In the Unique ID field select *CLUSTER_ID* (this will determine the mean centre for each cluster, instead of the whole dataset)
5. Click *Run*
6. A new, temporary layer (of five points) will be created. Use *Symbology* to ensure the mean centres are distinguishable from the accidents

We can also extract the geometries that define the 5 clusters.

1. In the *Toolbox* search for *minimum* and select *Minimum bounding geometry*
2. Choose *Clusters* as the *Input layer*
3. Choose *CLUSTER_ID* as the field (this is used to identify the separate clusters)
4. From *Geometry Type* choose *Convex Hull* then *Run*
5. Repeat step 4, this time generating *Envelope (Bounding Box)*

The *Clusters*, *Mean coordinates* and bounding geometry layers are *temporary*¹⁰; when QGIS is closed they will disappear. We can make them permanent.

1. Right-click on one of the temporary layers and select *Export > Save Features as*
2. From the *Format* choose *ESRI Shapefile*
3. Navigate to your working folder and specify a file name
4. Make sure the *CRS* (Coordinate Reference System) is *EPGS:27700...* and click OK

¹⁰ *Properties > Information > Storage* is displayed as *Memory storage*

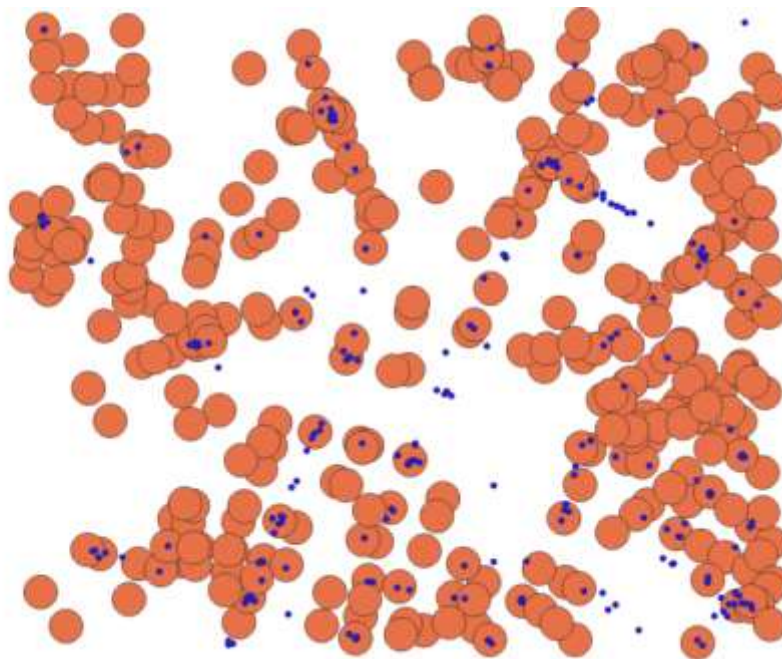
5. By default the new layer is opened. Remove the temporary layer (Right-click, *Remove Layer...*)

To be honest, the K-means algorithm is not that useful for this dataset, as road accidents tend to cluster around dangerous hotspots (usually junctions in urban areas), rather than geographical regions. In this next exercise we are going to attempt to identify dangerous junctions.

3.4 Point in polygon

The process for identifying dangerous junctions is:

- Add a buffer around each road junction
 - Count the number of accidents in each buffer
1. Open the vector layer *OSMnodes.shp*; these represent road junctions
 2. Open the *Buffer* tool (*Vector > Geoprocessing Tools > Buffer...*)
 3. Select *OSMnodes* as the *Input Layer*
 4. Select 30m as the *distance* (this is arbitrary) and *Run*:



5. Open the attribute table for the buffer layer (*Buffered*) and note that it has preserved the attributes of the source layer, including the unique identifier, *osmid*.

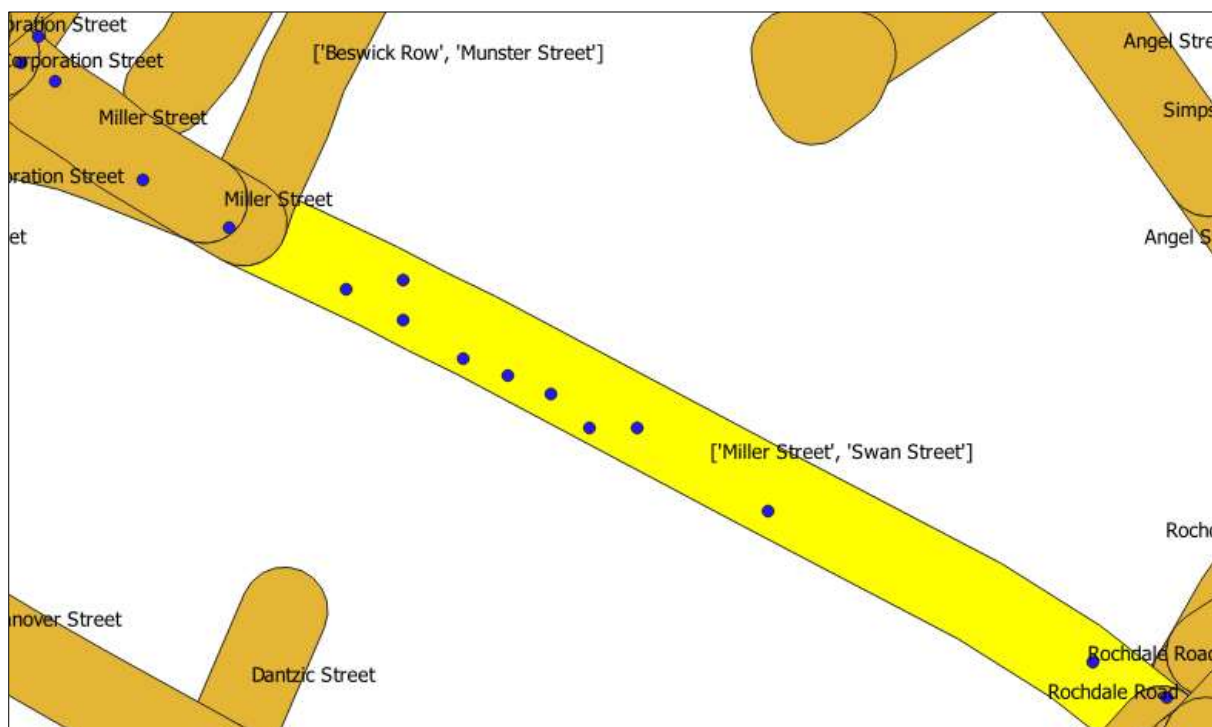
The next step is to count the accidents in each buffer.

6. Select *Vector > Analysis Tools > Count Points in Polygon*
7. Select the buffer layer as the *Polygons*
8. Select *accidents* as the *Points*
9. Under *Count field name* enter *accidents* and *Run*

10. This will create a new temporary layer named *Count*. Open its attribute table
11. Determine the junction with the most accidents (click on the column heading for the *accidents* attribute to sort it)
12. To view the junction in detail, select the row by click the number (1) in the first column of the attribute table and clicking the *Zoom map to selected rows* icon in the attribute table (it's a yellow magnifying glass).
13. Can you identify a key flaw with this method?

Another technique of identifying dangerous locations is to identify the section of road (or link) that has the most number of accidents.

14. Repeat the above process, but this time generate the buffers around the road links (*OSMlinks.shp*)
15. What is the name of the most dangerous link? In the example below, a 10m buffer was used:



4 Point and polygon analysis

In this section we are going to explore both point data (crimes) and polygon data (deprivation). The deprivation data is collected at a point (household) level but to anonymise the data it is aggregated to polygons.

Firstly, we are going to examine crime in the district of Leeds. But we first need to identify its boundary.

1. Save the current project, close it and create a new one named *crime*
2. Open the vector layer, *LA_boundaries.shp*

3. Open the attribute table for the layer, and sort it on the *lad16nm* field
4. Select the *Leeds* entry by clicking the feature ID
5. Check that the polygon is highlighted in the map
6. Close the attribute table, right-click the layer and select *Export > Save Selected Features As...*
7. Save the selected feature as a Shapefile, *leeds*, in your working folder
8. Close the original layer

Now we have the boundary of Leeds district we can add the crime data, which is currently in CSV format.

1. Select *Layer > Add Layer > Add Delimited Text Layer...*
2. Navigate to the *WY_crime_JF_2019.csv* file
3. Make sure:
 - a. The file format is *CSV*
 - b. *First record has field names* is checked
 - c. *Points coordinates* is selected and the X and Y fields are *Longitude* and *Latitude*
 - d. The *Geometry CRS* is set to *EPSG:4326*
4. Check the sample data window to see that everything looks right and click *Add* then *Close*

Our polygon layer for Leeds is in OSGB projection (EPSG:27700) and the crime data is in another projection (EPSG:4326). QGIS can cope with this but when we start comparing the two layers this might cause a problem, so...

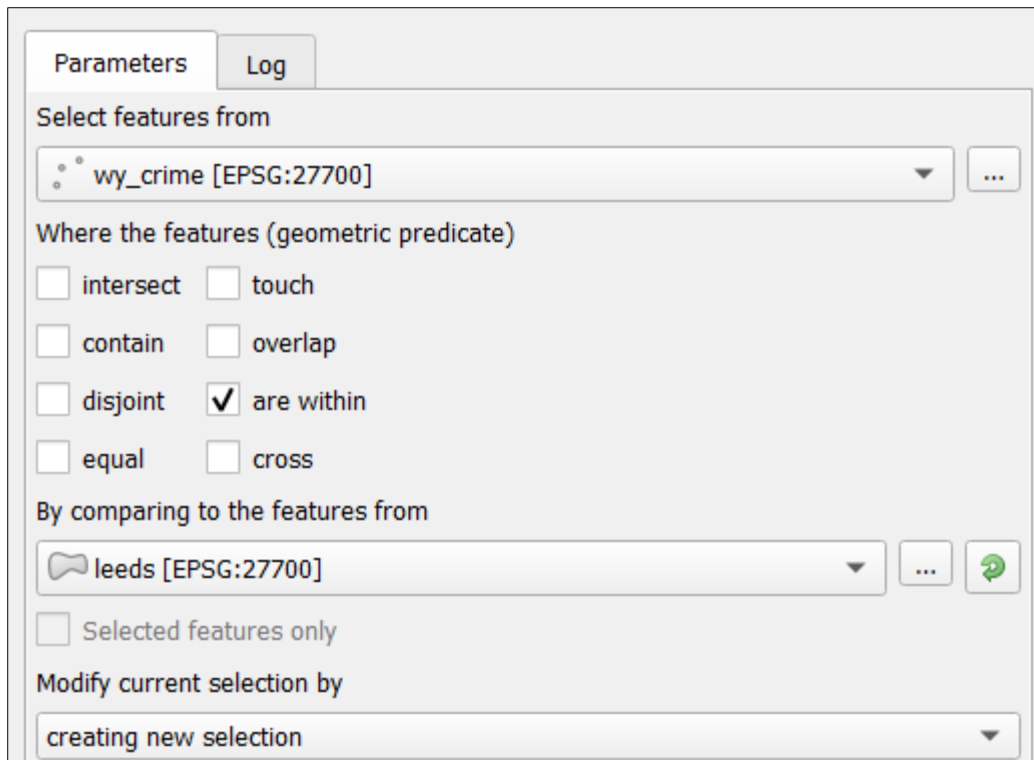
5. Right-click the crime layer then *Export > Save Features As...*
6. Make sure the format is *ESRI Shapefile* and choose a suitable *File Name* e.g. *wy_crime*
7. For the CRS choose *EPSG:27700*
8. Click *OK*
9. Remove the original crime layer (CSV file)

Both layers will now have the same CRS, but as you can see, the crimes dataset includes crimes outside the Leeds district boundary (it covers the West Yorkshire Police authority extent¹¹).

4.1 Select by location

1. From the *Vector* menu select *Research Tools > Select by Location...*
2. *Select features from* - the crime point layer
3. *Where the features – are within Leeds:*

¹¹ <https://www.justiceinspectors.gov.uk/hmicfrs/media/police-force-map.pdf>



4. Note the selected features
5. Right-click on the crime layer and select *Export > Save Selected Features As...*
6. Choose a suitable file name e.g. *leeds_crime*, and save the selected features as a *Shapefile*
7. Using the *Symbology* tab, categorise the crimes by *crime type*

It would be useful to determine the number of crimes of each type. We can do that by creating a new *virtual* layer.

8. Select *Layer > Create Layer > New Virtual Layer...*
9. In the Query box type:

```
SELECT "Crime type", COUNT(*) FROM leeds_crime GROUP BY "Crime type"
```

This is essentially a non-spatial SQL query.

10. Click *Add* then *Close*
11. The new virtual layer will appear in the *Layers* window; open its attribute table
12. What is the most common crime type?

4.2 Select by attribute

We are now going to focus on one crime type, *Violence and sexual offences*. Earlier we selected crimes by their location; we can also do a selection by attribute values.

1. Open the attribute table for the crime layer and click the icon *Select features using an expression*

2. From *Fields and Values* double-click *Crime type* to add it to the *Expression* window
3. Click or type “=” then click *All Unique* to list the possible values for this attribute
4. Double-click on *Violence and sexual offences*.

The expression should read:

```
"Crime type" = 'Violence and sexual offences'
```

This is effectively the WHERE clause of an SQL query. Here we have a very simple query but we can create more complex queries using more than one attribute and multiple logical operators such as AND and OR.

5. Click the *Select Features* button.

The features meeting this criteria will be selected in the attribute table and map window.

6. Right-click the crimes layer and select *Export > Save Selected Features As...* and save the features as a Shapefile e.g. *leeds_violence*.

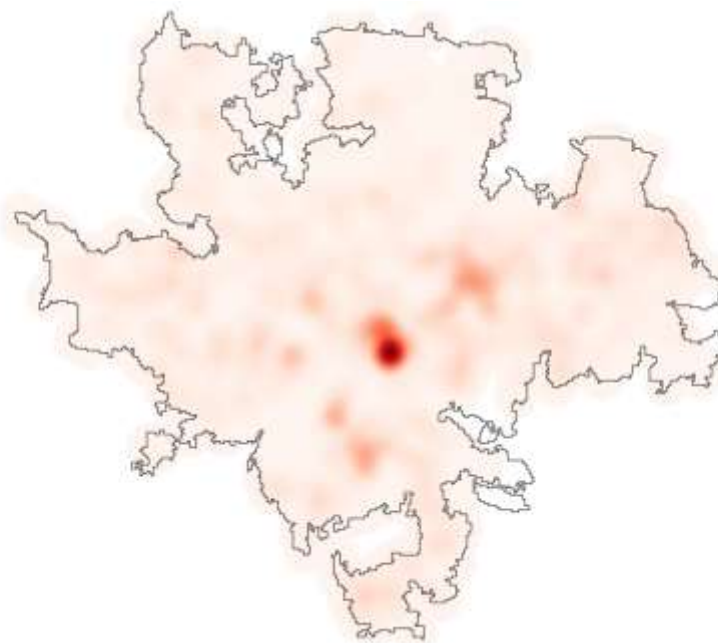
We wish to restrict the analysis to the *city* of Leeds rather than the full boundary of the local authority.

7. Open the *leeds_city* Shapefile, add a base map, e.g. OpenStreetMap, and check that the layer covers the extent of the city
8. Perform a *select by location* to extract only those violent crimes that take place in the city. Export the results as a new Shapefile – *leeds_city_violence*
9. Perform a nearest neighbour analysis of the violent crime data (see section 3.3.1); is it clustered?

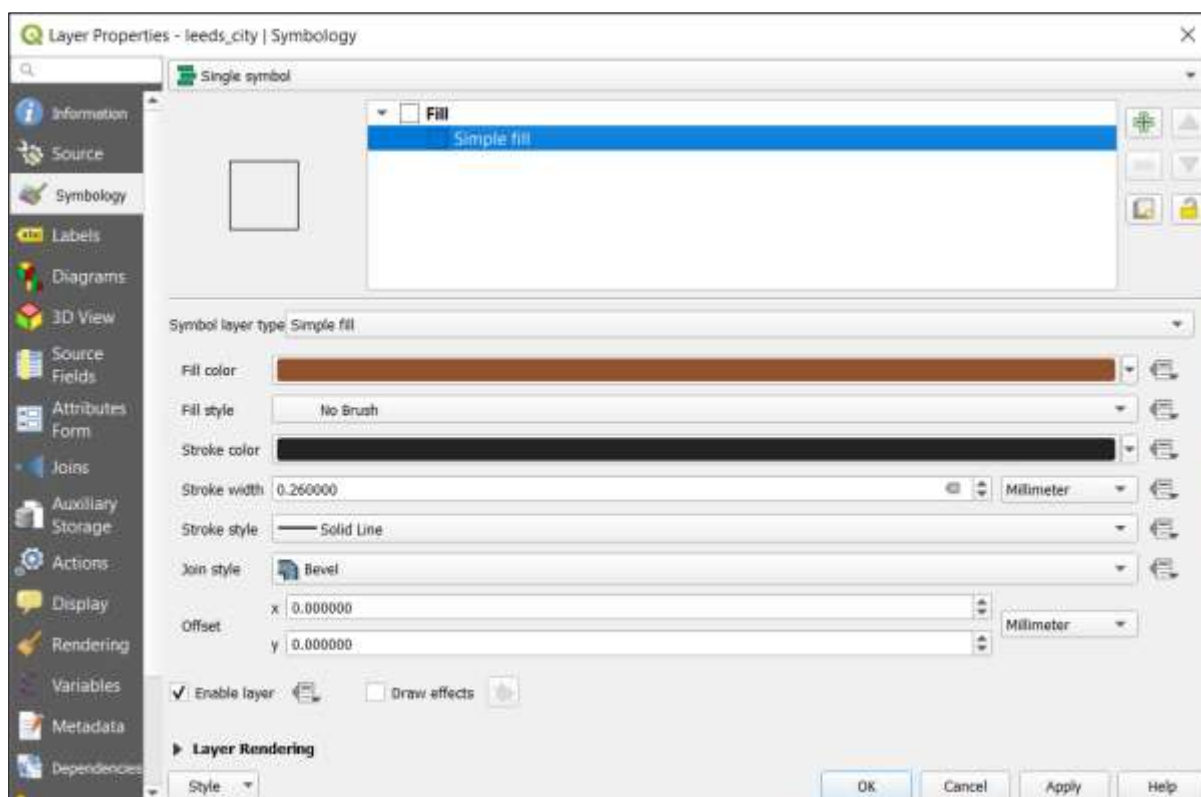
The next step is to produce a hotspot analysis of violent crimes in the city.

10. Open the Processing toolbox and search for *heatmap*
11. Open the *Heatmap* tool
12. Ensure that the *Point layer* is your layer of violent crime
13. You will need to experiment with radius but try *500m* for a start
14. The default cell size for the output raster is too small; change this by setting the *Pixel size X* to 50 (metres) – this will automatically change the other output raster values
15. Click *Run*

The output will be a raster image:



In the example above, a radius of 500m and an output cell size of 50m were used. The Symbology of the raster was changed to *Singleband pseudoColor* and a *Reds Color ramp* was applied. The city boundary was given a transparent fill by setting the *Fill style* to *No Brush*:

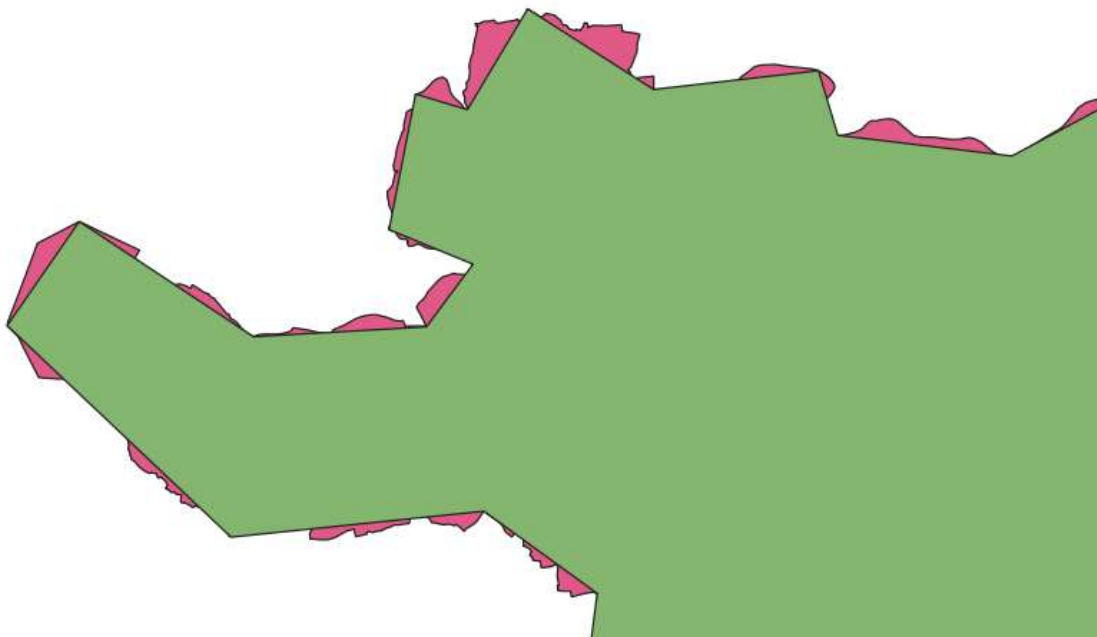


16. Using *OpenStreetMap* and the heatmap, determine which part of the city suffers most from violent crime.

4.3 Generalisation

Cartographic generalisation involves the reduction in information as a map is changed from one scale to another.

1. Hide all layers apart from the Leeds district boundary (*leeds.shp*), extracted earlier (page 7).
2. Now add the *leeds_generalised* Shapefile:



The image shows a detail of the generalised, simplified, boundary with the “ungeneralised” layer below. Both boundaries are supplied by the ONS on the same website¹², so it is important to select the most relevant dataset. If we used the generalised boundary when using the select (crimes) by location tool, some points might have been omitted, others included. Strictly speaking all boundaries will be generalised to some extent, in this case the layer on top is the more generalised of the two. This raises the question; why not always use the least generalised dataset? Depending on the study being undertaken a high level of detail (LoD), might not be necessary; the generalised dataset is smaller in terms of storage and also quicker for the GIS to process when performing analysis such as point in polygon.

¹² https://geoportal.statistics.gov.uk/search?collection=Dataset&sort=name&tags=BDY_LAD

4.4 Joining layers on a common attribute

In this section we are going to examine both crime and deprivation in the district of Leeds.

1. Save the current project
2. Remove all layers apart from the boundary of Leeds district (*leeds.shp*) and the crimes within this boundary (*leeds_crime.shp*)
3. Hide the crime layer
4. Save the project as *crimes_deprivation*

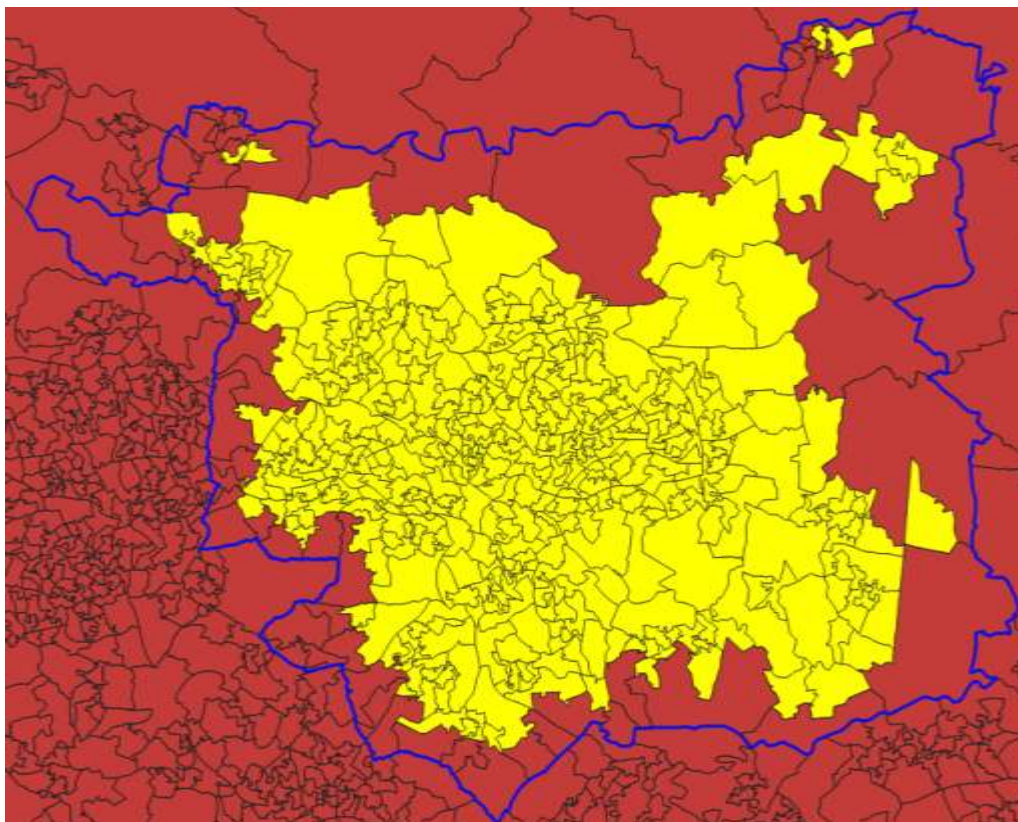
The deprivation data has been anonymised by aggregating it to polygons, in this case Lower Super Output Areas (LSOA), which is a geographic classification used by the UK census.

5. Add the vector layer *LSOA.shp*

As you will see, this layer covers a large area, but we are only interested in those LSOA that lie within the Leeds district.

6. Run a select by location to determine those LSOA that are *within* the Leeds boundary

You will notice that this hasn't quite worked properly:



Some of the LSOAs on the boundary have not been included. They do not lie entirely within the Leeds polygon; this is probably the result of generalisation. To get round this we can add a buffer to the Leeds polygon.

7. Add a 100m buffer (see page 6) to the *Leeds* polygon
8. Repeat the select by location but this time use the buffer
9. Check that all of the expected LSOAs have been selected and save the selected LSOAs to *leeds_Isa.shp*
10. Open the delimited text file *iod_2091.csv* in *Excel*. The columns are:
 - a. LSOA – the unique code for the LSOA
 - b. score – the deprivation score for the LSOA
 - c. rank – the deprivation rank for the LSOA, where 1 is the most deprived LSOA in the country
11. Add the file to QGIS (*Layer > Add Layer > Add Delimited Text Layer...*). There is no geometry information in this dataset so select *No geometry*

The file will be added to the *Layers* window but not mapped (obviously). The next stage is to add this data to the layer of LSOA polygons. This can be done by joining the two layers on a common field, the LSOA code.

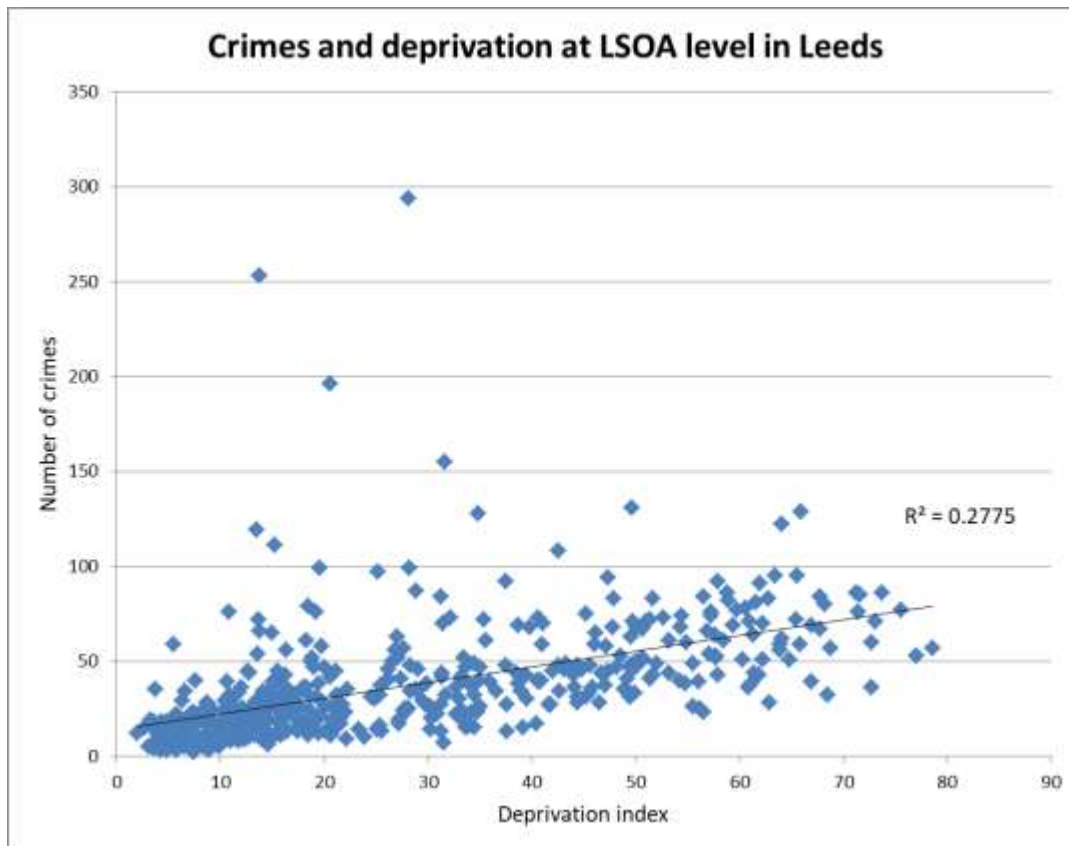
12. Open the *Properties* of the *leeds_LSOA* layer and select *Joins*
13. Click *Add new join*
 - a. Join layer: *iod_2019*
 - b. Join field: *LSOA*
 - c. Target field: *Isa11cd*
 - d. Joined fields: *score* and *rank*
 - e. Custom Field Name Prefix: *iod_*
14. Click *OK* and *OK*
15. Open the attribute table of the LSOA layer and note the two new fields

We now have the deprivation data for each LSOA. The next step is to add the crime data.

16. Use the *Count Points in Polygon* tool (page 6) to add a column displaying the number of *crimes* in each LSOA
17. Export the new temporary layer as *crime_deprivation.shp*
18. Open the attribute table and zoom the map to the LSOA with the most crimes. Use *OpenStreetMap* to identify the location.

We now have a layer that contains both the number of crimes and the deprivation index for each polygon. There is no easy way in QGIS of determining the correlation between these variables but we can export the layer as a CSV for import into another package.

19. Export the *crime_deprivation* layer as a CSV file, choosing to export only the *Isa11cd*, *iod_score* and *crimes* column



Here, the data is plotted as a scatter plot in Excel (with one outlier, in the city centre, removed). The R^2 value of 0.2775 indicates a small positive correlation between the two variables.

4.5 Join attributes by location

Earlier we counted the number of crimes in each LSOA. But what if we want to determine which LSOA each crime resides in? This is a useful technique if we then want to export the data into a database or spreadsheet for further analysis – we have the location information as a simple text-based attribute.

1. From the *Vector* menu select *Data Management Tools > Join Attributes by Location*
2. Select *leeds_crime* as the *Input layer* and *leeds_lsoa* as the *Join layer*
3. Select *within* as the predicate
4. From *Fields to Add* select *lsoa11cd* (the code for the LSOA)
5. For *Join Type* choose a *one-to-one* join
6. Click *Run*

A new temporary layer will be created which will contain the same data as the crime layer but with an extra field for LSOA.

7. To check the results, zoom in, then select the *Identify Features* tool and click on a crime

8. In the *Identify Results* window make sure the *Mode* is *Top Down* and click on the crime again. The *Isa11cd* value of the crime should match that of the underlying LSOA:

Feature	Value
Joined layer	
LSOA name	Leeds 038D
(Derived)	
(Actions)	
Crime ID	1f90d15281128fb9ac97e75c69472f52f789f2350e15dc3f10fbaa...
Month	2019-01
Reported b	West Yorkshire Police
Falls with	West Yorkshire Police
Longitude	-1.558356000000000
Latitude	53.822403000000001
Location	On or near Farm Hill South
LSOA code	E01011450
LSOA name	Leeds 038D
Crime type	Criminal damage and arson
Last outco	Investigation complete; no suspect identified
Context	
Isa11cd	E01011450
LSOA name	Leeds 038D
LSOA name	Leeds 038D
LSOA name	Leeds 038D
LSOA name	Leeds 038D
LSOA name	Leeds 038D
LSOA name	Leeds 038D
LSOA name	Leeds 038D
LSOA name	Leeds 038D
leeds_Isoa	
objectid	11121
(Derived)	
(Actions)	
objectid	11121
Isa11cd	E01011450
Isa11nm	Leeds 038D
Isa11nmw	Leeds 038D
st_areasha	399043.099463500024285
st_lengths	3230.849048420395320

Mode: Top down Auto open form

View: Tree

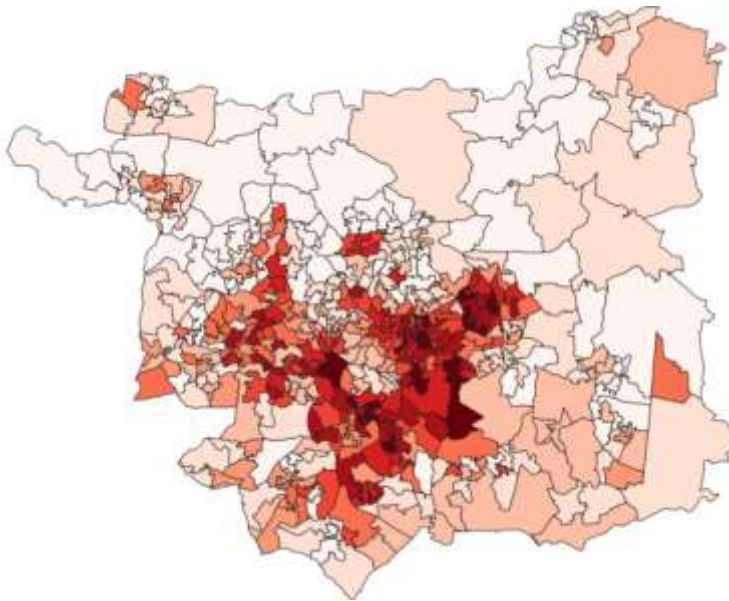
Why do you think that so many crimes share the exact same locations?

4.6 Cartographic outputs

In this section we will generate a map of the deprivation data that we produced earlier.

1. For the *crime_deprivation* layer open the *Symbology* tab and select *Graduated*
 - a. For *Column* choose *iod_score*
 - b. Choose a *Color ramp* e.g. *Reds*
 - c. Choose *9 classes*
 - d. Choose *Mode of Natural Breaks (Jenks)*
 - e. Click *Apply*

The output should be something like:



Interpret the map – do you notice any patterns in the distribution of deprivation around the city of Leeds?

The next step is to add some context to the map by adding some place names.

1. Add the Delimited Text Layer *names.csv*, ensuring *First record has field names* is checked and noting that the X and Y fields are *GEOMETRY_X* and *GEOMETRY_Y*. Set the Geometry CRS to *EPGS:27700*

This dataset is very big and exceeds the extent of our study area. The next step is to select only those name points that are in the Leeds boundary. To do this we do a *select by location*. First try this:

2. Select features from *names* that are *within crime_deprivation*

Note the time that this takes. If you get bored then *Cancel* the operation. Let's speed it up.

3. Firstly, deselect any points that have already been selected
4. From the *Vector* menu select *Geoprocessing Tools > Dissolve*
5. Choose *crime_deprivation* as the input layer

We have "dissolved" the 482 polygons into a single one.

6. This time do a select by location by selecting those *names* that are within the *dissolved* layer

Notice how much quicker this is. Why is this?

7. Save the selected features as *leeds_names.shp*

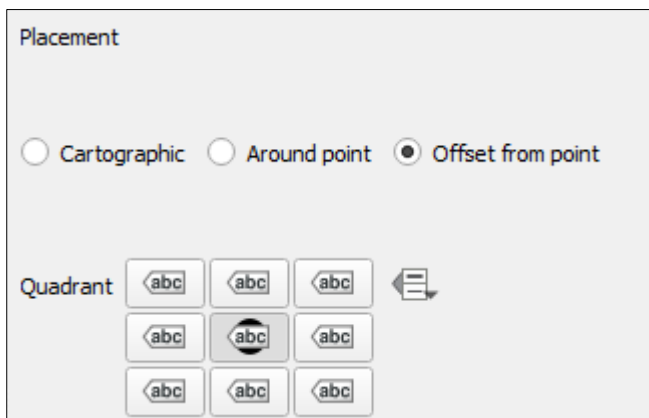
8. Open the attribute table for the *leeds_names* layer and note the variety of *local types*. Since the map is going to focus on the city of Leeds we are going to restrict the layer to just *suburban areas*
9. Perform a select by attribute (section 4.2) to select only those records that have a *local_type* of *suburban area*
10. Export the subset of names as a new Shapefile
11. Add labels for the layer using the *Name1* field (see section 3.2)

The labels should appear as:



However, the locations are not points as such, they represent suburban areas, so we are going to move the label to be over the point and then hide the point.

12. From *Labels* select *Placement* to be *offset from point*
13. Make sure the central Quadrant is selected:



14. Finally, as we don't think of suburban areas as points, we can remove the point symbol – *Symbology* > *No symbols*

We can export the map as an image to be used in e.g. *Word*

15. *Project* > *Import Export* > *Export Map to Image*

However, QGIS has a tool called the *Print Composer* that allows you to add elements – North arrow, scale bar, legend – to a page.

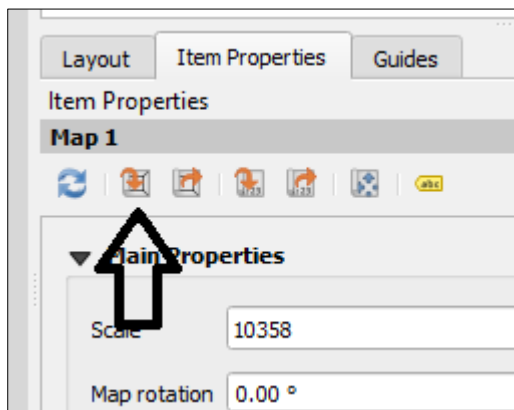
1. From the Project menu select *New Print Layout*
2. You will be prompted to enter a title e.g. *Leeds deprivation*

A new empty layout window will appear. The first stage is to add a map to the page.

3. From the *Add Item* menu select *Add Map*
4. Draw a container for the map

The contents of the map window will mirror the current view in the QGIS map window.

5. Return to the main QGIS window and change the map window e.g. zoom in/out, hide/show layers
6. Return to the Layout window and in the Map 1 properties window, click the button for *Set Map Extent to Map Main Canvas Extent*:



This window will change depending on the currently selected *Item*. We will see later how to fix the map view.

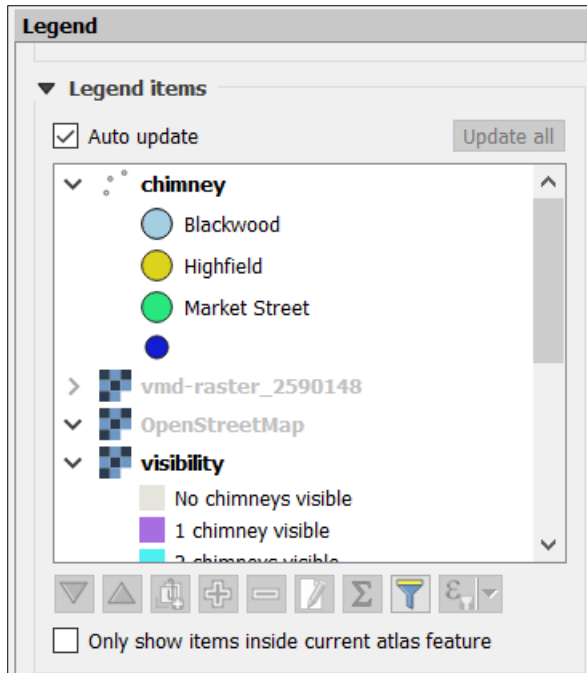
7. Add a *Frame* to the map item – look under *Item Properties*


Adding a legend

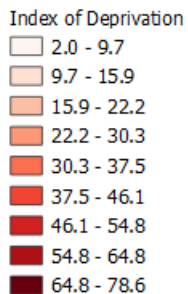
8. From the *Add Item* menu select *Add Legend*
9. Draw the legend on the canvas – note that it will take a while to populate.

The legend's content is determined by the individual layers' symbology and by default will include all of the layers in the *Layers* window in QGIS.

10. In the *Legend* toolbox, uncheck the *Auto update* box:



This will enable the editing tools below the list of legend items. You can select any unwanted layer in the list and click the remove button -  - to remove it. In this example, I have removed all but the *crime_deprivation* layer. I have also used the *edit* button to rename the layer as it appears in the legend:



Adding a title

11. From the *Add Item* menu select *Add Label* and draw the container for the label
12. In the *Item Properties* window enter the text for the title and set the font.

Adding a scale bar

13. From the *Add Item* menu select *Add Scale Bar*
14. Draw and then configure the scale bar:



Here I have chosen a *Double box* style and set the *Right segments* to 5.

Adding an inset

Now we want to add some context to the map; what area are we looking at? We are going to add an inset that will show the study area using OpenStreetMap.

15. Back in the QGIS main window add OpenStreetMap: *Browser Window > XYZ Tiles*
16. Note how the OSM layer now obscures your original map in the *Layout* window
17. Back in the QGIS *Layers* window hide the OSM layer and in the *Layout* window click *Update Map Preview* for the map item.

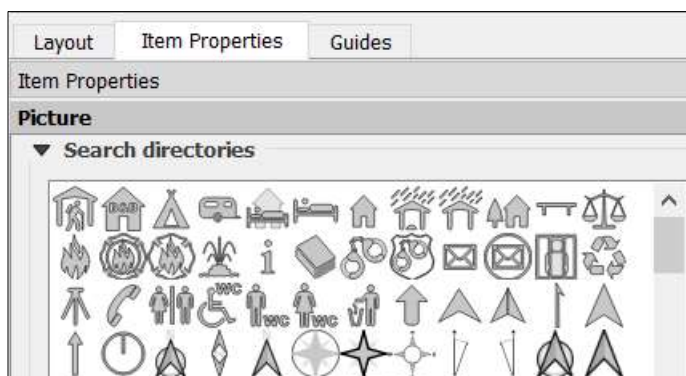
This will return your original map. We need to fix its contents.

18. On the map properties, under *Layers*, check *Lock layers*
19. Return to the QGIS main window and show the OSM layer and zoom to an appropriate level
20. In the *Layout* window add a new map and position it
21. Click *Set to map canvas extent*
22. If you want to pan the inset map (or the main map) select the map item then click the *Interactively Edit Map Extent* button
23. Once satisfied with the inset map, set *Lock layers* for it too.

Adding a North arrow

Finally we should add a North arrow to the map.

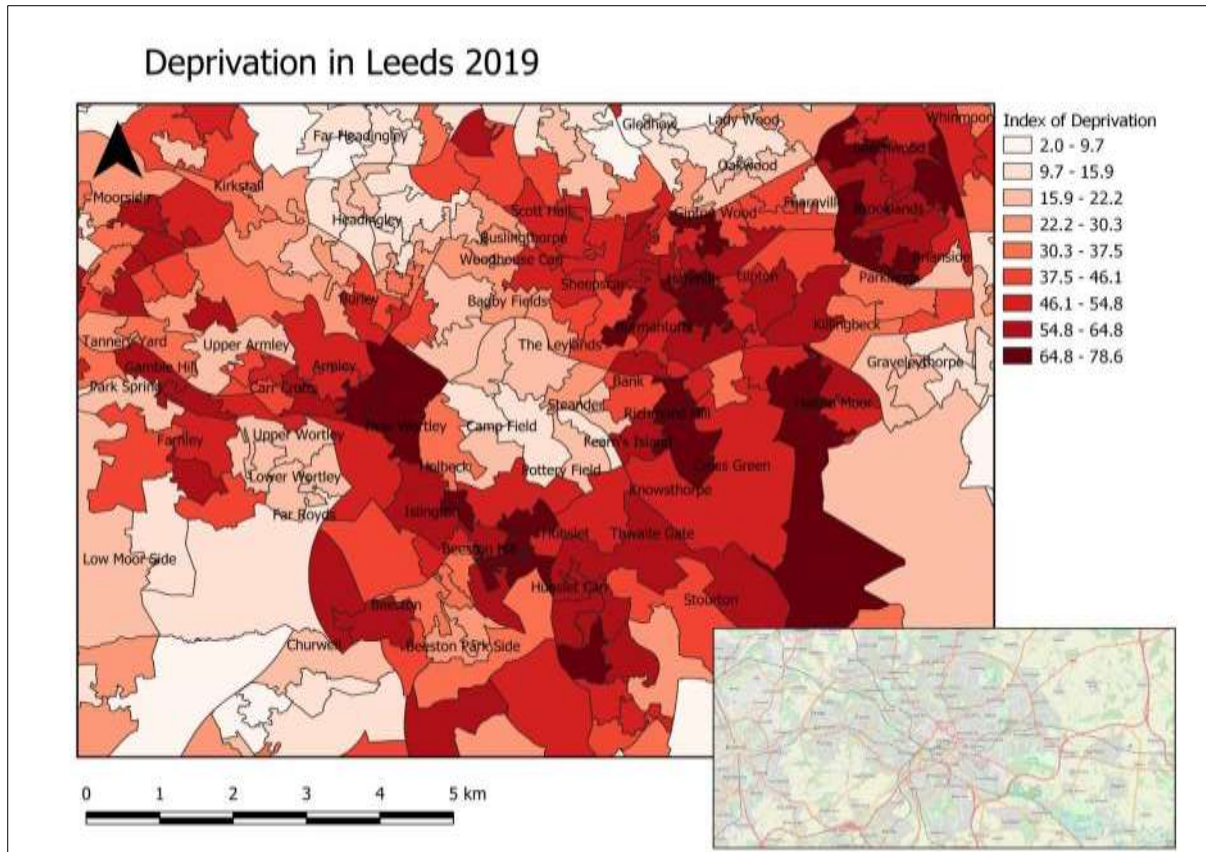
24. From the *Add Item* menu select *Add Picture*
25. In the *Item Properties* expand *Search Directories*:



26. Choose an appropriate symbol

Exporting the layout

27. Once you are happy with the layout's content, *Save* it and from the *Layout* menu select *Export as Image*
28. Save the file as in whatever format you desire:



29. Save the layout and close the *Layout* window

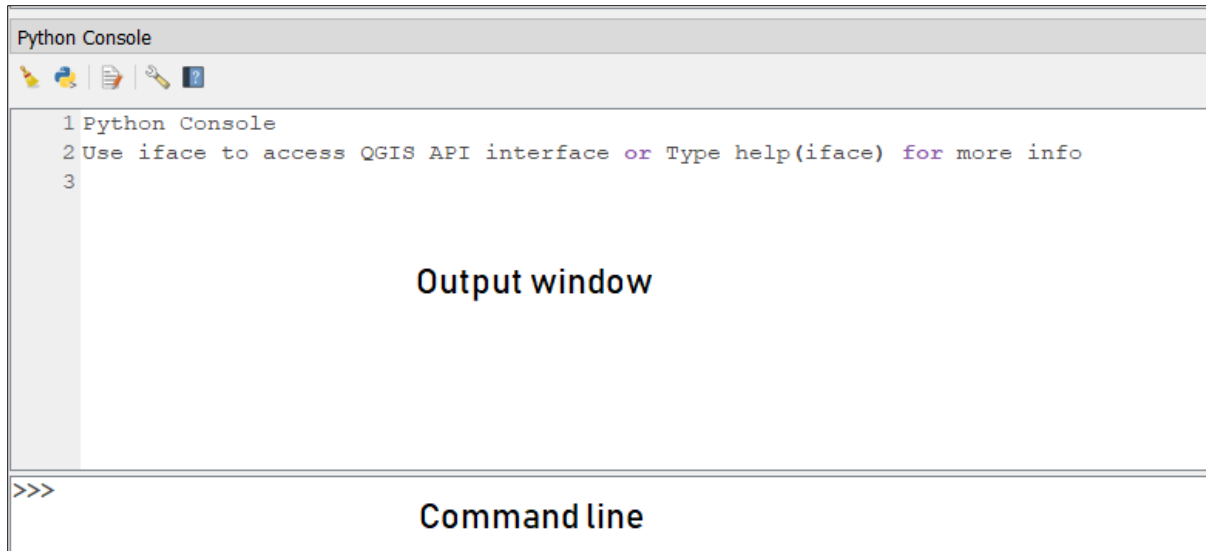
30. The layout can be accessed from the *Layouts* section on the *Project* menu. Note that the layout's parent project has to be open.

5 Python with QGIS

In this section we will look, briefly, at how we can use Python within QGIS to automate tasks. The details of the Python code will not be examined but you should get a reasonable idea of how Python operates within the QGIS environment.

5.1 The Python command line

3. Start a new QGIS project and open the *accidents* and *OSMnodes* shapefiles
4. Select *Python Console* from the *Plugins Menu*. A new window will appear:



5. At the command line, type the following, pressing <Enter> after each line

```
>>> import sys
>>> sys.version
```

6. Note that the console uses predictive typing. What version of Python is QGIS using?

We can explore QGIS layers from the Command line.

7. In the *Layers* window, click the *accidents* layer to make it active
8. At the command line, type:

```
>>> my_layer = qgis.utils.iface.activeLayer()
>>> my_layer.featureCount()
```

9. Does the value returned from the *featureCount* method match that displayed in the attribute table?



For detailed information about any object in the QGIS API, such as *activeLayer*, go to the online documentation <https://qgis.org/pyqgis/3.0/index.html> and search for the object name.

10. Open the attribute table for the *accidents* layer and note that the second column displays the year of the accident
11. Next:

```
>>> my_layer.maximumValue(1)
```

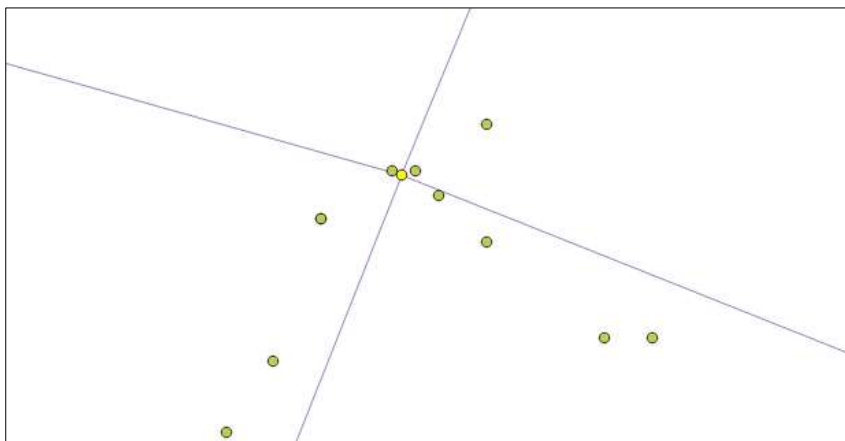
We number from zero, so to access the second column we use 1.

As well as issuing singular commands, we can run scripts from within QGIS.

5.2 Running Python scripts

Earlier, we used buffers to identify the most dangerous junctions. A major flaw with this method is that some accidents were counted more than once. In this example, the script iterates over each accident and finds its nearest junction.

1. From the Python Console click the *Show Editor* icon
2. A new, editing window will appear
3. In that window click *Open Script* and navigate to the *junctions.py* script
4. Spend some time and see if you can determine how the script works, then click the *Run Script* button
5. Note the output to the Python console
6. Note the contents of the Map window and note how the script selects the most dangerous junction and zooms to it:



12. Modify the script and change the junction threshold to 10m. Does this affect the result?

We can also use Python to edit the contents of a layer.

7. In the same editing window or in a new one (click *New Editor*) open the script *accidents.py*

This script again identifies the nearest junction to each accident but this time writes the junction ID to a new column in the accidents layer (*node_id*).

8. Open the accidents layer attribute table to check the output
9. Label the nodes with the node ID and then use the *Identify Features* tool to check the results.

6 Data from web services

So far we have been using spatial data held in files that have been downloaded from data portals. In QGIS it is also possible to consume spatial data provided by web services.

Web Mapping Services (WMS) deliver spatial data as raster data. Web Feature Services (WFS) deliver spatial data in the vector format. We will use the State of Massachusetts's excellent 'MassGIS'¹³ WFS server for our WFS experiments.

14. Create a new QGIS project
15. Add a new WFS layer - *Layer > Add Layer > Add WFS Layer*
16. Under *Server Connections* click *New*
17. Under *Name* enter MassGIS
18. For the URL enter:

`http://giswebservices.massgis.state.ma.us/geoserver/wfs`

19. Click *OK*
20. Under *Server Connections* click *Connect*

A list of dataset served by the service will appear.

21. *Add* a number of layers then *Close* the window – you can use the *filter* box to search for topics
22. For each layer explore its attribute table – note that the standard vector tasks such as select by attribute can be carried out as for vector data stored locally
23. Open the *Properties* of a layer and view its *Source* information – what CRS is the data projected in?

In this example I have added Rapid Transit stations and lines, ferry routes and town boundaries:

¹³ <https://www.mass.gov/orgs/massgis-bureau-of-geographic-information>



The advantages of web services are that the data is not stored locally and that if the dataset is updated then you will always have the latest version. Of course, the latter relies on the data provider updating and maintaining the web service.

24. Experiment with adding some WMS layers from the MassGIS service, using the following URL:

<http://giswebservices.massgis.state.ma.us/geoserver/wms>

25. Earlier we added Google Maps as a backdrop (page 2). Add the Bing Aerial layer as context using the URL:

<http://ecn.t3.tiles.virtualearth.net/tiles/a{q}.jpeg?g=1>

7 Spatial analysis with GeoDa

“The first law of geography: Everything is related to everything else, but near things are more related than distant things.”

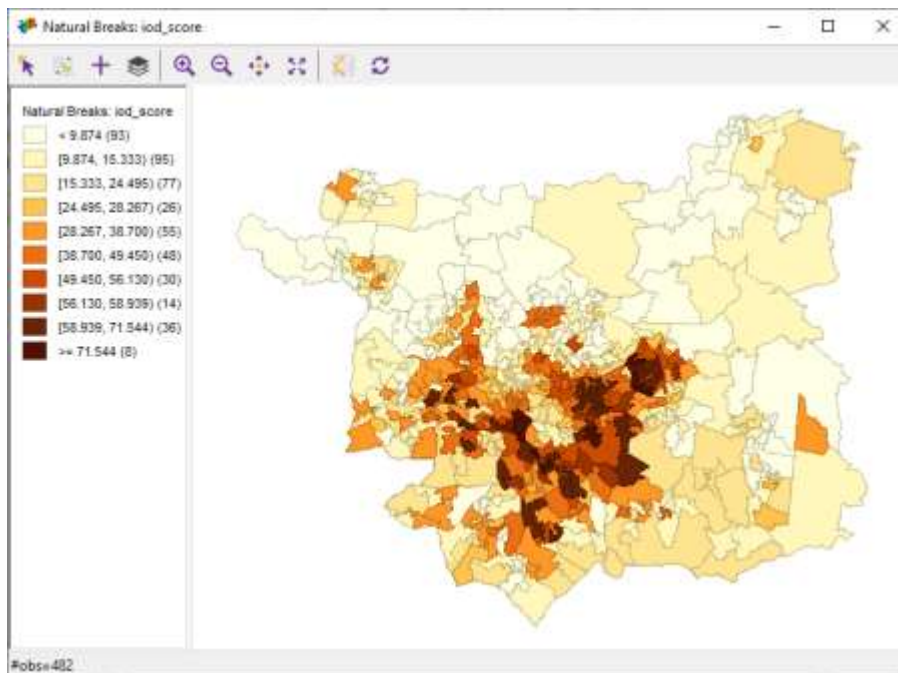
This session so far has focussed on using QGIS, a general purpose GIS, for spatial analysis. But there are more specialised open-source GIS. In this section we will use the *GeoDa* application to investigate spatial autocorrelation; *the correlation of a variable with itself over space*.

1. Download¹⁴ and install the *GeoDa* application, this should only take a few minutes

¹⁴ <https://geodacenter.github.io/download.html>

2. Open GeoDa; you will be asked to open a data source. Here drag the *crime_deprivation.shp* file from Windows Explorer to the window
3. To create a simple choropleth, from the menu bar select *Map > Natural Breaks Map* then the number of classes. You will then be asked to select a variable to map – choose *iod_score*

The output should look like:



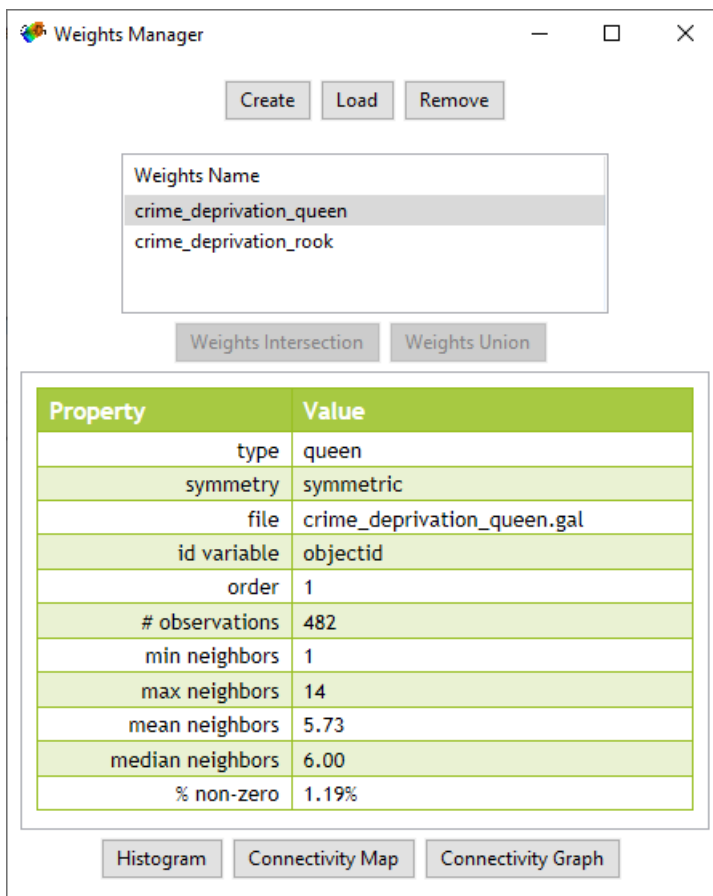
This is no more than what QGIS can do (although a lot more easily!). GeoDa's strengths are in its analysis. We are now going to exam the spatial autocorrelation of the deprivation data. Firstly, we have to generate a *Weights* table.

The *Weights* table defines how we express the weight we give to each element's neighbours. Here we are going to use "contiguity" measures¹⁵.

4. From the menu bar select the *Weights Manager* (a big W)
5. Click *Create*
6. From *Select ID* variable choose *objectid*. GeoDa will use this field to uniquely identify each polygon
7. Ensure *Queen Contiguity* is selected then click *Create*
8. Save the weight set as *crime_deprivation_queen.gal* in your working folder
9. Now choose *Rook Contiguity* and click *Create*
10. Save the weight as *crime_deprivation_rook.gal* in your working folder
11. Close the *Weights File Creation* window

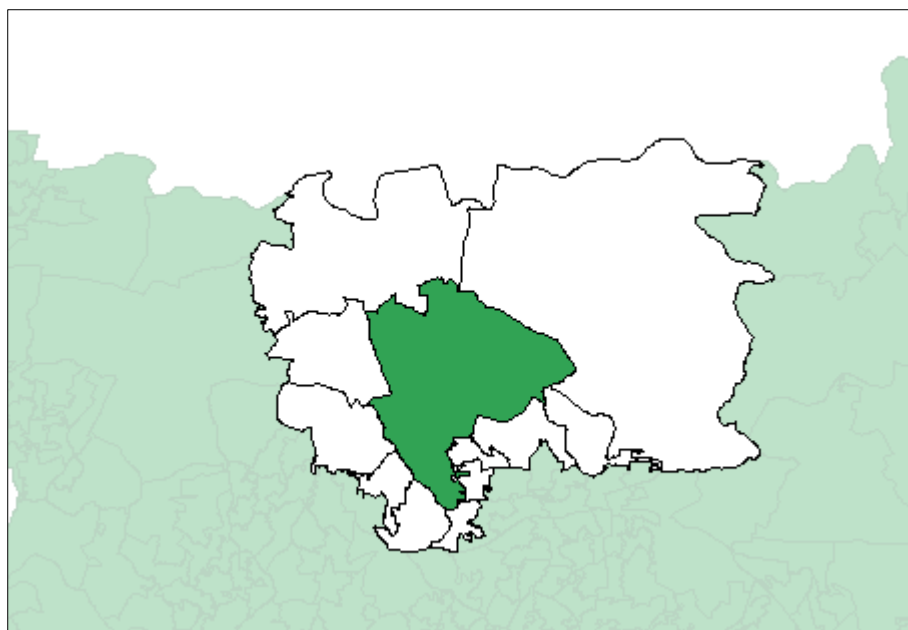
¹⁵ https://geodacenter.github.io/workbook/4a_contig_weights/lab4a.html

12. In the Weights Manager window, select *crime_deprivation_queen* and *crime_deprivation_rook* in turn, noting the values for “neighbors” - is there a significant difference?



Property	Value
type	queen
symmetry	symmetric
file	crime_deprivation_queen.gal
id variable	objectid
order	1
# observations	482
min neighbors	1
max neighbors	14
mean neighbors	5.73
median neighbors	6.00
% non-zero	1.19%

13. Choose either queen or rook and click *Connectivity Map*:



As you mouse over each LSOA it will display which others LSOAs will be treated as neighbours in the analysis.

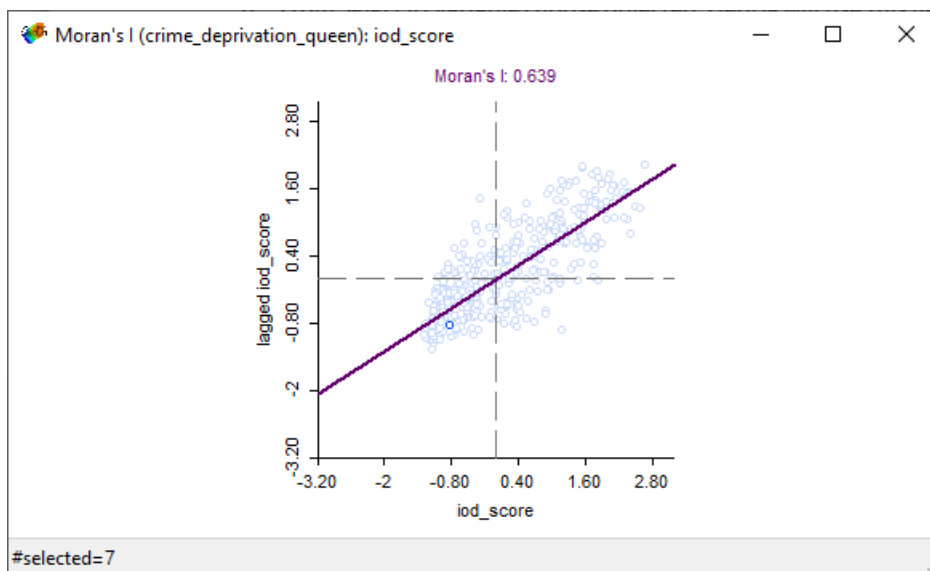
14. Open a *Connectivity map* for the other weighting
15. Notice as you select polygons in each map they are also selected in any other map.

7.1 Global spatial autocorrelation

The first test we are going to perform is a global measure of spatial autocorrelation – Moran's I ¹⁶

16. From the main menu bar select *Space > Univariate Moran's I*
17. Select *iod_score* as the variable to investigate and then select a weights table

The output will display a scatter chart and the Moran's index:



Here a value of between 0 and 1 indicates that there is positive spatial correlation of deprivation in the region. This is not unexpected given our interpretation of the maps of deprivation in QGIS. A value of zero would mean there is no spatial autocorrelation (randomly dispersed data)¹⁷.

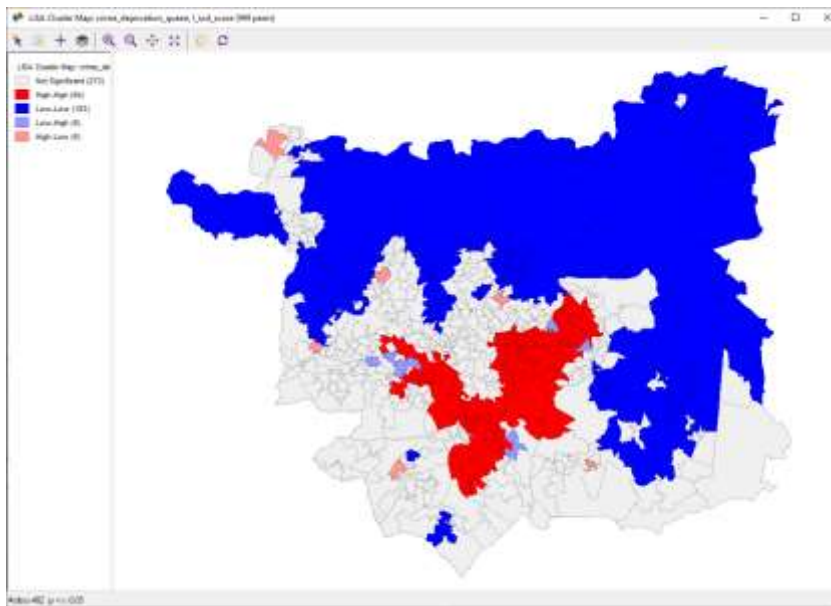
7.2 Local spatial autocorrelation

We can also examine autocorrelation at a local level.

18. From the main menu bar, select *Space > Univariate Local Moran's I*
19. Again chose *iod_score* as the variable and one of the set of weights
20. From the next window check all three boxes
21. Examine the cluster map:

¹⁶ https://geodacenter.github.io/workbook/5a_global_auto/lab5a.html#morans-i

¹⁷ See here on how to interpret the index - www.statisticshowto.datasciencecentral.com/morans-i/



Here the pattern of spatial autocorrelation is clear – the areas of high deprivation surrounded by neighbours also with high deprivation (High-High) and similarly, the areas of low deprivation with neighbours also of low deprivation (Low-Low).

As ever, the interesting results are the outliers, Low-High and High-Low.

Refer to the GeoDa documentation for a full guide to interpreting the outputs¹⁸.

22. Repeat the analysis, but this time use the *crimes* variable. You can use the same sets of weights you generated earlier. How does the value for global Moran's I and the patterns for local Moran's differ from the deprivation data? Which of the two variables, crime or deprivation shows more evidence of spatial autocorrelation?

8 Licensing

When using data supplied by a third party it is important to provide acknowledgements.

For example, for the Ordnance Survey Open Data used earlier, we would add the following to any outputs:

Contains Ordnance Survey data © Crown copyright and database right 2019

Document revision history

Version 1: Nick Gould – nickgould@live.co.uk - December 2019

¹⁸ https://geodacenter.github.io/workbook/6a_local_auto/lab6a.html