

Open Source Data and Methods: A tutorial using OSM data with QGIS for location analysis

This tutorial illustrates the huge potential offered by the crowd-sourced data model, underlines its flaws and demonstrates methods for overcoming issues of quality and inconsistency. This tutorial will show you what a crowd-sourced geographic dataset looks like, and illustrates its potential. In the homework you will contribute to the Open Street Map (OSM), the largest crowd-sourced map project on Earth, by uploading real data for the benefit of yourself and anyone around the world.

This tutorial is available from the Elogeo repository and can be found by searching for it here: <http://elogeoh.nottingham.ac.uk/xmlui>. Up-to-date versions, including input data are here: <https://dl.dropboxusercontent.com/u/15008199/osm-challenge-public.zip>

Please send any comments/suggestions to rob00x@gmail.com or <http://robinlovelace.net>

Introduction

This session builds on the “food desert” analysis, by introducing a new dataset of supermarkets in Sheffield, taken from the Open Street Map. In Task 1 you will be introduced to OSM data and the program QGIS, by making a custom map of an area you are familiar with. Task 2 shows the real potential of OSM data, by illustrating how it can be used to analyse spatial monopolies of supermarkets operating in Sheffield. The aim is to illustrate the near endless possibilities that arise from combining crowd-sourced data with knowledge of the local area and modern GIS programs.

A sub-aim is to promote an open-source ethic in general: open-source data and programs can help remove the profit motive from digital mapping and make GIS more inclusive.

Software used

The tutorial was developed in QGIS 2.0, a free and open-source GIS program, updated to the latest version in summer 2013. It is recommended that you follow the tutorial using QGIS as it is free, stable and has add-ons specifically designed for dealing with OSM data. The tutorial can also be conducted using other software but it may not be possible to undertake all of the steps. In cases where tasks cannot be completed, you can cheat, by opening pre-made files that were saved in the previous step, based on their filename.

Instructions on how to install QGIS are available at www.qgis.org. The tutorial assumes no previous experience of QGIS, so will also serve as an introductory tutorial to the benefits of open-source GIS, as well as crowd-sourced data.

The crowd-sourced data model

Fast computers, broadband access and the smart phone revolution have led to changes in information access. Database interaction now need not be a simple one-way process whereby a centralised “system administrator” role (e.g. Google Maps) sends information to a passive user in a closed cycle. The crowd-sourced data model (perhaps best exemplified by Wikipedia) involves user feedback at every stage, which requiring more communication in the system (Fig. 1).

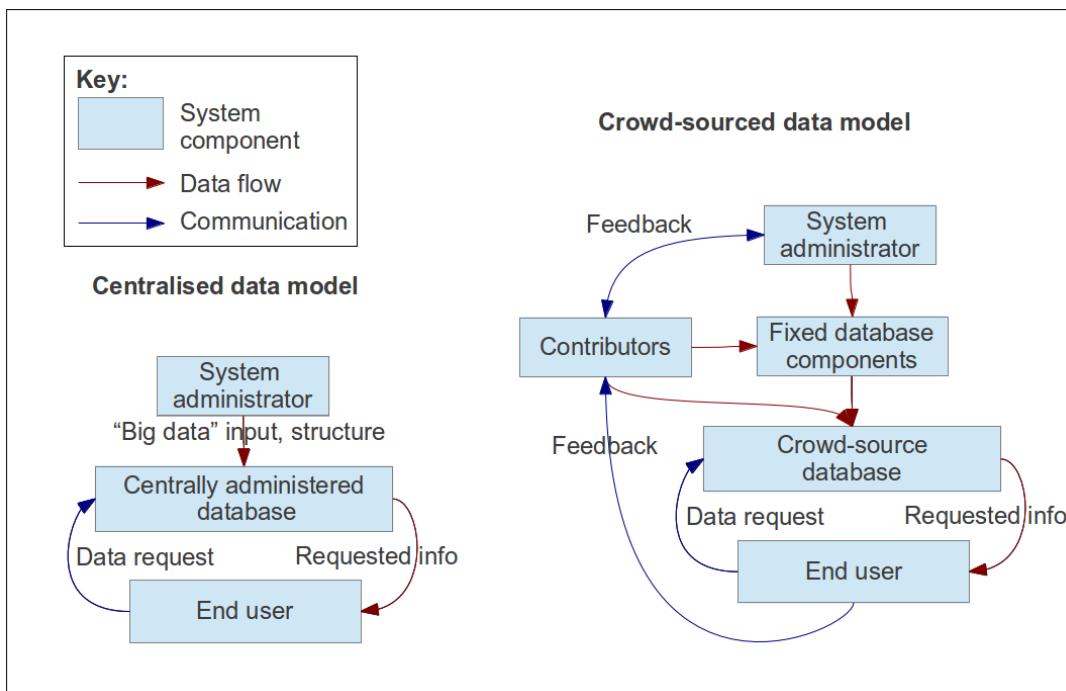


Fig. 1: Schematic diagrams of centralised and crowd-sourced data models.

Open Street Map

Open Street Map (OSM) is the most widely used crowd-sourced map worldwide. Because the data is made available to external organisations, OSM has come to form the backbone of a number of other online maps, notably Cloudmade, which facilitates mapping applications for mobile devices.

OSM data availability

Open Street Map is usually used like Google maps: checked over briefly, for example to find out how to get from A to B in a new city. The key difference, however, is that Google hides the raw data behind its maps. The user only ever sees a series of images, with no way to analyse the data. You cannot, for example, ask Google maps to draw buffers around certain features such as parks or to you show only large roads.

OSM data, by contrast, is Open Source. That means that anyone can download or modify any part of the georeferenced data, according to its legally defined Open database Licence (ODbL). However, getting the useful OSM data onto your desktop GIS may not be as easy as you might imagine (see Appendix “Importing OSM data”).

Rather than worry about the details of downloading the data, we will assume that useful data has been converted into a GIS friendly form already, and proceed to explore the data.

Task 1: Explore and present OSM data (30 minutes)

As with any new skill or dataset, often the best way to start is to play about. Therefore this first task is not linear or prescriptive. Your objective is simply to create a map of an area of an area that you are familiar with, focussing on the features that most interest you (any part of the world can be accessed, but choosing an English location will provide a relatively small – county level – chunk of

rich information). It will also help you to get acquainted with QGIS. This first task is based on self-learning, so you figure out how to use QGIS yourself. If you get stuck at any point, check the QGIS user manual (<http://www.qgis.org/en/documentation/manuals.html>), search for help online, or ask for help.

The first stage is to download the OSM data, which have been handily processed into bitesized chunks by Geofabrik.de and made available at the following URL:

<http://download.geofabrik.de/openstreetmap/>

If you are looking for South Yorkshire data, for example, navigate into Europe, Great Britain, England, until you find the list of English counties. You will be presented with 3 options for each area:

<input type="checkbox"/> south_yorkshire.osm.bz2	31-Oct-2012 09:24	15M OpenStreetMap data, bzip2 compressed
<input type="checkbox"/> south_yorkshire.osm.pbf	31-Oct-2012 05:25	9.9M OpenStreetMap data, protobuf binary format
 south_yorkshire.shp.zip	31-Oct-2012 05:47	15M ESRI Shapefile (EPSG:4326), zipped

The most user-friendly of these for desktop GIS users is the third option, the trusty old shapefile, provided in a zipped folder.

Once you extracted the data in a suitably named folder (e.g. “soyo-shp-osm”) in your partition of the computer’s hard drive, you are ready to go. It is recommended that easy-to-interpret layers such as roads and rivers are loaded into your GIS first, followed by less distinctive spatial datasets such as points. Play around!

Playing around

Focussing on things that interest me – pubs, bicycle paths and how to get to the Tyrrell’s crisp factory from Weobley and Hereford, I created the following map (Fig. 2).

Do not waste time fiddling with the style for each to begin with. The important thing is to explore the data. Are bus-stops (nestled within the “points.shp” file), for example, located where they should be? Are the shops (confusingly located in both points and buildings) correct – what’s missing?

List any issues on a piece of paper, we’ll discuss them before starting Task 2.

If your specific area of interest is not well catered-for, consider changing your area of interest. The aim is to see what OSM data IS good for and to get acquainted with QGIS for creating a simple map.

Tip: if you want to see an overview map of your area to put it in context, try downloading the OpenLayers plugin from the plugin menu. There are dozens of other excellent plugins, take a look around.

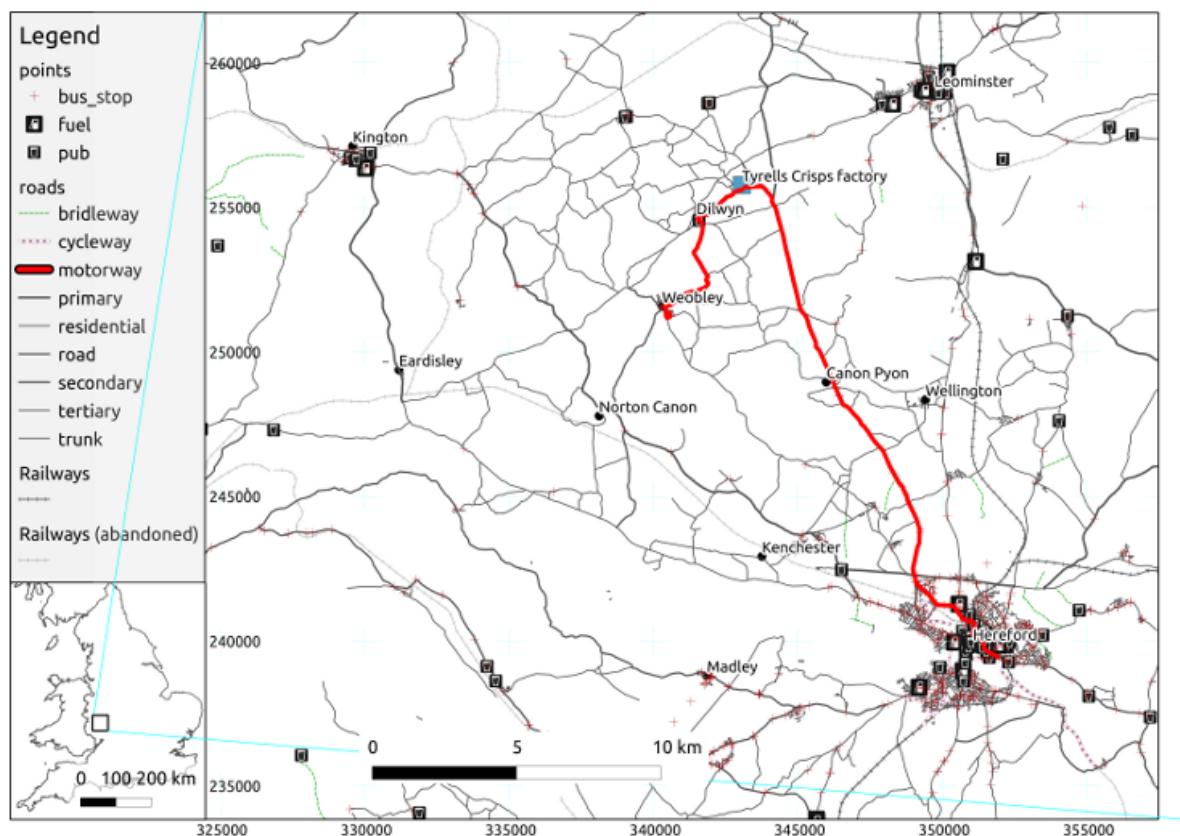


Fig. 2: A map of OSM data focussed on pubs, petrol stations, and Tyrrell's crisp factory.

Export your map with Print composer

When all the data you want has been loaded into the GIS, try using the map composer window to create a ready-to-print map. Once you have understood the basics of QGIS, it's time to move on to Task 2.

Task 2: Import and clean the data (1 hour)

In this task you will process OSM data on supermarkets to investigate spatial monopolies in south Yorkshire. You may have noticed that the points provided in the shapefiles downloaded from www.geofabrik.de do not contain supermarkets. The data for this tutorial are available from here: <https://dl.dropboxusercontent.com/u/15008199/osm-challenge-public.zip>. Download the file “osm-challenge-public.zip” and extract the folder into a sensible location. You will find three sub-folders: “input-data”, “processed” and “soyo”, in addition to this tutorial and some related presentation slides. The vital input datasets are saved in the folder – wait for it – “input-data”!

Open the “supermarket-polygons” and “soyo-points” files. Take a look at their locations and their attributes. Do the data make sense? The map should look like this (the project's CRS should be set to “OSGB 1936 / British National Grid”, by clicking Settings > Project Properties in the top menu):

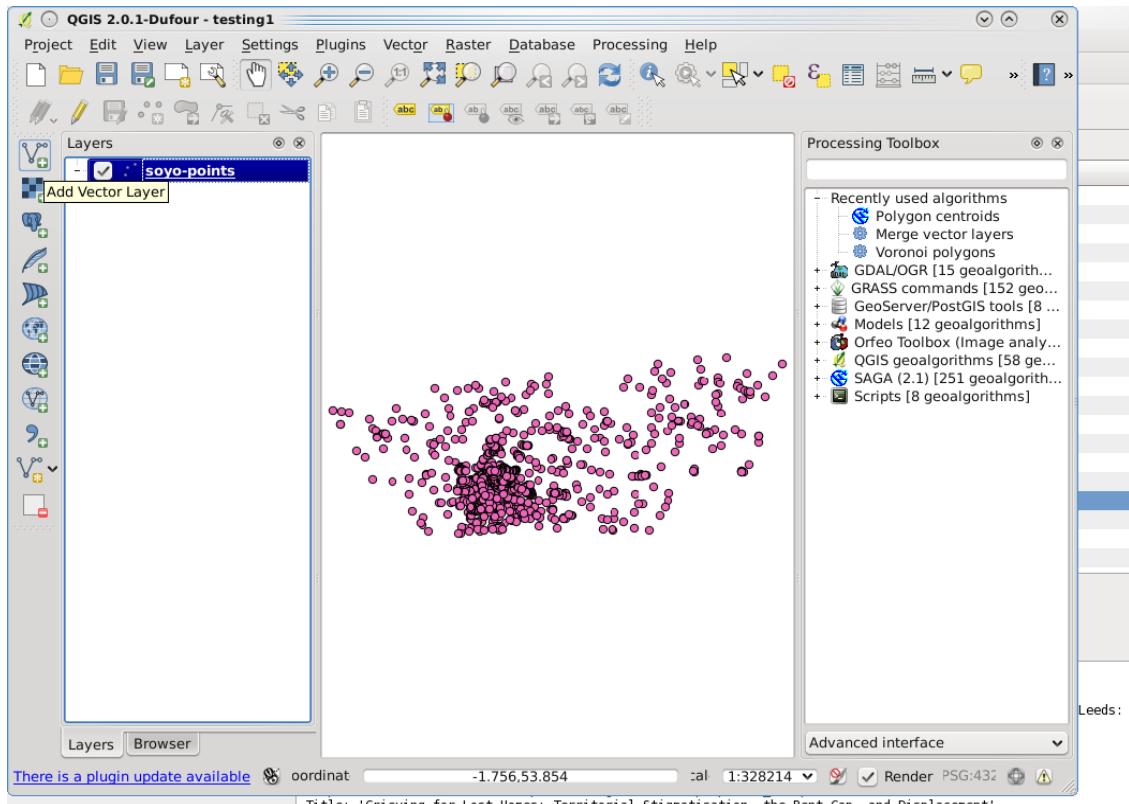
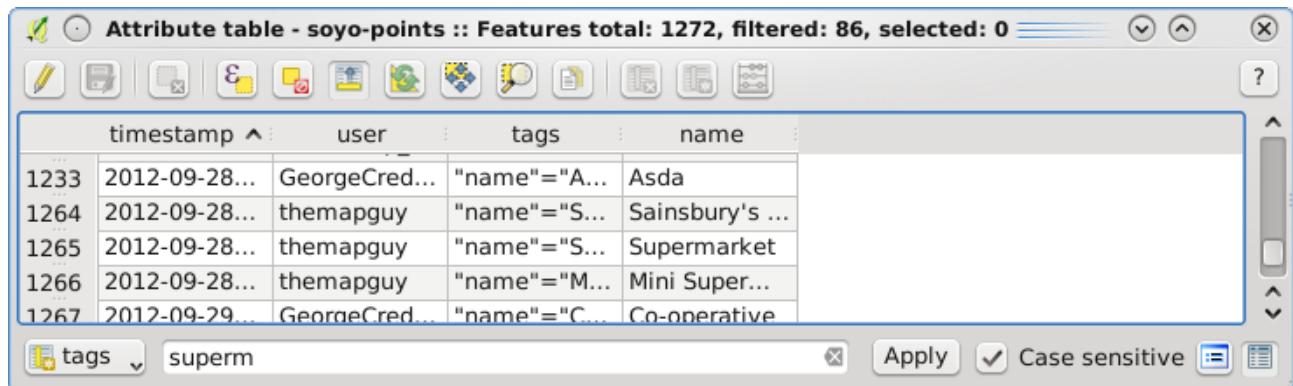


Fig. 3: Screenshot of the point data loaded into QGIS. Note the “Add vector layer” dialogue and button at the top.

Fig. 3 shows there is an excessive amount of information, with more than 1000 points – not all of them are supermarkets, so many are not needed now.

We will create a new layer, with the excess points removed. Open the attribute table – by

right-clicking on the layer – and click on “tags” inside the “Column Filter” menu in the bottom left hand dropdown menu. To show only supermarkets type “superm” in the dialogue box in the variable (Fig. 4). This will show all the points tagged as supermarkets. Select them with the mouse and check the map – the supermarket points should change colour.



The screenshot shows the QGIS attribute table for the "soyo-points" layer. The table has columns: timestamp, user, tags, and name. There are 1272 total features, 86 filtered, and 0 selected. A filter bar at the bottom shows "tags" and "superm".

	timestamp	user	tags	name
1233	2012-09-28...	GeorgeCred...	"name"="A...	Asda
1264	2012-09-28...	themapguy	"name"="S...	Sainsbury's ...
1265	2012-09-28...	themapguy	"name"="S...	Supermarket
1266	2012-09-28...	themapguy	"name"="M...	Mini Super...
1267	2012-09-29...	GeorgeCred...	"name"="C...	Co-operative

Fig. 4: Attribute table of soyo-points after filtering for supermarkets.

Next, right-click on the layer “soyo-points” and click “Save Selection As...”. You can save the layer as “supermarket-points” in the folder “processed”. As you will see, this shapefile already exists – as a backup in case you go wrong – just save over it or give it a new name. (The polygon data was pre-filtered from the .osm file in the same way.) Now remove the original “soyo-points” layer from your project – it’s a distraction (right click on the layer and left click “Remove”).

Now you have cleaned the point data and loaded the polygon data, an immediate issue should become clear: non-compatibility between point and polygon data. The polygon shapes *are* useful for some purposes (providing an estimate of store area), but for the current task points will do.

Task 2.1: Convert the supermarket polygons into points

To convert the supermarket-polygons layer into points, we will use the “Polygon centroids” command. To find this, click on “Processing” in the top menu then select “Toolbox”, if the toolbox is not already activated. Search for “centroids” in the search bar, and the function should appear as shown in Fig. 5. (Linux users: SAGA must be set-up correctly for this to work.) Save the output file as “points-from-polys”, also in the “processed” folder (right-click, save layer) and add it to the map. Your map now should look a little like this (Fig. 5):

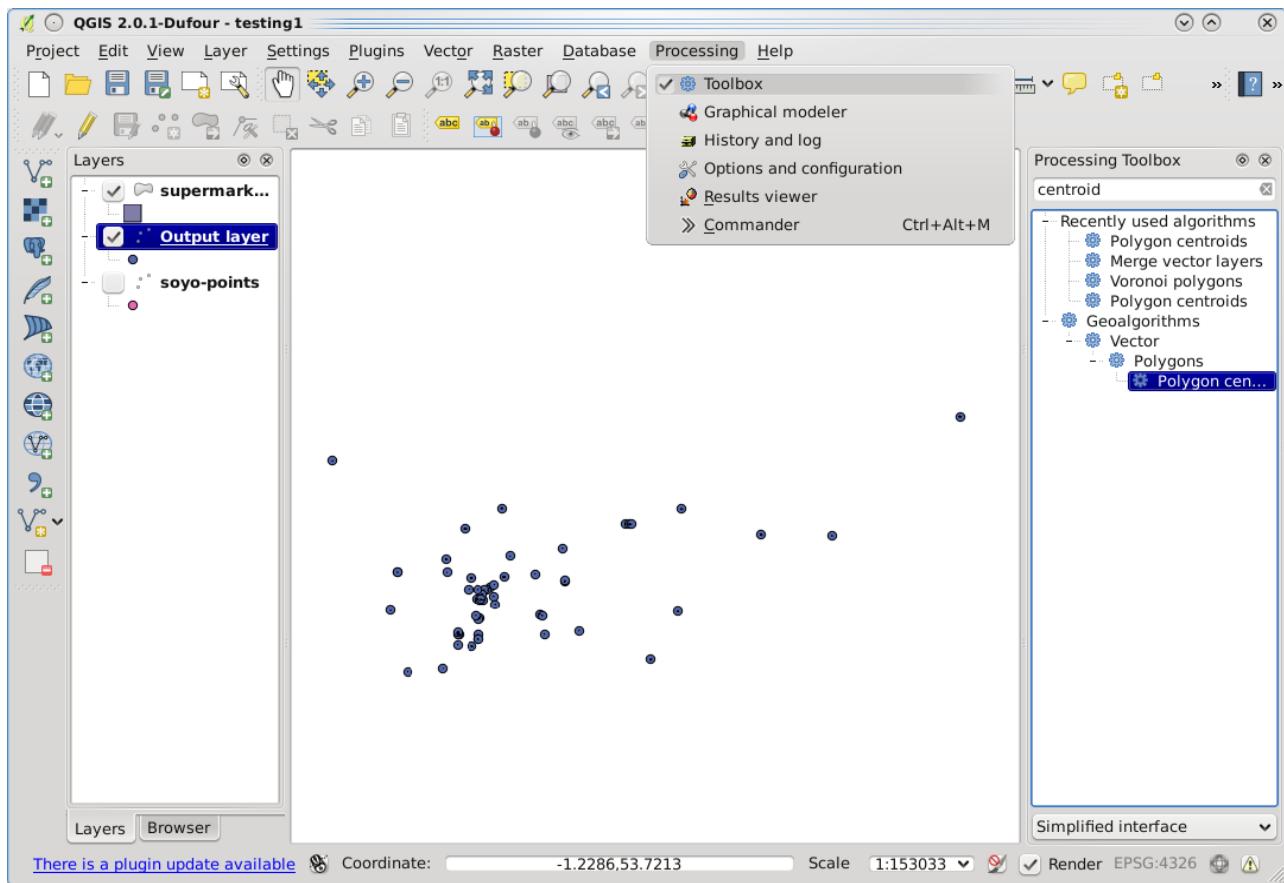


Fig. 5: The result of running the points-to-polygons command (red dots).

Task 2.2: Merge the supermarket points layers

Once the new points have been created, merge this new layer with the pre-existing supermarket-points (your challenge: find out how) and save the merged layer as “merged”. Take a look at the attributes of this new file.

Task 2.3 Standardise the “messy” names

Open the “merged” file and look at its attributes. You should notice is that the data is irregular – there are many different names to describe the same company. The Cooperative, for example is described in many ways including “Co-op”, “The Co-operative”, “Coopoerative” (Fig. 6):

	A	B	C	D	E	F
1	timestamp,C.255	user,C.255	tags,C.255	The Co-operative	place,C.255	amenity,C.255
2	2011-06-03T11:34:53	Matt Cleaver	"building"="commercial","name"="The Co-operative","shop"="supermarket"	Co-Op Swallownest		
3	2011-06-04T08:22:25	stordoff	"name"="Co- Op Swallownest","shop"="supermarket"	Asda		
4	2012-09-08T15:38:35	RobinLovelace	"name"="Asda","shop"="supermarket","source"="Bing"	Wilkinson's		
5	2011-06-11T09:51:49	Meersbrook	"building"="yes","name"="Wilkinson's","shop"="supermarket"	Aldi		
6	2012-08-10T14:48:50	Meersbrook	"building"="yes","name"="Aldi","shop"="supermarket"	The Co-operative Food		
7	2012-04-06T09:35:57	Matt T	"building"="yes","name"="The Co-operative Food","operator"="The Cooperative"	Tesco Local		
8	2011-10-09T17:38:04	Waugh Bacon	"name"="Tesco Local","shop"="supermarket","source"="local_knowledge"	Sainsbury's		
9	2011-10-02T12:12:16	Meersbrook	"building"="yes","name"="Sainsbury's","shop"="supermarket"	Tesco		
10	2012-07-21T14:54:06	Meersbrook	"addr:postcode"="S7 2QB","addr:street"="Abbeylease Road","building"="yes","name"="Netto"	Netto		
11	2011-10-02T14:57:05	Meersbrook	"building"="yes","name"="Netto","shop"="supermarket"	Iceland		
12	2011-10-02T14:57:06	Meersbrook	"building"="yes","name"="Iceland","shop"="supermarket"	Spar		
13	2012-10-27T13:34:26	Meersbrook	"addr:street"="Abbeylease Road","building"="yes","name"="Spar","shop"="supermarket"	Tesco Extra		
14	2012-08-08T13:35:50	Meersbrook	"building"="yes","name"="Tesco Extra","shop"="supermarket"	Radway Post Office		
15	2011-11-27T14:55:11	Meersbrook	"building"="yes","name"="Radway Post Office","operator"="Spar","shop"="supermarket"			

Fig. 6: Screenshot taken from the .dbf file of the supermarket polygons, opened in LibreOffice calc.

This is a good illustration of variable data quality inherent of crowd-sourced data. However, using our computing skills we can clean up the names. There are two ways to do this: the fast way and the slow way.

The slow way is to edit the attribute table manually, re-typing the desired name each time a faulty name appears. This works, but with 56 polygons and 85 points to rename, is rather boring, especially when we remember that computers were invented to automate repetitive tasks, not create new ones (think Angry Birds).

The fast way is to use a computer program to do the job. QGIS has an inbuilt regular expressions editor that can search for and replace patterns of characters. The field calculator (found within the Attribute Table panel) is used to do this. Let's see how to replace all names containing "Tesc" with "Tesco": (We need to do this to get rid of superfluous information like "Tesco, Infirmary road" or "Tesco local".) Simply type the following command into the field editor (Fig. 7). Warning – you need to enable editing first:

```
regexp_replace("name",'Tesc.*','Tesco').
```

Dump the results in a new field, (perhaps called "standard_names") by ticking "Create and new field" and typing the desired name. The Tesco example was quite easy. What about when we have many forms of a name, such as Co-op, Co-Op, The Co-operative, etc., just to say "Co-operative"?

Design a regular expression to capture all these meanings:

```
regexp_replace( "standard_names", '.*Co.*','Co-operative')
```

Note: the "+" symbol means the search includes any letters preceding "Co". The .* means anything following it to. Regular expressions are VERY USEFUL, it's worth spending some time experimenting with them to get all or your supermarket names standardised (see Fig. 8 for another example). Using command-line functions can be tricky – just the slightest typo can lead to errors – so feel free to skip to the next stage, by opening the file "supermarket-merged".

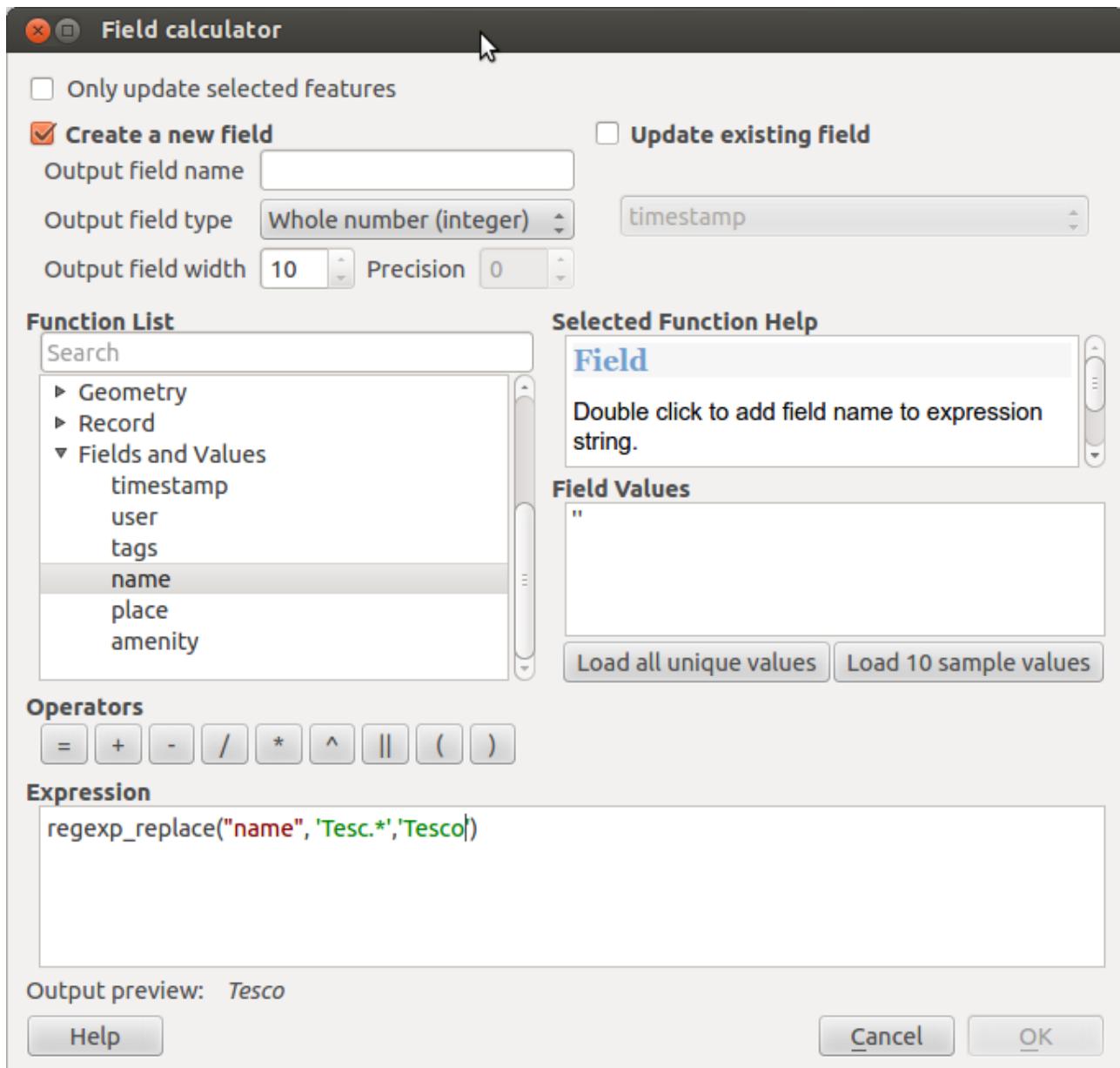


Fig. 7: Using regular expressions to standardise the names of all Tesco stores.

For more information about editing attribute data using QGIS, please refer to the follow question, asked to the community: <http://gis.stackexchange.com/questions/40637/how-to-easily-edit-attribute-data-using-regular-expressions>

This is also a good example of the support offered by the open-source GIS user community.

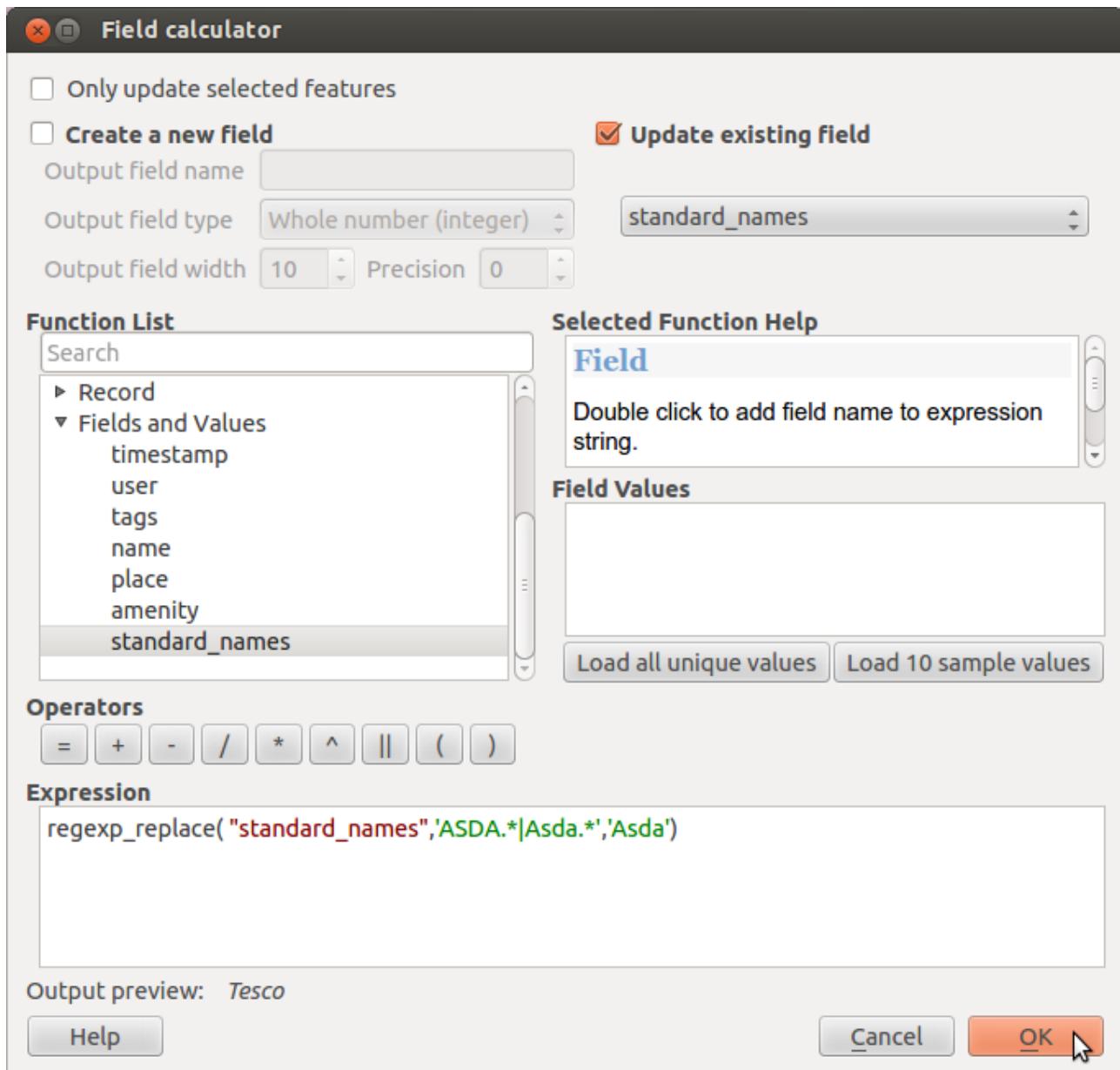


Fig. 8: Regular expression to standardise all cells containing Asda, ASDA and Asda local... The “|” symbol means “or”.

Task 2.4: Rename rare supermarkets as “Other”

For purposes of simplicity we can rename these as “Other” in the analysis we will focus on only 5 of the largest supermarkets, identified in Fig. 6. An easy way do this is to using a spreadsheet program such as Excel, to open and edit the .dbf file, saving the attribute table as a .csv file and then joining the edited attributes to the shapefile by timestamp. Bonus point: find a fast and effective way of doing this in QGIS. After your supermarkets have been simplified into 4 to 6 categories – manually or otherwise – its time to save your work and move on. De-activate edit mode and save your supermarket layer as “merged-joined”. (You can skip to the next stage by opening “supermarket-merged-joined”).

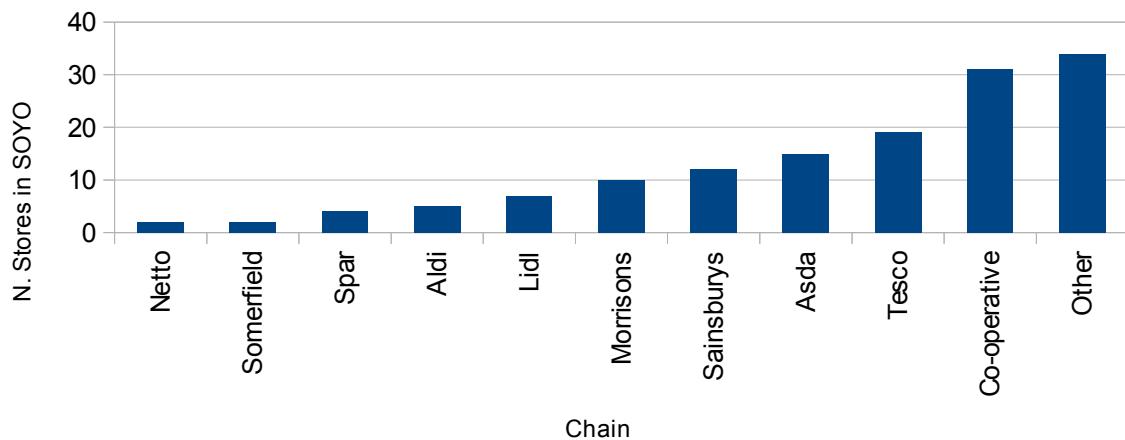


Fig. 9: Bar chart of the most numerous supermarket chains in South Yorkshire

Task 3: Analyse and polish results

This is where the fun starts. Now that we have consistent supermarket data, we can start the analysis. Create Voronoi polygons using your GIS skills. The results are shown below:

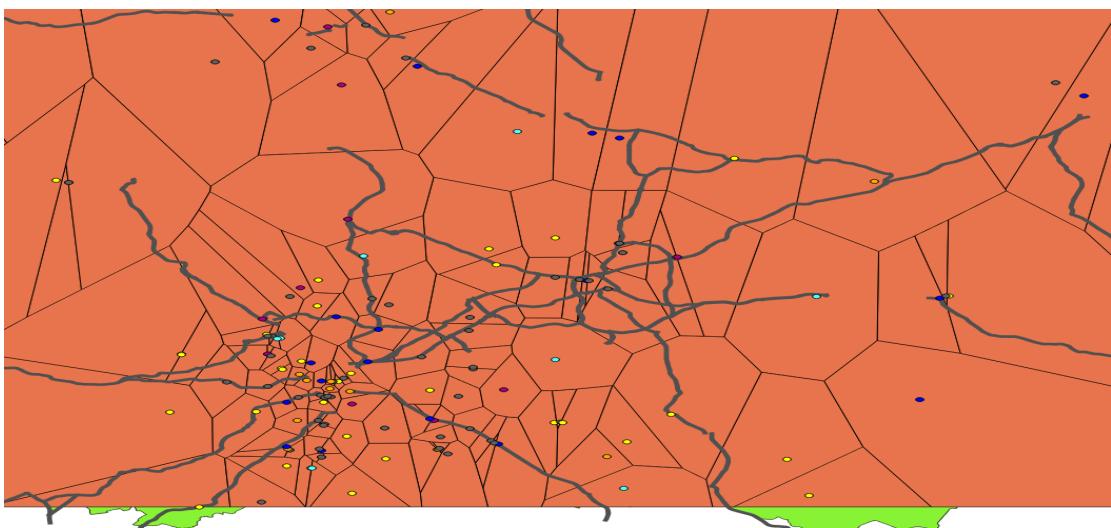


Fig. 10: Voronoi polygon of supermarket territories in South Yorkshire

Explore the voronoi polygons, zooming in on tightly packed area. You should notice that some polygons seem to cross directly over a single supermarket. Zooming in further should reveal that, in some cases, this is not due to supermarkets locating right next to each other (although this does happen – see Fig. 10), but due to replication of the supermarket data – they are represented as both points and polygons, as illustrated below (Fig. 11):

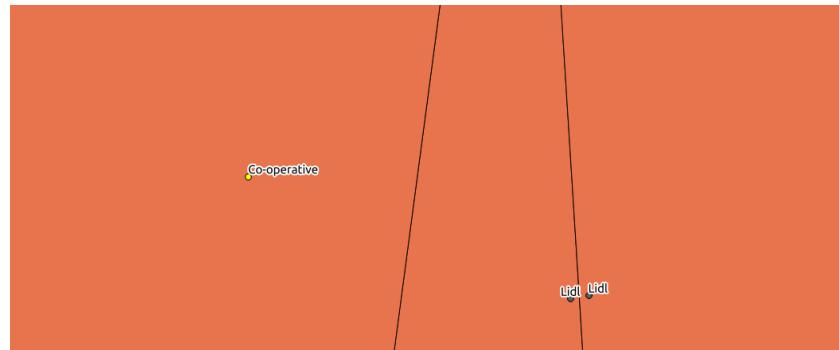


Fig. 11: Illustration of duplicate supermarkets leading to faulty Voronoi polygons

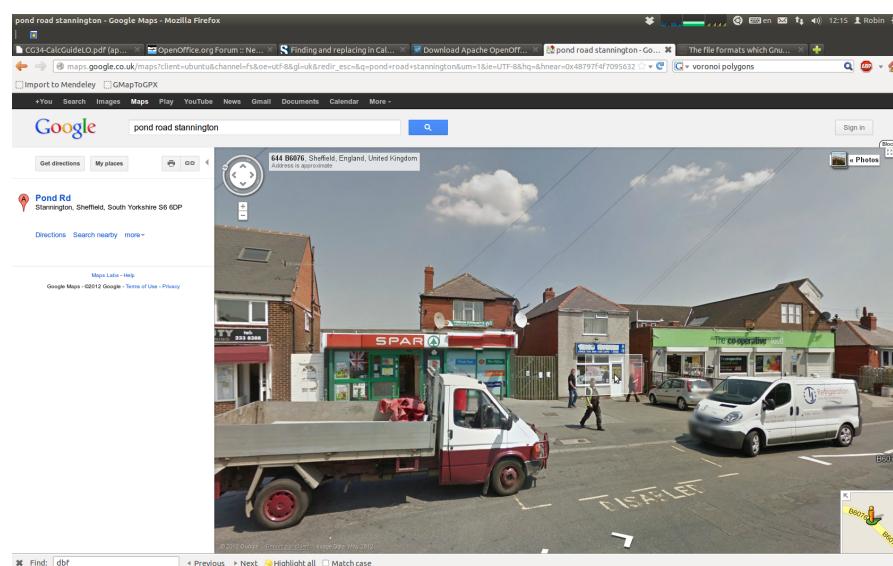


Fig. 12: Illustration of two supermarkets locating side by side, on Stannington Road.

To deal with this, clean-up the latest supermarket layer by removing duplicate supermarkets. Once this is done, re-create the Voronoi polygon. Notice that the attributes of the points have automatically been transferred to the voronoi polygon layer. This should allow the creation of a “territory map” to represent how supermarkets compete for space in South Yorkshire. Fig. 13 illustrates how your map should look (as a general guide) after today’s tutorial:

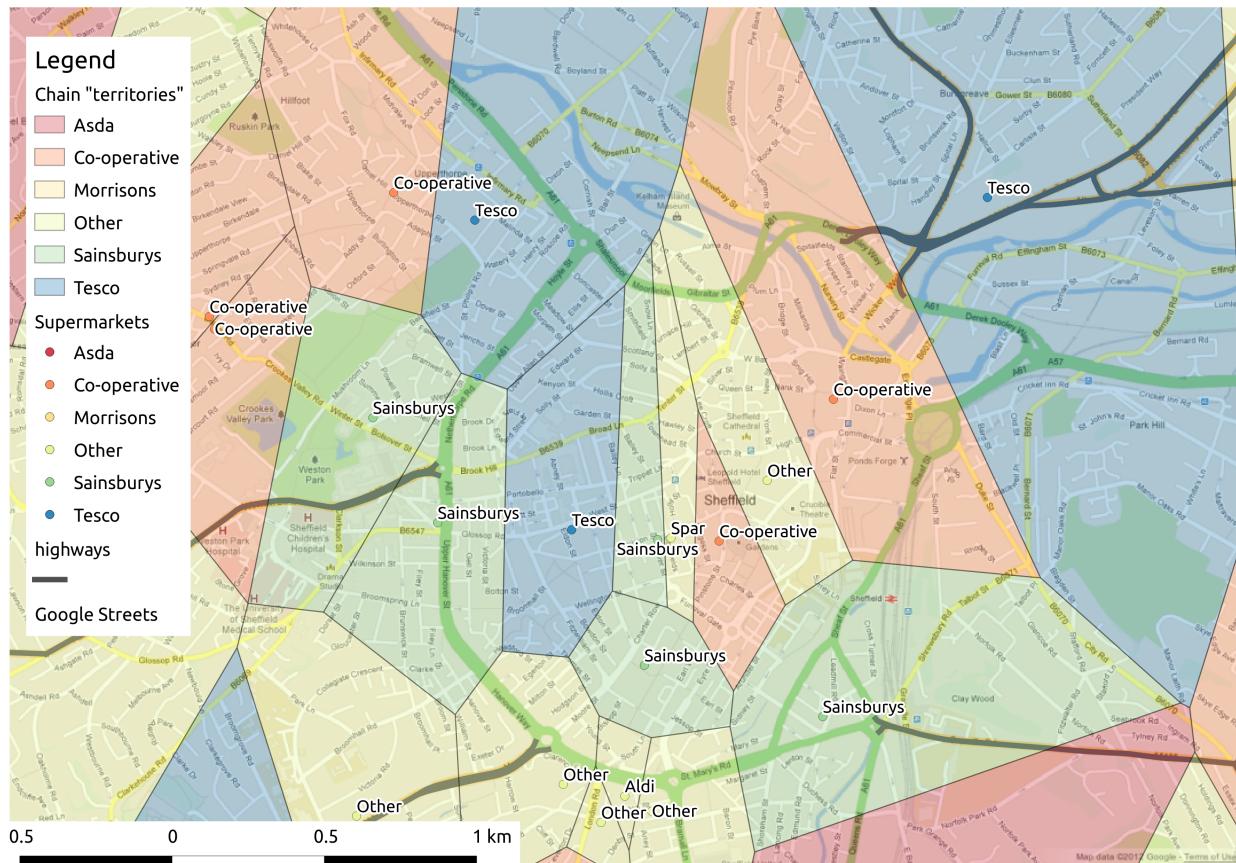


Fig. 13: Supermarket territories in central Sheffield. Note the large areas that surround large stores such as the Tescos in Burngreave (top right on the map) and the valuable “territory” surrounding the new Sainsbury's on Glossop Road.

Taking the analysis further

This exercise has illustrated the potential of open source data analysis for real-world applications. Although the data is of variable quality, this is likely to become less of an issue as more people contribute to the OSM database.

The analysis presented in this tutorial could be further developed in many directions, including:

- Estimation of the market share e.g. in terms of population with 1 km of each store, passing footfall and accessibility of the supermarkets and for each chain.
- Investigation of the link between supermarkets and smaller stores such as greengrocers, butchers etc. Hypothesis: these stores tend to survive only far from large supermarkets.
- Investigation of territorial battles over time, using historical OSM data.
- Web-mapping applications to show where people are highly reliant on a few large chains.

Optional homework: investigate supermarket size and spatial monopolies

Another way to take the analysis further is to analyse floor space, represented by the area of the supermarket polygons. Early in the practical, we converted these polygons into point data to merge with the more commonly found point data. This is not ideal, as it loses some useful information about store size.

Your homework for this week is therefore to convert at least one supermarket point into a supermarket polygon. Then investigate the links between store size and the 'territory' of each supermarket using basic statistical and geoprocessing techniques such as correlation calculations and buffers.

Appendix: Importing OSM data

An overview of the options is available here:

http://hub.qgis.org/projects/quantum-gis/wiki/Using_OpenStreetMap_data. There are, at the time of writing (November 2012), three main options:

- Download the data directly from www.openstreetmap.org . This is quick and easy, but only allows a maximum of 50,000 nodes (a small fraction of a city) to be downloaded, so is not suitable for anything but local analysis. Another downside of this for some GIS programs such as ArcGIS is that the output filetype (*.osm) cannot be read directly.
- Download pre-processed data from a third party such as www.geofabrik.de. This option is advantageous as pre-made chunks of data, corresponding with the different areas are made available. However, the data is often not up-to-date, and may not be comprehensive.
- Download larger areas (up to the entire planet) from regularly updated versions of planet.osm: <http://wiki.openstreetmap.org/wiki/Planet.osm> . The disadvantage of this is that files can be very large; the advantage is the data will be up-to-date and comprehensive.

Because the datasets can be very large, the ultimate solution is to use a geo-database such as Oracle spatial (proprietary) or PostGIS. A very brief introduction to the latter is as follows:

First a PostGIS database must be set-up, with the correct user permissions, projections, and tables. The data for Sheffield was downloaded from <http://download.bbbike.org/osm/bbbike/> (the command-line tool osmosis can also be used to cut-up national osm datasets). The following command was used to extract this data and place it into the PostGIS database entitled "gis". This is complex process with many obstacles. If you're confident, and have spare time, give it a go! (click [here](#) for a good reference on topic).