INTRO TO MAPPING WITH QGIS

Robert Gebeloff
@gebeloffnyt
Dataharvest 2017

In the hierarchy of data journalism, there are spreadsheets. And database managers. And then there's mapping -- another powerful way to analyze data. While working with data tables can tell you a lot about a dataset, plotting values on a map often reveals patterns that cannot be seen in tables of figures and charts.

Traditionally, the data journalism community has embraced mapping software called Arcview, sold by ESRI, and many consider ESRI products to be the state-of-the-art.

But for small and non-profit news organizations, free, open-source software is a more practical solution, and in that sense, QGIS fits the bill. QGIS is also lauded because it supports both PCs and Macs.

QGIS is slightly more quirky and bug-prone than Arcview, but will almost always serve as an adequate substitute for the type of analytical mapping most data journalists wish to perform.

In this class, we will introduce you to mapping with QGIS -- demonstrating how to create maps with polygons, how to navigate the map with the software, and then how to join data for analysis into the map. We will also briefly touch on how to add "point" data to the map --data that has geographic coordinates (latitude, longitude) attached.

But first, of course, you need to install the software…

The Windows download page is here:

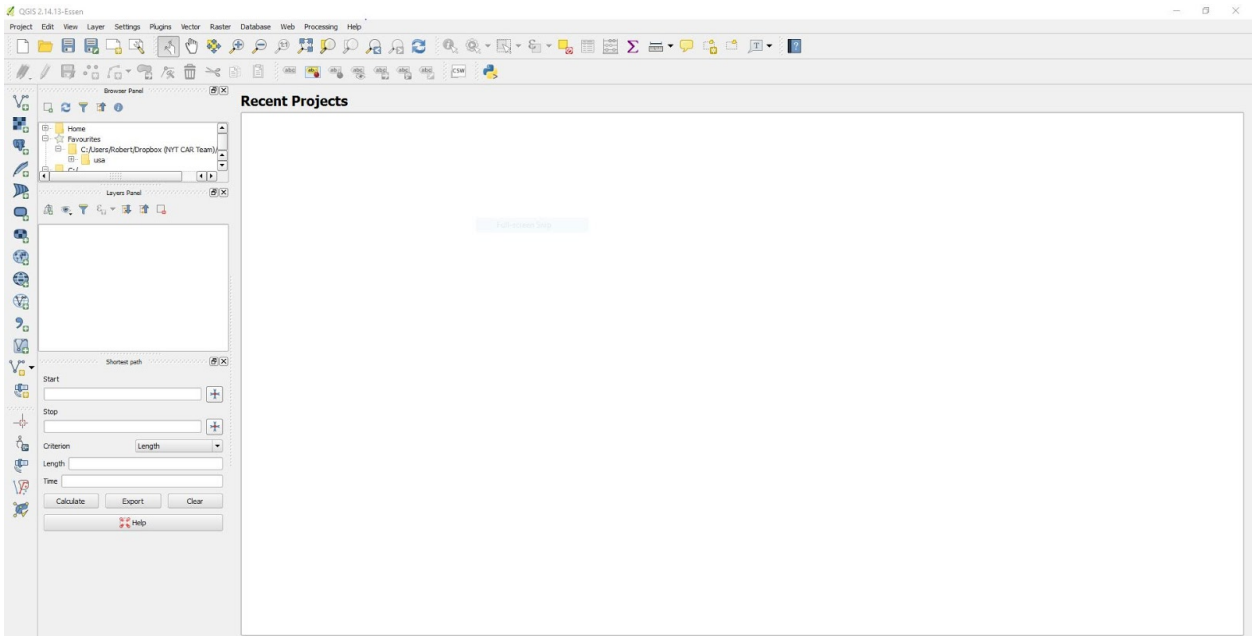
https://www.qgis.org/en/site/forusers/download.html

For this class,we're going to skip the very recent new version and go to the stable previous version 2.14. If you're on a newer system, you can download the 64-bit standalone installer, for older systems, you'll need 32-bit.

The download tends to be slow, but the install is straightforward. You can install the practice datasets if you want, but we're going to use our own data in this lesson.

For Mac users, the software is hosted by a 3rd party and can be downloaded here:
http://www.kyngchaos.com/files/software/qgis/QGIS-2.14.10-1.dmg

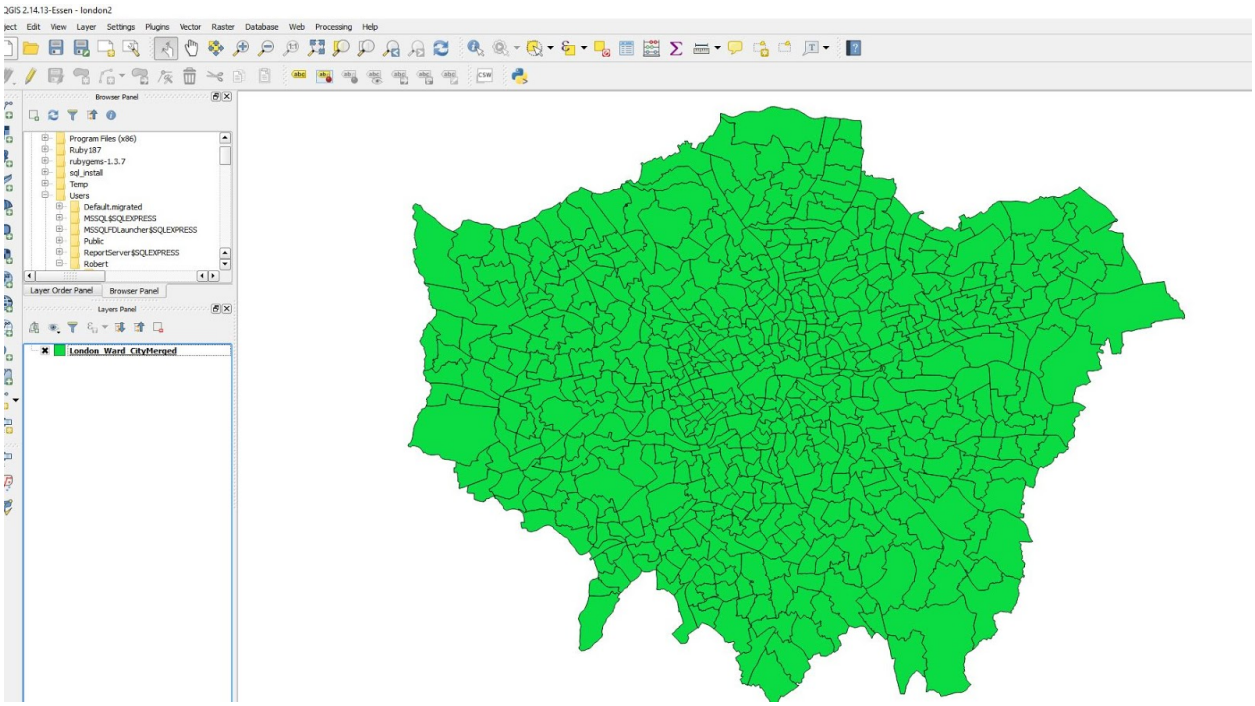
Note that because this isn't "official" Mac software, you will probably need to right-click on the download and then select "Open" instead of simply double clicking on the download file. You will also have to install some complimentary packages included in the dmg - Gdal, NumPy, matplotlib -- before installing the main QGIS file.



Like many programs, you can operate QGIS using the text menus on top, or by using the buttons along the top and down the left side. The best way to become familiar with these options is to load a map file and explore it.

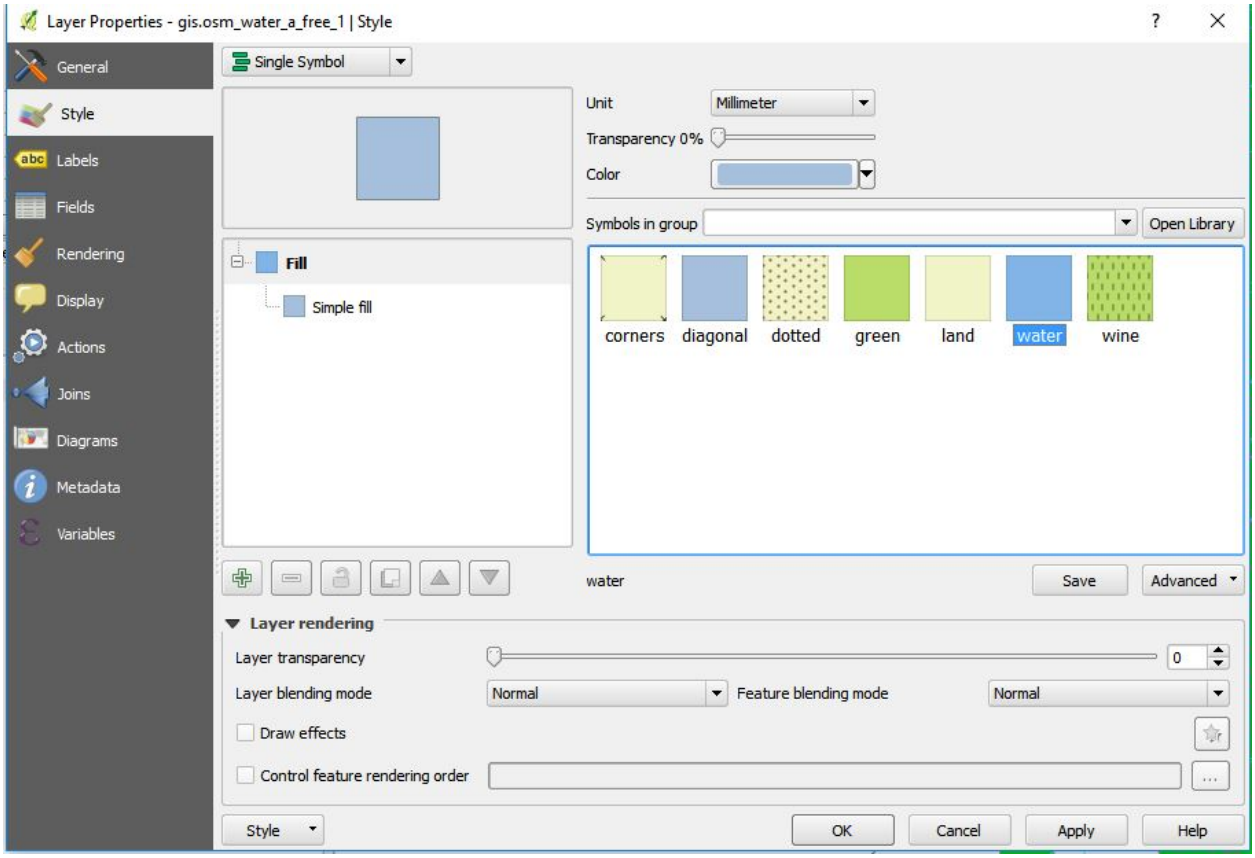
For this exercise, let's visit London and see what we can find out about the city's various wards. Download the main map file from here: http://bit.ly/2pPsCy8

Unzip the file, and then in QGIS, find the button near the top left that says "Add Vector Layer" when you hover over it. This will prompt you to find the map you just downloaded and unzipped. Within the unzipped directory, you want to find the "London-wards-2014_ESRI" directory, and then the file called "London_Ward_CityMerged.shp". Note how a map file is made up of many components, but most of the time, it's the .shp file that you're looking for.

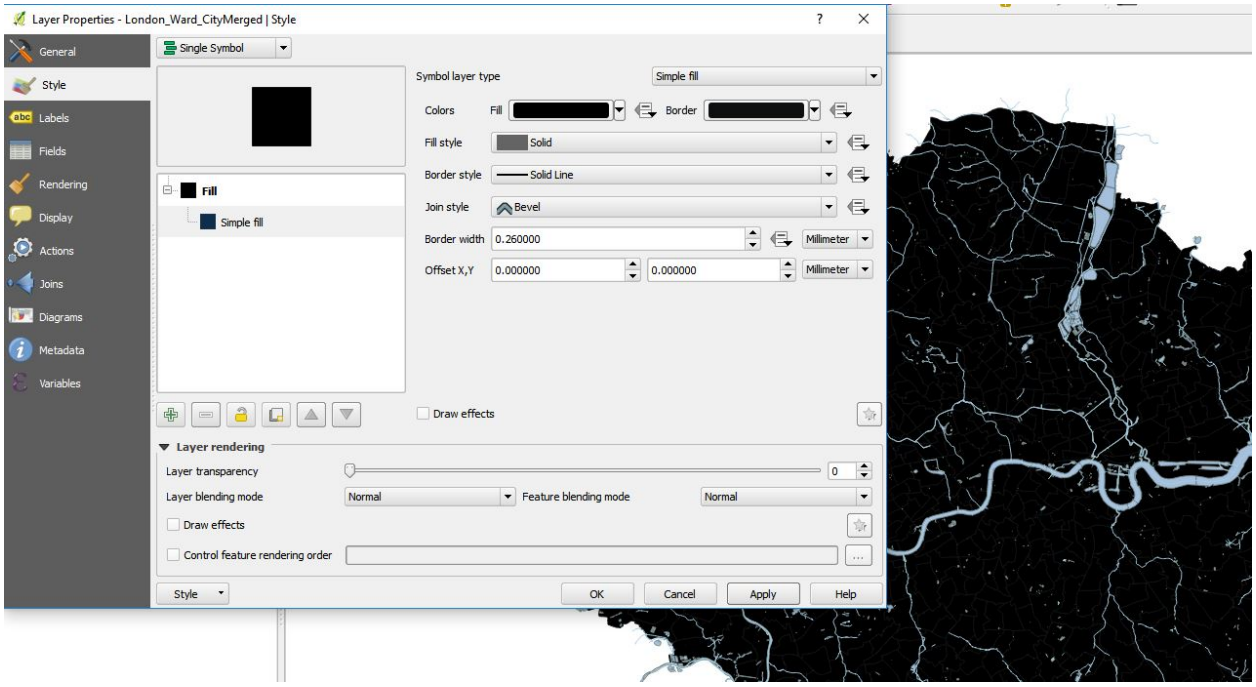


We can get some more decorative map components from the free Open Street Map project. The entire set of free files is here: http://bit.ly/2pvCmQa, but for now, I've pulled out one component that we'll use, the waterways, which you can grab from here: http://bit.ly/2pgrRQx

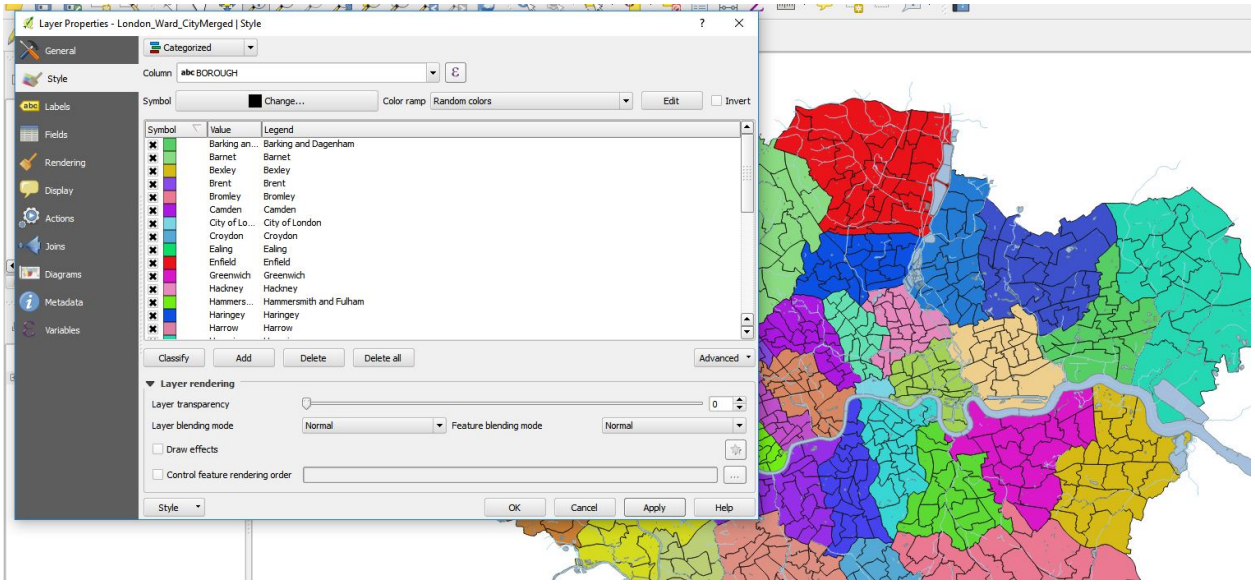
Unzip the waterway file and add both shape files -- one is for major waterways, one includes the more minor streams. We can make these map layers look like water by changing the color. Double-click on the layer name, and under the style menu, choose the color for water:



Let's also for now, to show the contrast, change the color of our Wards to black. Because the border of the wards is also black, this will really create a contrast between land and water:



Can we do anything more useful with the style menu? Double click on wards again and change the drop-down menu to categorized, pick the borough column and click the “Classify” button, then hit apply to preview the changes and ok to finish:



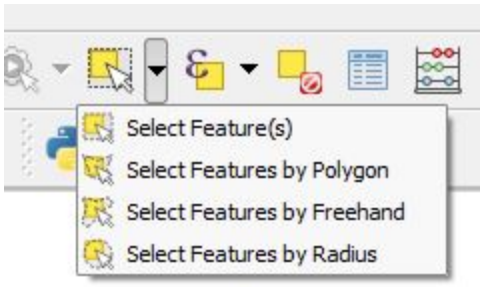
A quick overview of some tools we can use to examine the map:

The Info Button:

The Zoom Buttons: 

The Pan Button:  :

Zoom to Full Extent: 



The Select Buttons:
Selecting data is an important part of GIS analysis because you often want to subset the data you are working with.

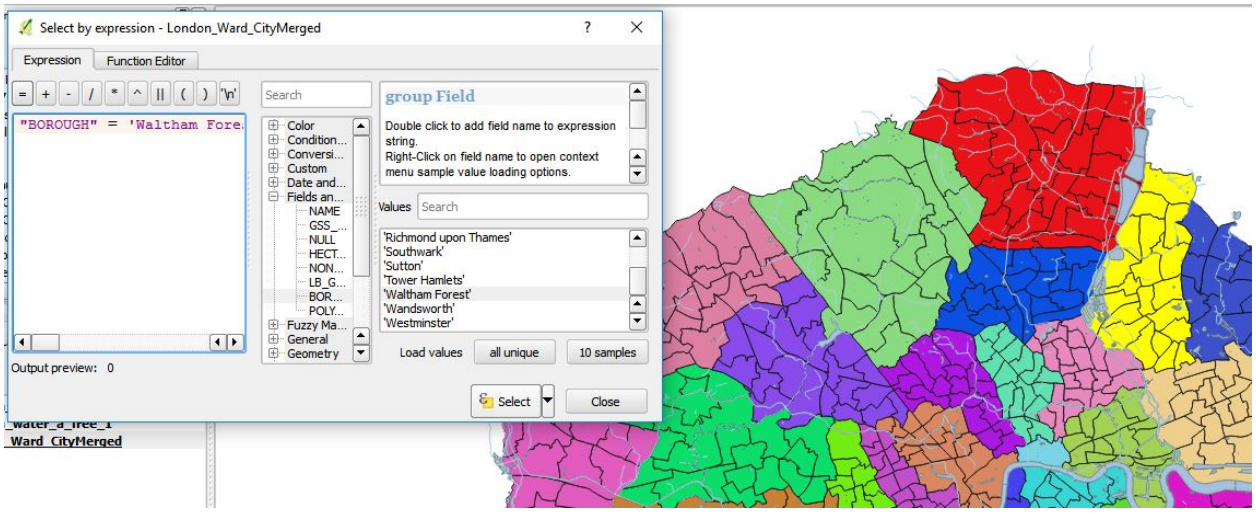
Let's take these one at a time.

Select by Polygon allows you to draw a polygon on the map and right click to select all wards that touch the shape you drew.

Select by Freehand allows you to do the same but without the right click.

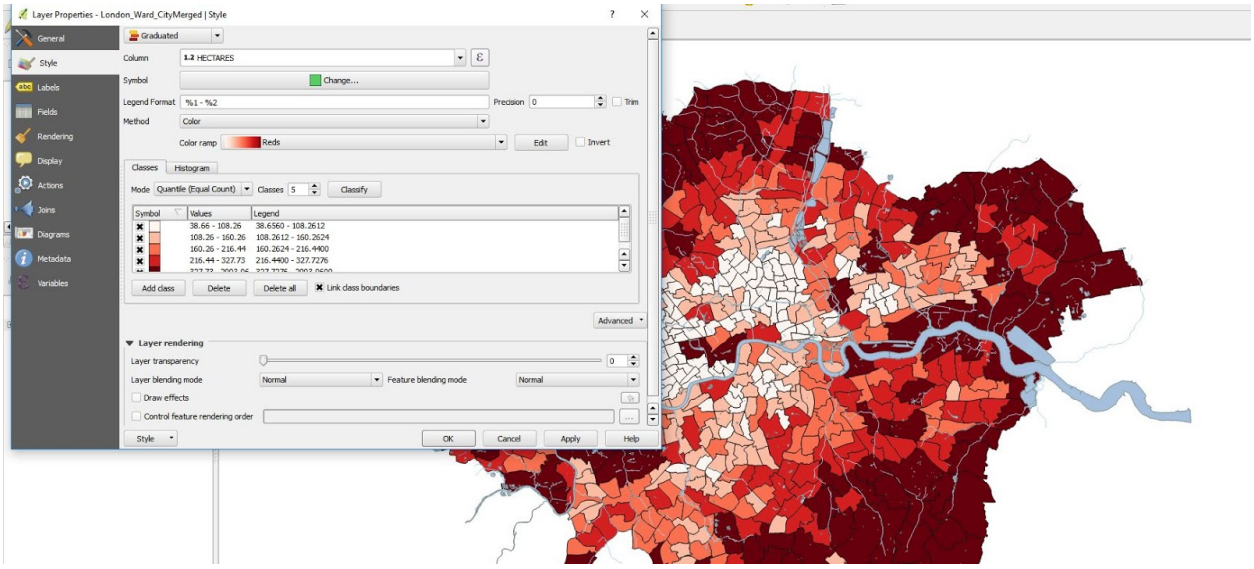
Select by Radius allows you to draw a circle and select all Wards within the circle.

These are mildly useful, but the more useful button is in the middle, Select by Expression:



To clear the selection, click the deselect button  from the top of the screen:

How else can we style this map? Let's change the style from Categorized to Graduated, and pick Hectares, and for mode, pick Quantile. To keep our contrast going, change the color ramp to Reds, and then hit apply:



What happened here? QGIS examined the hectares column and created five equally sized groups. It figured out that to be in the bottom fifth, a ward would have to be between 38.66 and 108.26 hectares, up until the largest group, for which a ward would have to be greater than 327.73.

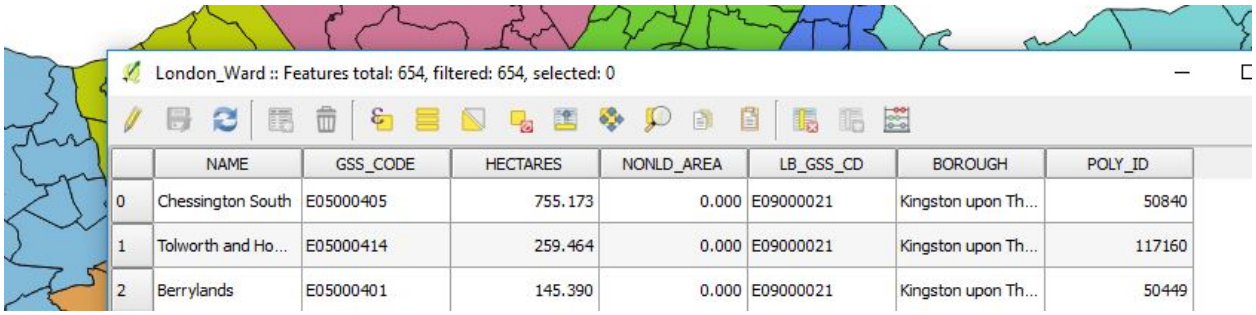
Pause here for a second and explore the other classification systems built into QGIS. For example, what if you change the number of classes? Or use "Pretty Breaks".

What's important to note here is that the decision you make here is a form of journalistic decision-making that is akin to many other reporting tasks: Who do you interview for a story? What questions do you ask? What quotes do you use in the story? In other words, when you create a color-coded "choropleth" map, you need to choose your classification rules wisely and fairly based on the underlying data.

Let's put this thought to use by bringing in some more meaningful data, The London Data Store has some rich data summarized at the ward level. Let's take a look at a file they provide called "ward profiles" -- I've created a local copy for our class here. Let's open this up and see what's inside:

New Code	Names	Borough	Hectares	Square Kilometres	Hectares	Square Kilometres	2001 (Census)	2006	2011 (Census)	2013	2018	2023
E05000026	Abbey	Barking an	128	1.3	126	1.3	8238	8150	10268	10989	12337	13810
E05000027	Alibon	Barking an	136	1.4	136	1.4	6853	6769	7659	7715	7732	7732
E05000028	Becontree	Barking an	129	1.3	129	1.3	8967	8725	9044	9436	10481	10946
E05000029	Chadwell H	Barking an	338	3.4	338	3.4	2778	2774	2984	3032	3162	3236
E05000030	Eastbrook	Barking an	345	3.5	345	3.5	2956	2922	3072	3101	3115	3144
E05000031	Eastbury	Barking an	144	1.4	144	1.4	7087	7262	8129	8210	8268	8338
E05000032	Gascoigne	Barking an	114	1.1	109	1.1	8718	9861	11517	13007	15739	17849
E05000033	Goresbroo	Barking an	128	1.3	128	1.3	8139	7894	8882	8928	8961	8961
E05000034	Heath	Barking an	200	2.0	200	2.0	4872	4956	5422	5498	5596	5646
E05000035	Longbridge	Barking an	165	1.6	165	1.6	5440	5801	7020	7156	7233	7233
E05000036	Mayesbroc	Barking an	186	1.9	186	1.9	5064	5021	5602	5641	5683	5710
E05000037	Parsloes	Barking an	124	1.2	124	1.2	7316	7331	7959	8039	8080	8080
E05000038	River	Barking an	354	3.5	313	3.1	3308	3195	3515	3675	4570	5816
E05000039	Thames	Barking an	789	7.9	669	6.7	1294	1477	1615	2229	3320	4269
E05000040	Valence	Barking an	129	1.3	129	1.3	6845	6845	7696	7805	8044	8160
E05000041	Village	Barking an	215	2.1	215	2.1	4563	4440	5052	5098	5169	5239
E05000042	Whalebon	Barking an	155	1.6	155	1.6	6242	6134	6853	7024	7400	7658
E05000043	Brunswick	Barnet	318	3.2	318	3.2	4677	4664	5181	5358	5606	5748
E05000044	Burnt Oak	Barnet	207	2.1	207	2.1	7498	7978	8853	9164	9530	9651
E05000045	Childs Hill	Barnet	309	3.1	309	3.1	5690	5894	6499	6674	6951	7355
E05000046	Colindale	Barnet	264	2.6	264	2.6	5348	5590	6505	8319	11437	14244
E05000047	Coppetts	Barnet	272	2.7	272	2.7	5440	5707	6359	6507	6598	6543
E05000048	East Barn	Barnet	375	3.7	375	3.7	4161	4161	4321	4444	4627	4747
E05000049	East Finch	Barnet	251	2.5	251	2.5	5897	5984	6395	6564	6694	6634

What we're going to do is "join" this data to the data in our map. And just like any database join, we need to have something in common between the two datasets. The column labeled "new code" looks promising. Let's see if we have that in our map. Click the "Open Attribute Table" button in QGIS 



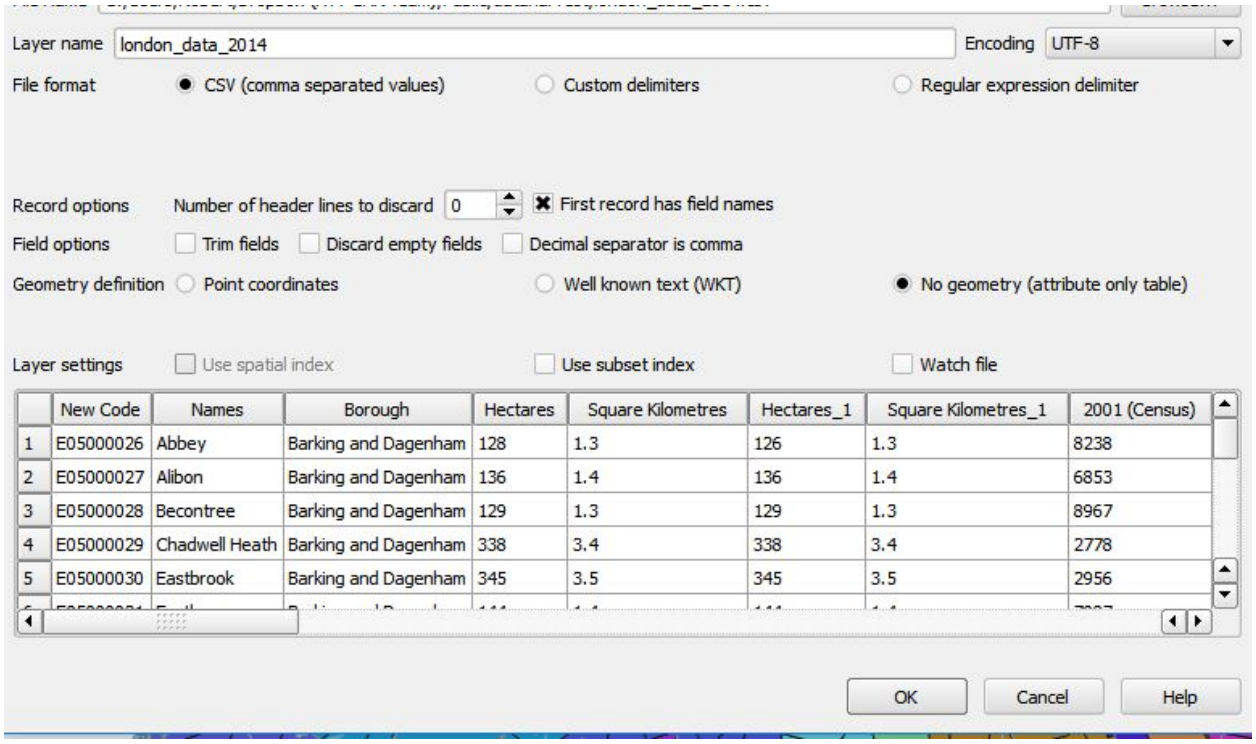
Voila -- it looks like we can join the spreadsheet's "new code" to the map's "GSS_CODE".

Before we can do that, there's one other step. We need to save the spreadsheet as a csv file, a generic data storage format that QGIS can read. We also need to clean the file up and remove

any extraneous text -- as we'll see in a moment, sometimes QGIS has trouble determining whether a CSV column is text or numeric, and in this case, we want our data to be numeric. In Excel, scroll to the bottom of the file to make sure there is no extraneous text. (If there is delete it -- we only want rows and columns of data with one row for headers.)

Then click on File, Save As, and choose csv as the format. (**NOTE**: On a Mac, you must choose Windows CSV as the format because Excel creates CSVs with different line endings on the two systems.) Excel will prompt you a couple of times- click OK, then Yes, then close the file in Excel, and click "Don't Save" -- we've saved it the way we want to already.

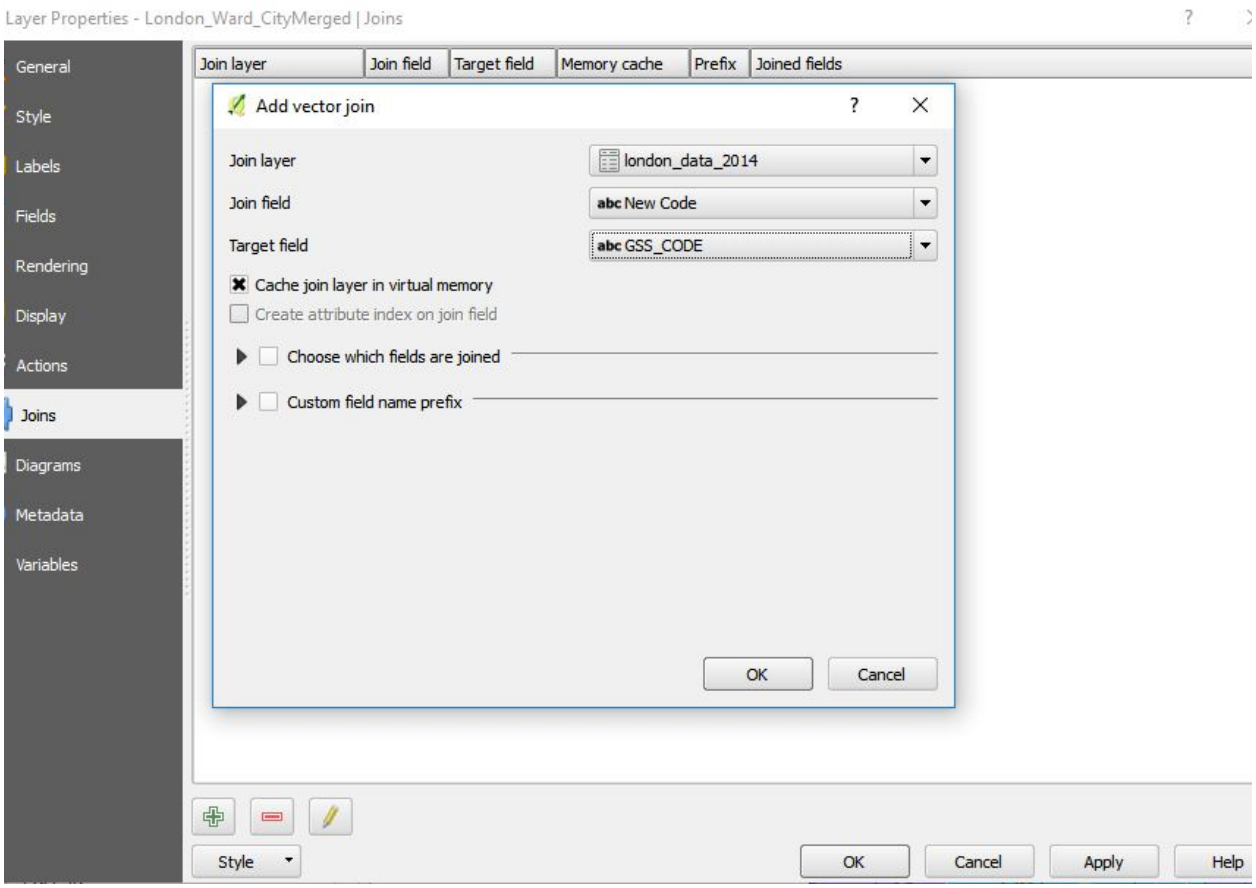
Now, in QGIS, to add the data file, click on the add CSV button: 



	New Code	Names	Borough	Hectares	Square Kilometres	Hectares_1	Square Kilometres_1	2001 (Census)
1	E05000026	Abbey	Barking and Dagenham	128	1.3	126	1.3	8238
2	E05000027	Alibon	Barking and Dagenham	136	1.4	136	1.4	6853
3	E05000028	Becontree	Barking and Dagenham	129	1.3	129	1.3	8967
4	E05000029	Chadwell Heath	Barking and Dagenham	338	3.4	338	3.4	2778
5	E05000030	Eastbrook	Barking and Dagenham	345	3.5	345	3.5	2956

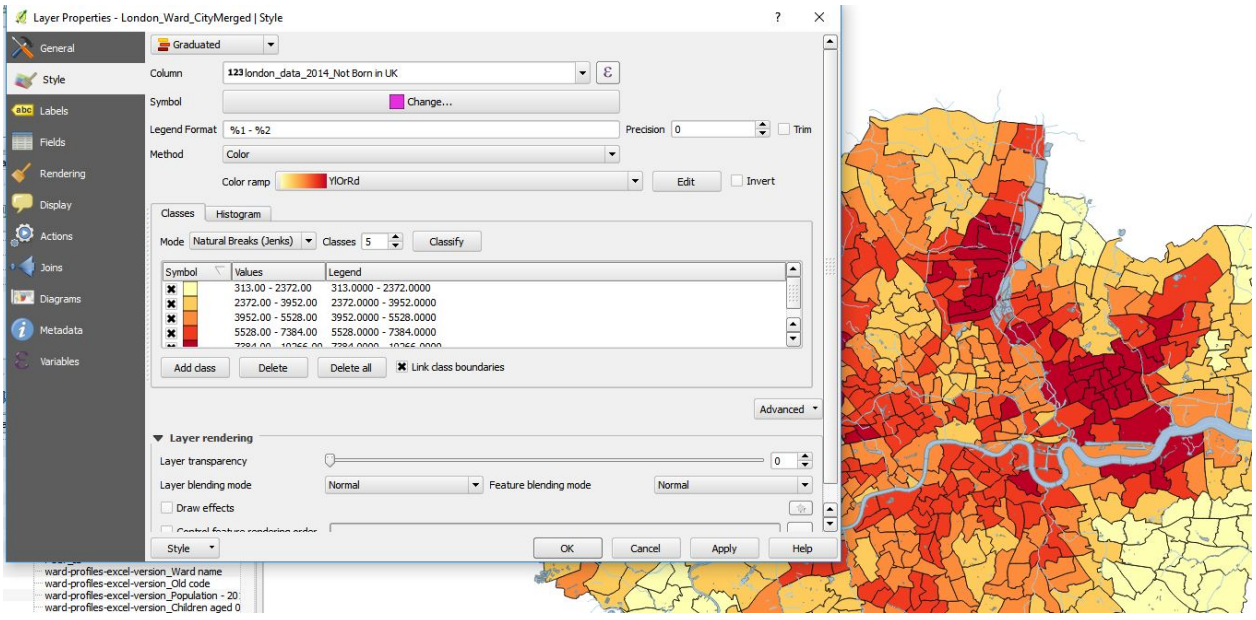
Make sure "First record has field names" is checked, and under geometry, select "no geometry", and then click ok. The data file is added to our list of layers.

Double-click the "London_ward" listing to get back to its attributes, and find the "Joins" menu and hit the plus sign to set up a new join:



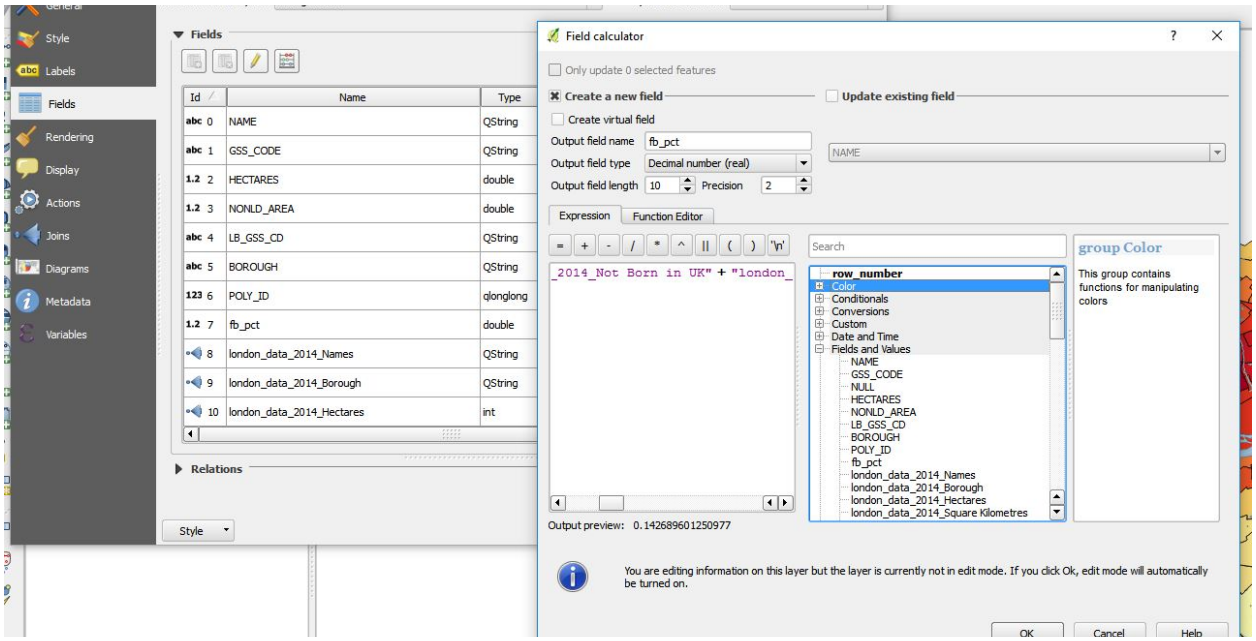
Pick the ward-profiles as the join layer, and tell QGIS you want to join “New code” to the target “GSS_Code.” Then click Ok.

Now if we go back to style the Wards layer, all of the numeric fields from london_data_2014 will be available to us. Let’s pick “Not Born in the UK” as our column, yellow-orange-red as our color ramp, and natural breaks for our buckets:



So now we have something interesting -- a map that shows us the foreign-born population of the UK. But can anybody think of a way to make this map more accurate and meaningful?

Yes, by making our map based on the raw population, we're in danger of simply showing population density. What if there are more foreign-born people in some of the wards simply because there are more people overall in the wards? What we really need to do is calculate the PERCENTAGE of the population that is foreign-born, and for this, we can use the QGIS field calculator.

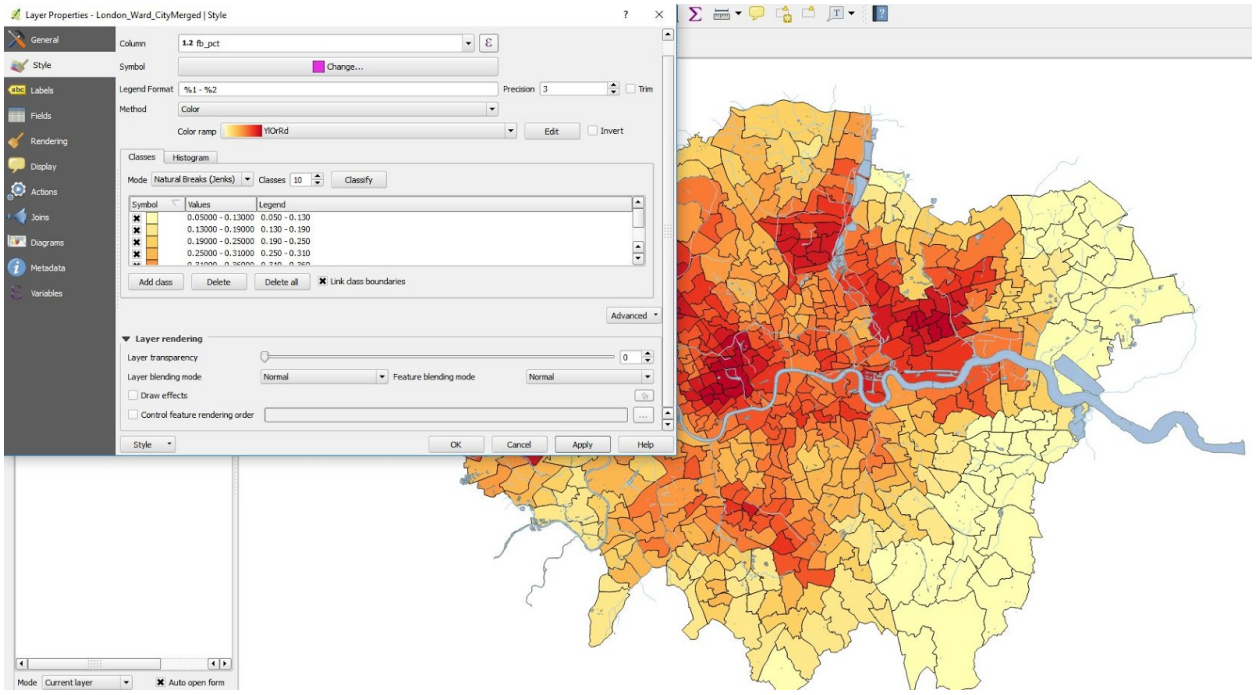


Under the London Ward fields menu, click on the abacus icon to create a new field. Let's call the new field fb_pct and because we're going to be calculating a percentage, let's give it a field type of decimal, with a precision of 2 (if we don't specify this, QGIS will round it up to a whole number, which we do not want).

We''ll use the QGIS GUI menu to create our formula:

"london_data_2014_Not Born in UK" /("london_data_2014_Not Born in UK" + "london_data_2014_Born in UK")

We hit ok, then click on the pencil to get out of editing mode and save our changes. We can now make a map based on our new field, fb_pct:



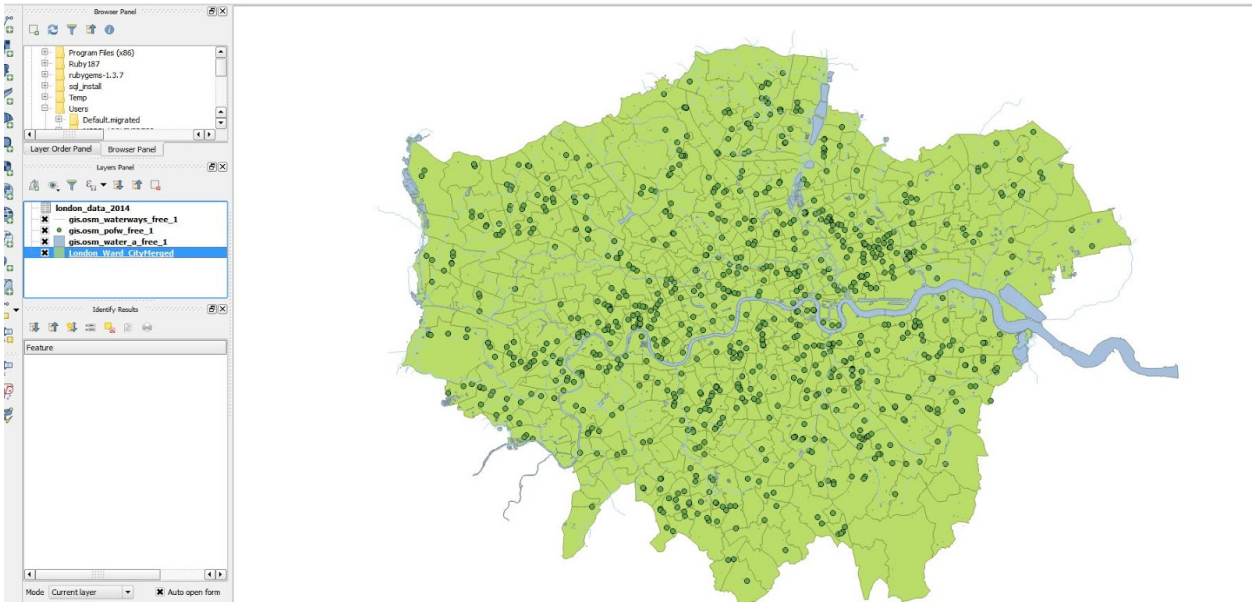
This brings us to the end of Part I of our lesson. If there is still more time, think of other variables that would make for a good map and have a go at it.

PART II -- THE ART OF THE SPATIAL JOIN

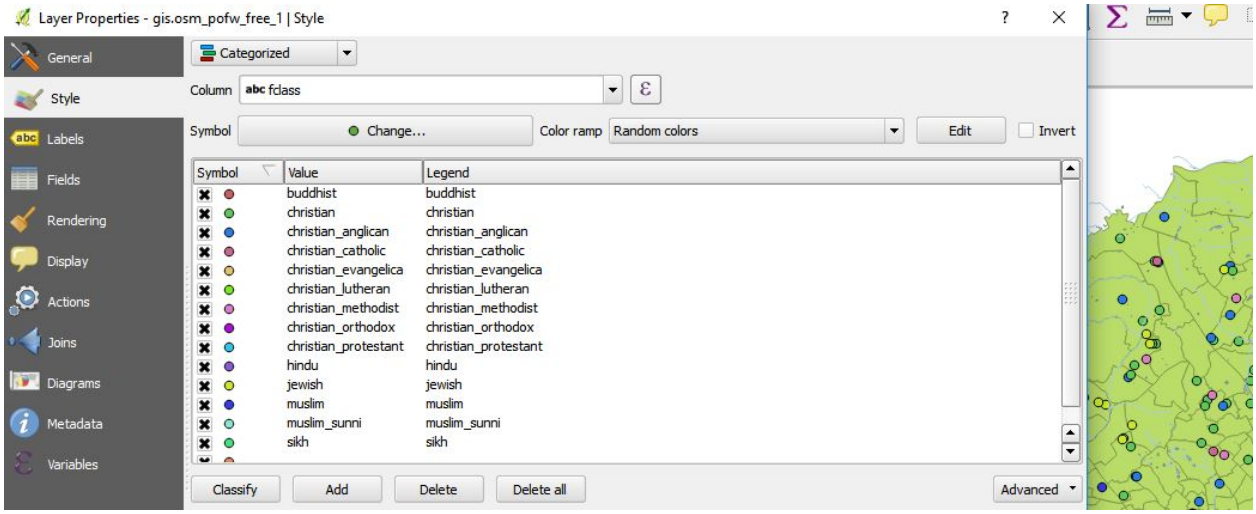
So far, we've explored the basics of mapping with QGIS and worked with polygon layers. We've joined summary data about London wards to a map of London wards and created a "choropleth" style map to show some patterns.

In this lesson, we work work with "point" data -- dots on the map representing a specific place or an incident that occurred at the place. We will show you how to find patterns in your points, how to convert raw GIS coordinates into QGIS points, and most interesting of all, show you how you can marry points to polygons via a "spatial join" and take your analysis to a new level.

So to review what we have so far, you should have a QGIS project containing a map of wards in the greater London area. To this let's add a layer containing places of worship in the Greater London Area, which I obtained from Open Streetmap but have posted here: http://bit.ly/2qjsYgz Once you download and unzip the file, you can add the places of worship as a layer just like you would a polygon map:

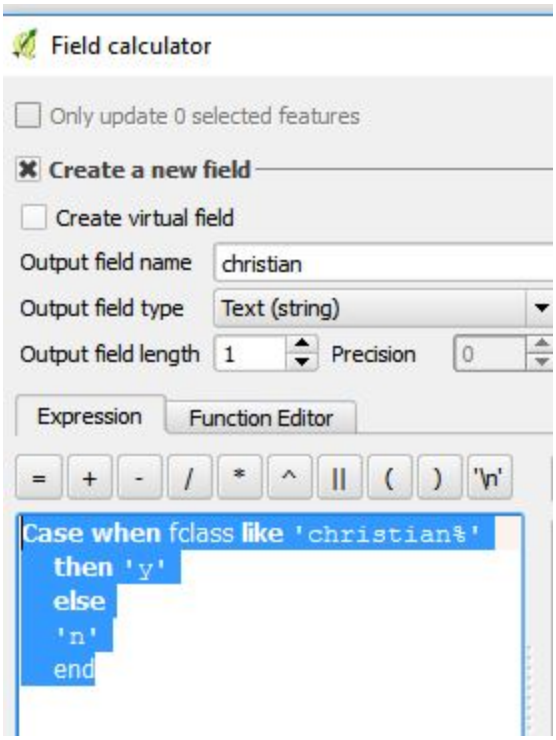


And just like a polygon map, there are certain simple analytical looks you can create for the data. For example, let's color the points by category:



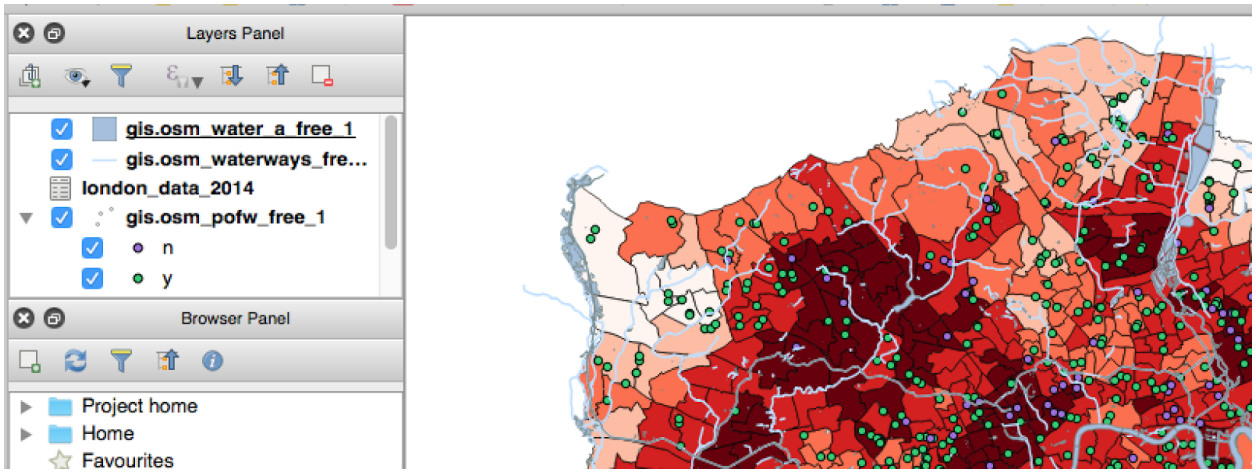
There are a lot of denominations. Perhaps we want to collapse the categories into two groups -- Christian denominations and non-Christian. We can do that with the field calculator. Click on Fields, then the abacus, and then enter a new field name called christian, we'll make it text with a length of 1 and create it with a classic SQL Case..when statement:

Case when fclass like 'christian%'
then 'y'
else
'n'
end



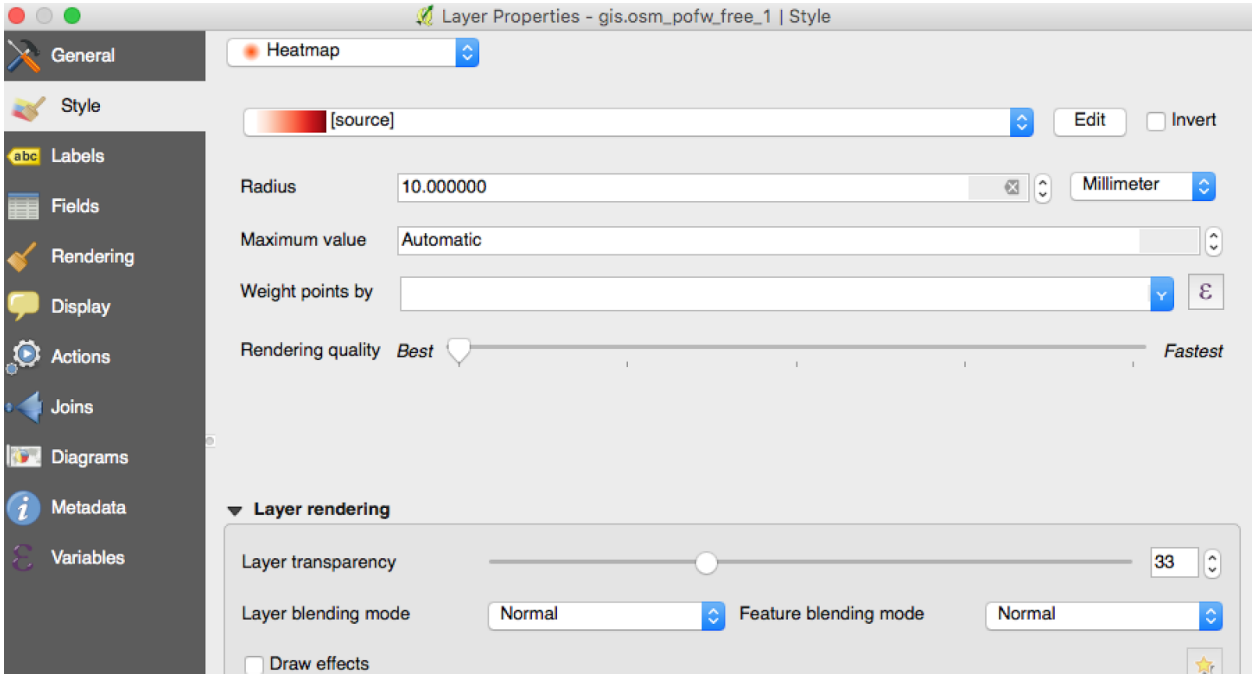
Once we execute it , we'll get a new field in our point map file that we can use to classify the points by two groups:

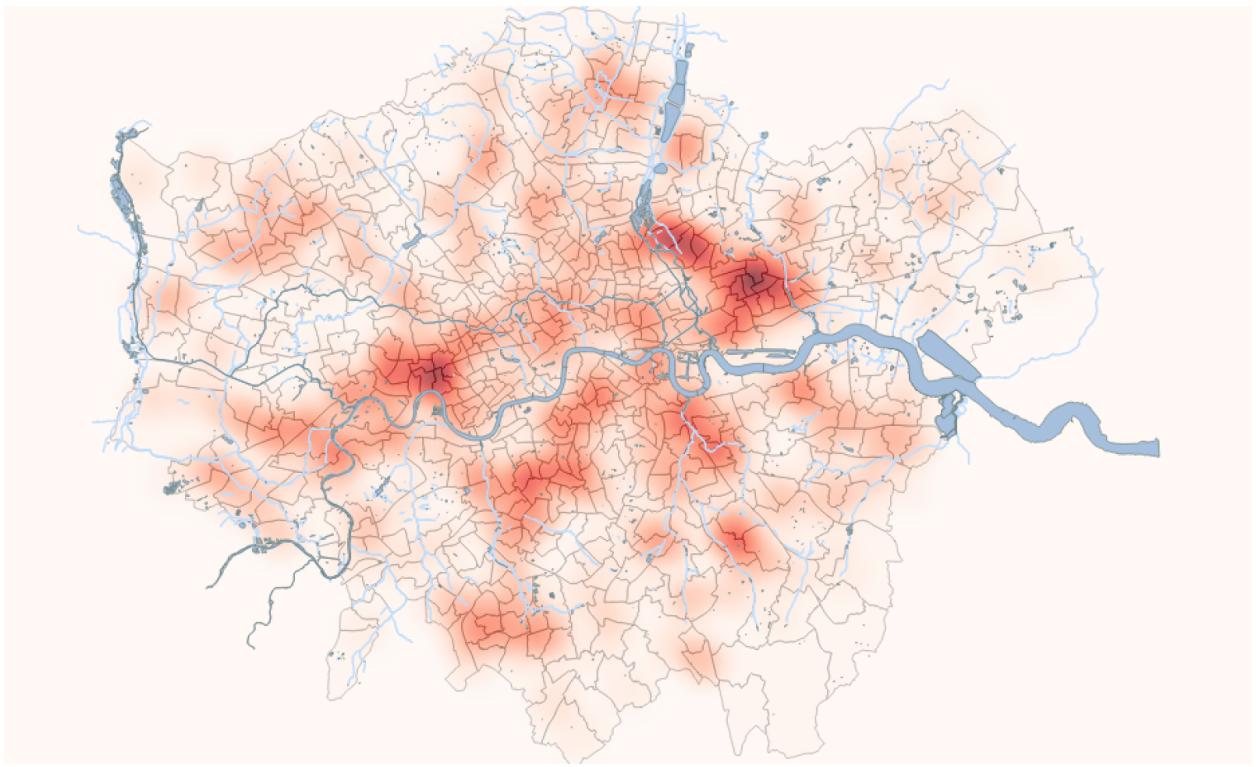
3100	christian	*NULL*	y
3101	christian_anglican	St Andrew's Chur...	y
3300	muslim	Wimbledon Mosque	n
3102	christian_catholic	St. Gregory's	y
3101	christian_anglican	St Margaret's Ch...	y



Another popular visualization that can be done with point data is a heat map. We can convert the points to visual “blobs” representing the density of the points -- in this case, the visualization would show where churches are most concentrated.

To make the heat map pop, let’s first change the ward map to a single symbol, preferably a light color. Then on the place of worship theme, choose Heatmap, pick the red color ramp, and set the layer transparency to 33.





Very cool to look at. You can play around with the parameters and see how your visualization changes. But there are two problems with heat maps: 1) What happens at the edges of our map? Because we are missing data from churches outside of greater London, we're underestimating the edges and 2) How easy would this be to explain to our readers?

Another approach is to do a “spatial join” -- to assign each point on our map to the polygon it sits inside of. This approach will yield for us a count of houses of worship for every ward in greater london.

But first… We’ve up until this time ignored an important aspect of GIS work. As we know, at least for now, Earth is round. But our computer screens are flat. For simply viewing maps, this is an aesthetic issue -- does our map look like maps we’re familiar with or does it appear distorted? For doing spatial calculations, however, accounting for the bend of the earth becomes more crucial.

The map’s we’re using come to us with a “projection” built in. A projection is a set of parameters that estimate the curve of the earth, and there are literally hundreds of projections to choose from. Some projections are better to use if you’re showing a nation or continent, others are more accurate if you’re showing a local area.

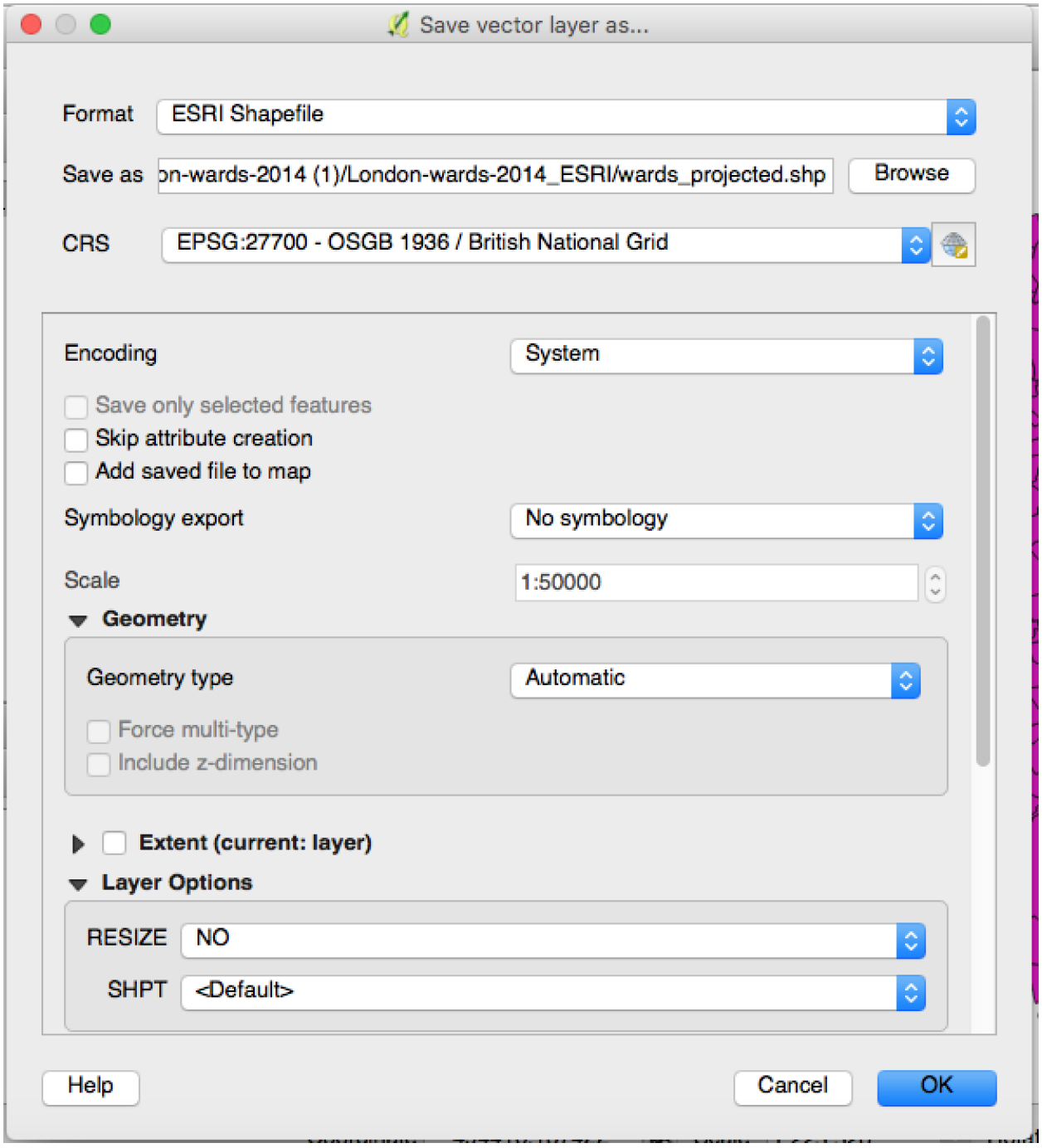
What’s most important here is that we use a projection that is commonly accepted for London, and that all of our map layers are using the same projection.

Adding to the confusion, by default QGIS “adjusts” maps on the fly so they display properly even if they are in different projections. Again, that’s fine for making visuals, but doesn’t help us with spatial calculations.

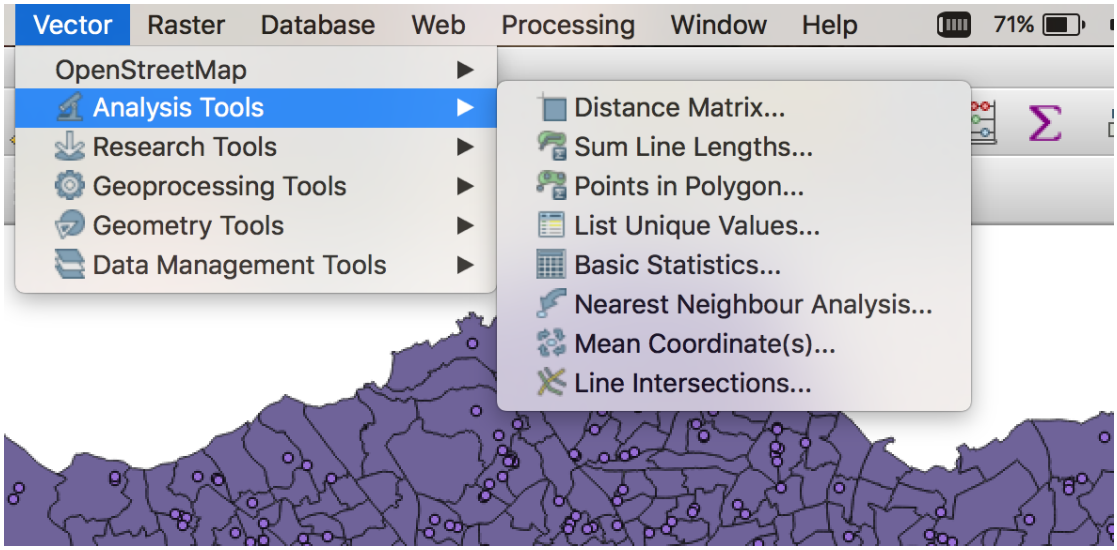
So what I like to do in a situation where I’m dealing with projection changes is to do it in steps. I like to close the project and do the reprojections as separate tasks, and then create a new project with the projected layers.

So let’s get started. A bit of Googling reveals that a commonly used projection for the UK is called The British National Grid, so we’ll go with that.

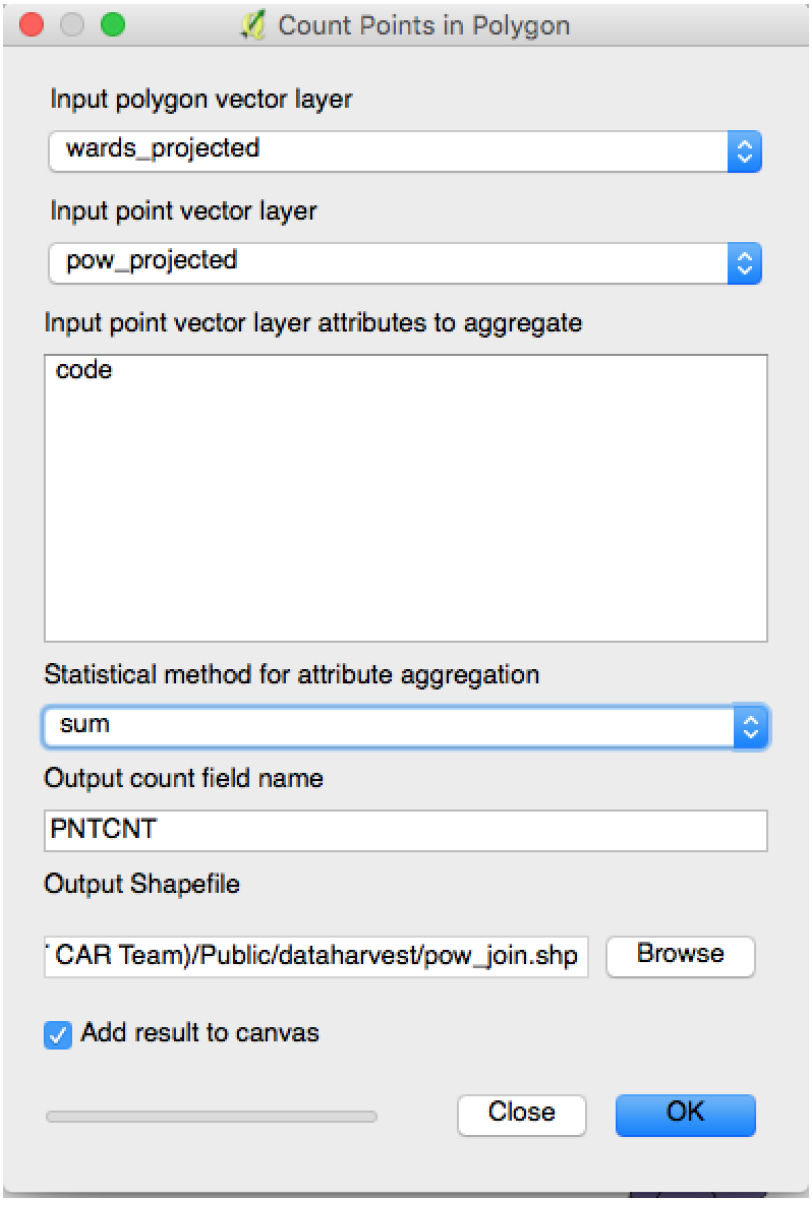
Let’s save our current project and create a new one and bring the wards layer in again. We will immediately right clik on the wards and do a Save As and assign this layer to our new projection and give it the new name “wards_projected.”

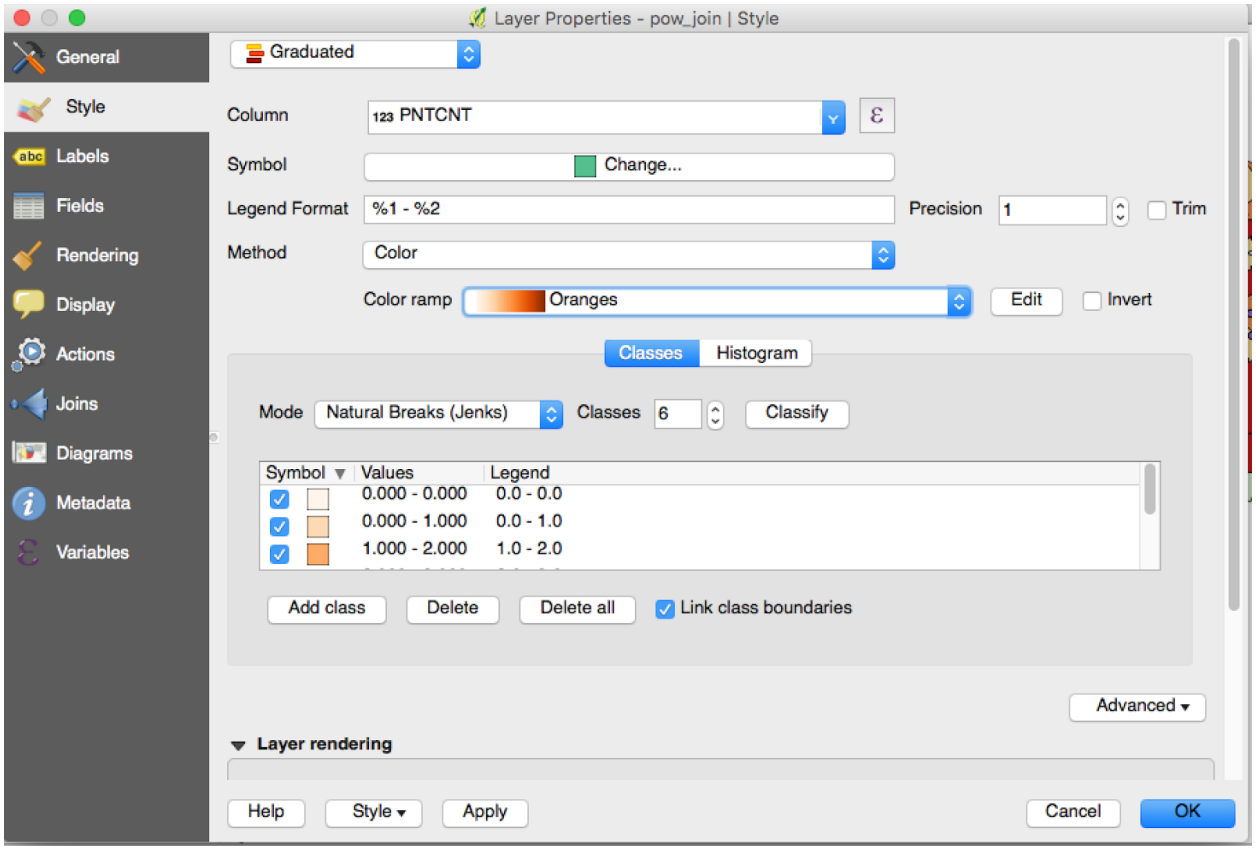


Let's do the same with our places of worship theme and wave that out as "pow_projected." Now let's create a new map project, add both themes, and conduct our spatial join. QGIS comes with tons of tools for doing spatial calculations, the one we want is under Vector | Analysis Tools | Points in Polygon.

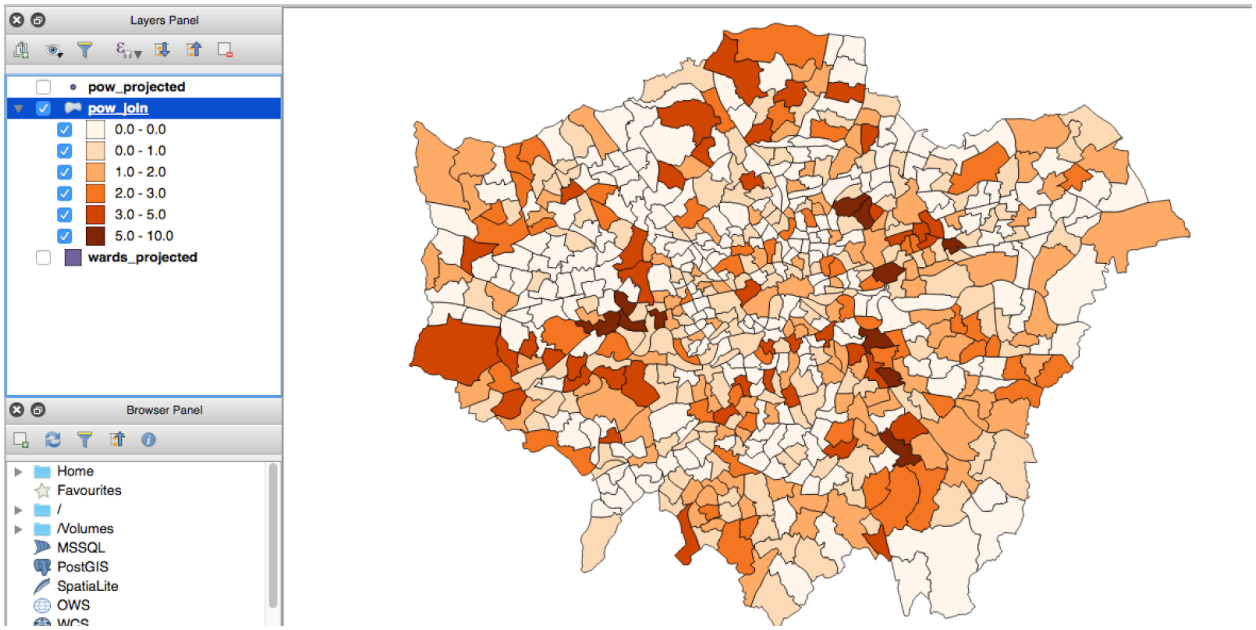


From here a menu pops us that allows us to set the parameters for our spatial join:

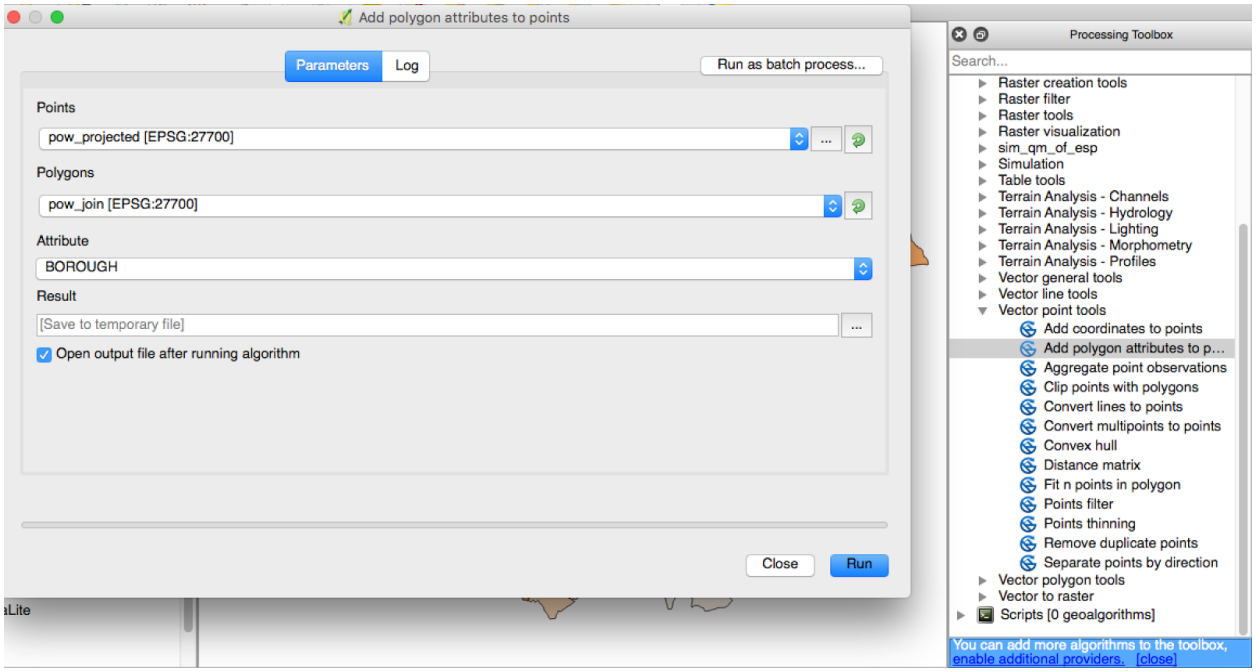




Now we can make a choropleth based on the values…



Conversely, we could have added the polygon's attributes to the points. Let's say you wanted to quickly make a list of places of worship in Bexley. To do this, you'd look under Processing | Toolbox | Vector Point Tools:

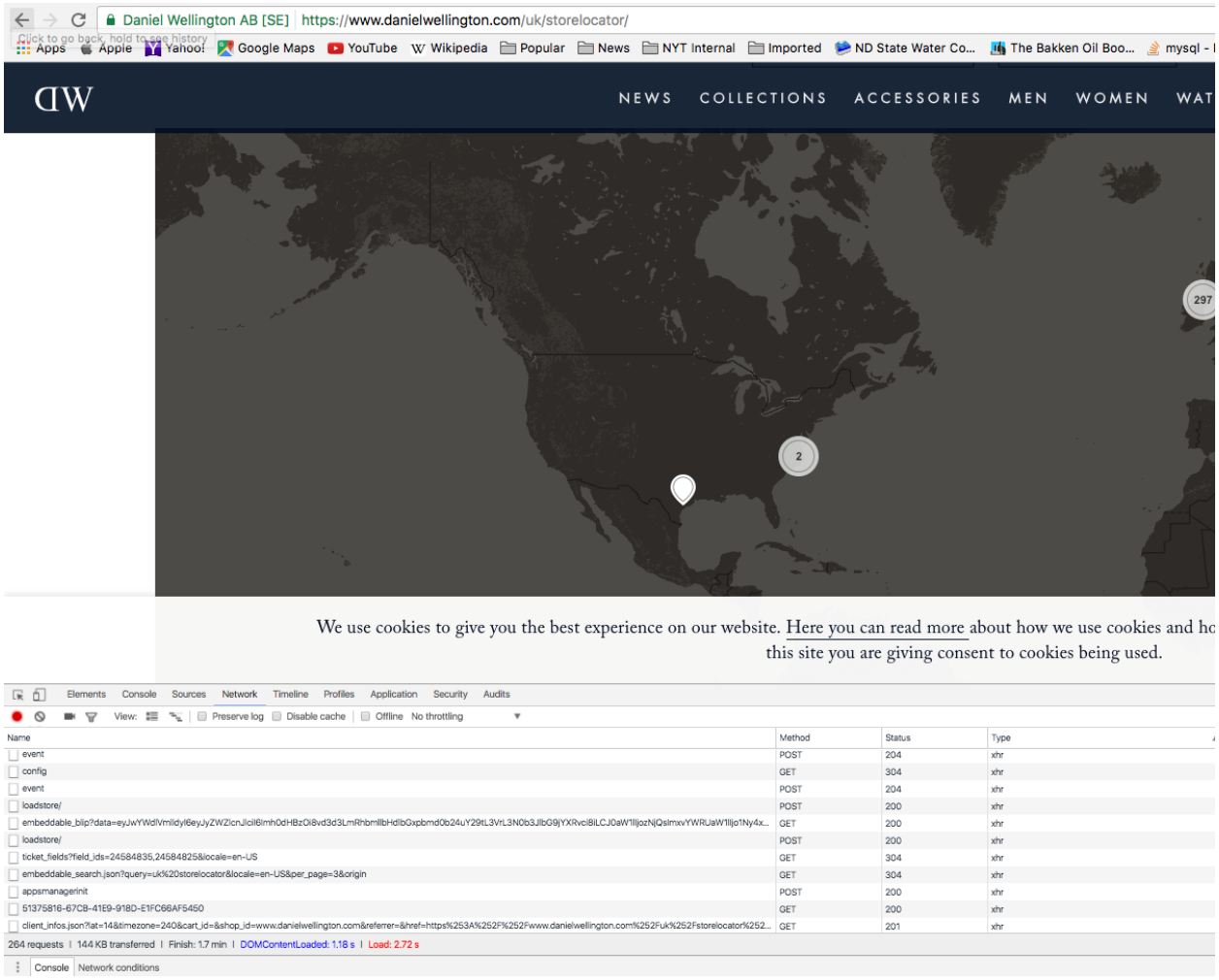


Let's close with one more example that will combine a number of skills and hopefully bring this all home.

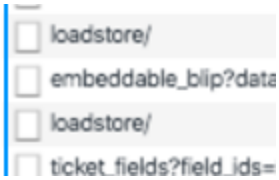
Often, we run into data on the Web that we can view, but that we wish to analyze further. Sometimes the data provider will give us a download link, but sometimes, the data is on the Web in a format in a display only format.

Let's say we were working on a story about where retailers choose to locate stores. Many retailers have "find my store" tools online, but don't provide a complete list of all locations. However, if you look under the hood, you'll learn that the underlying data to many Web sites is gettable if you know where to look.

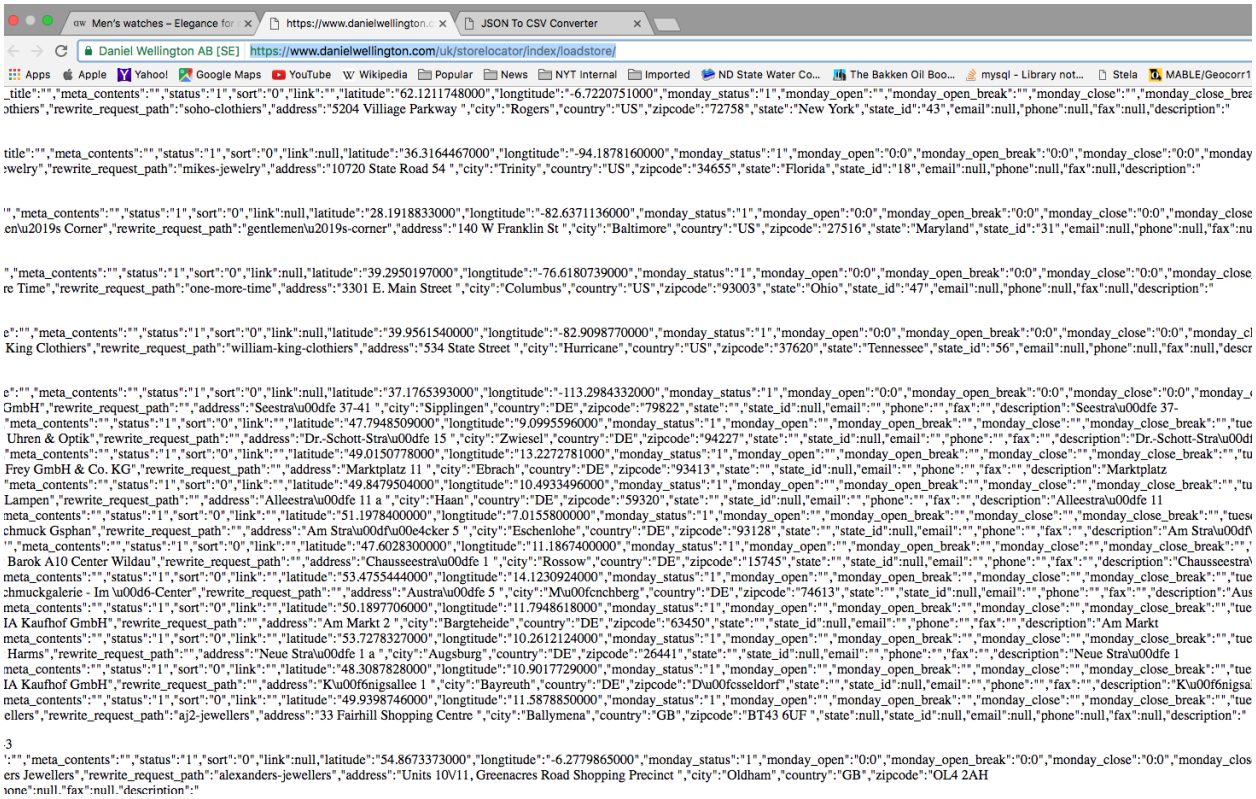
I don't wear a watch, but just for example's sake, let's look at the Daniel Wellington chain of watch dealers.



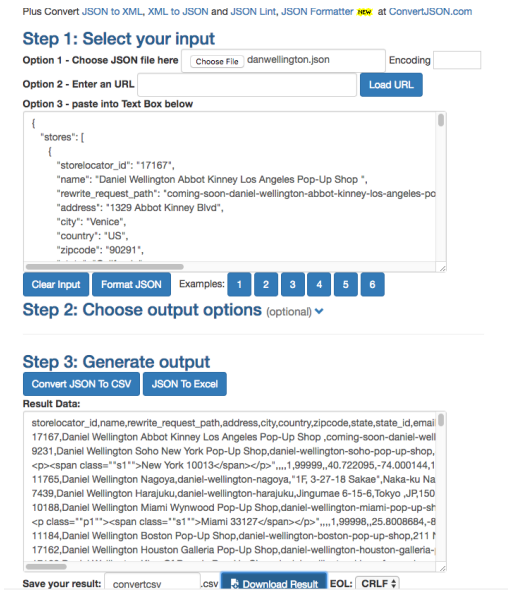
If you go to the store locator page, and look at developer tools/network resources in your browser, you'll find that the site grabs data from a hidden page called "loadstore/"



If you open up that link you'll see it leads to data about all stores in .json format, a commot format for supplying data to a javascript Web site.

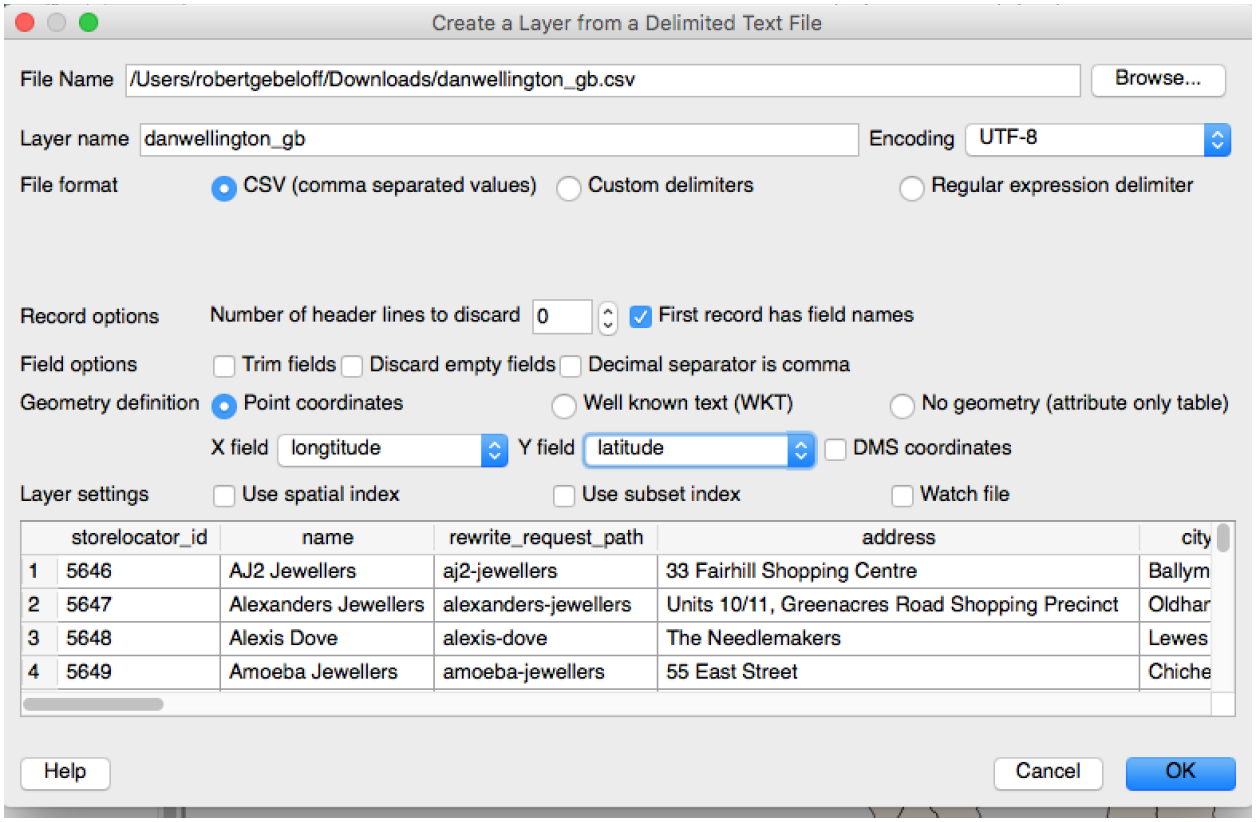


The data includes the latitude and longitude for every store, which we'll need to add the data onto our map. First we need to convert the json format to csv, which we can do with one of the many free online converters online. I this case, I've saved the data into a file called danwellington.json and uploaded it to convertcsv.com.



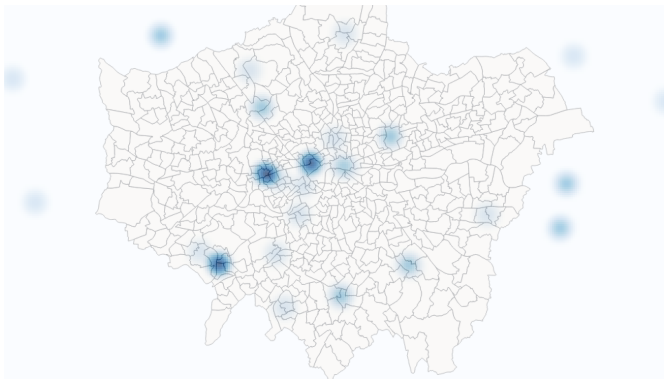
The site parses the json and converts it to csv, which I can then download onto my hard drive. I've saved a copy here.

Now let's download add this data to QGIS.

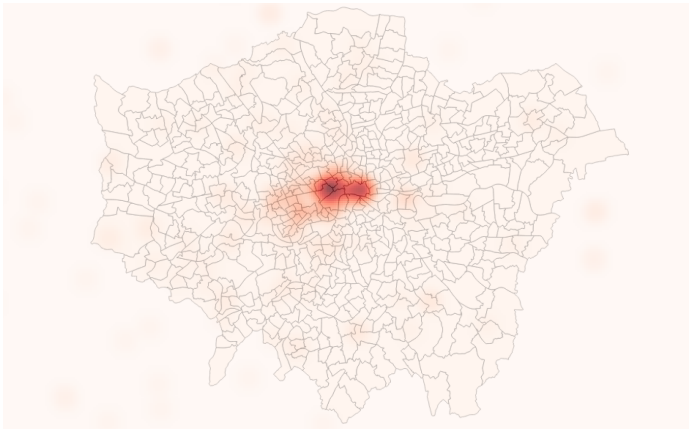


We add a csv file like before, but this time, we tell QGIS we have point coordinates and assign longitude to the X field and latitude to the Y field.

We can then convert our points to a heat map:



Let's add some more data to the map. Several Internet users as a hobby track Starbucks locations. I grabbed a November, 2016 version of the Starbucks file and posted the UK locations as a csv file here.



Let's download it and a second heat map layer. As you can see, a Starbucks is much more concentrated in the center, whereas the watch

retailer is more evenly distributed in shopping centers throughout the region.

Of course, with the skills we've covered in this class, you can do much more. You could do a spatial join with the Wards data, and then do per-capita calculations. You could tabulate the average demographics of a ward with a Starbucks location. More practically, you could find Census data for smaller levels of geography and join that to your points.

The point is, spatially joining data is just as powerful and interesting as joining data in a database manager. Which, if you're a data journalist, is pretty interesting.