Spatial Data Visualisation: Advanced Techniques In QGIS

Session 3

Written and Presented by **Tom Armitage,**

For the **CDCS, University of Edinburgh**

WELCOME TO SESSION 3

In this session we will be looking at the following ideas:

- Viewing Election data:
    - As Cartograms
    - As Dot Density maps
- Animating data that changes over time:
    - Animating Heatmaps
    - Animating Point data sets

ELECTION MAPPING: CARTOGRAMS

Cartograms can be a great way to represent data that has different spatial densities. It is a way of accounting for the effects of small areas having more things and large areas having less. For example, London has a small area relative to Scotland but has almost twice the population. On election maps this means that numerous constituencies in London appear very small and insignificant and the large areas in Scotland appear much larger and therefore seem more important. A cartogram warps the map to account for these differences but still retains some of the original boundary shapes to make the map familiar.

We are now going to make a map of the London EU referendum votes from 2016 and re-proportion the areas using the cartogram plugin based on the number of voters.

Add the **LondonBoroughEURef.gpkg** to the map.

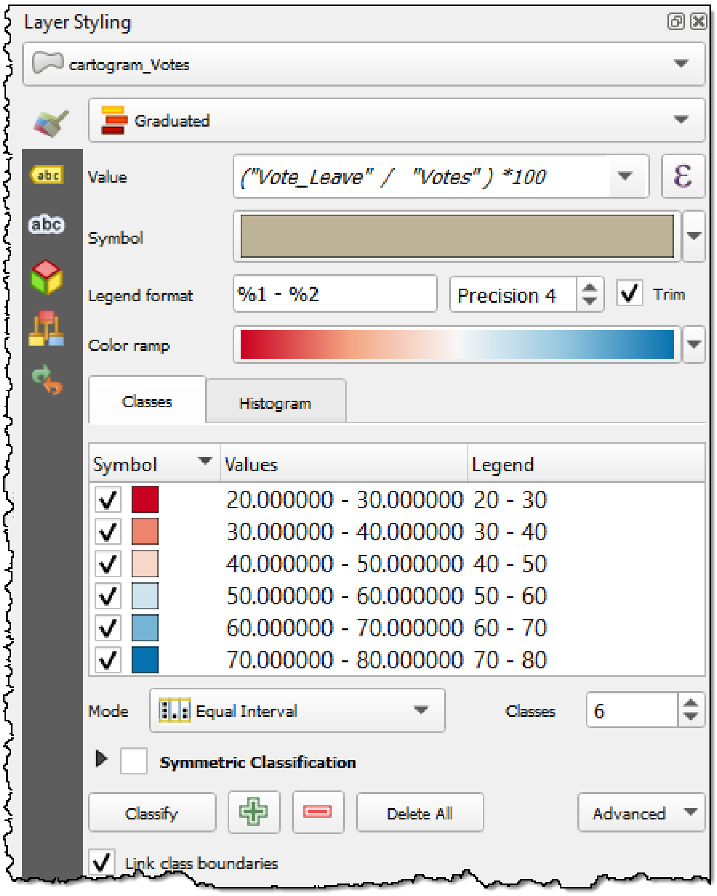
Change the styling to be **Graduated.**

Change the value to be the following expression:

```
("Vote_Leave" / "Votes") *100
```

Select the **RdBu Color ramp** (You may need to go into the **All Color ramps** list from the drop-down arrow.)

Make **6 equal interval** classes and edit them (click on each row) to make a fair representation:



Layer Styling

cartogram_Votes

Graduated

Value: ("Vote_Leave" / "Votes") *100

Legend format: %1 - %2 Precision 4 ☑ Trim

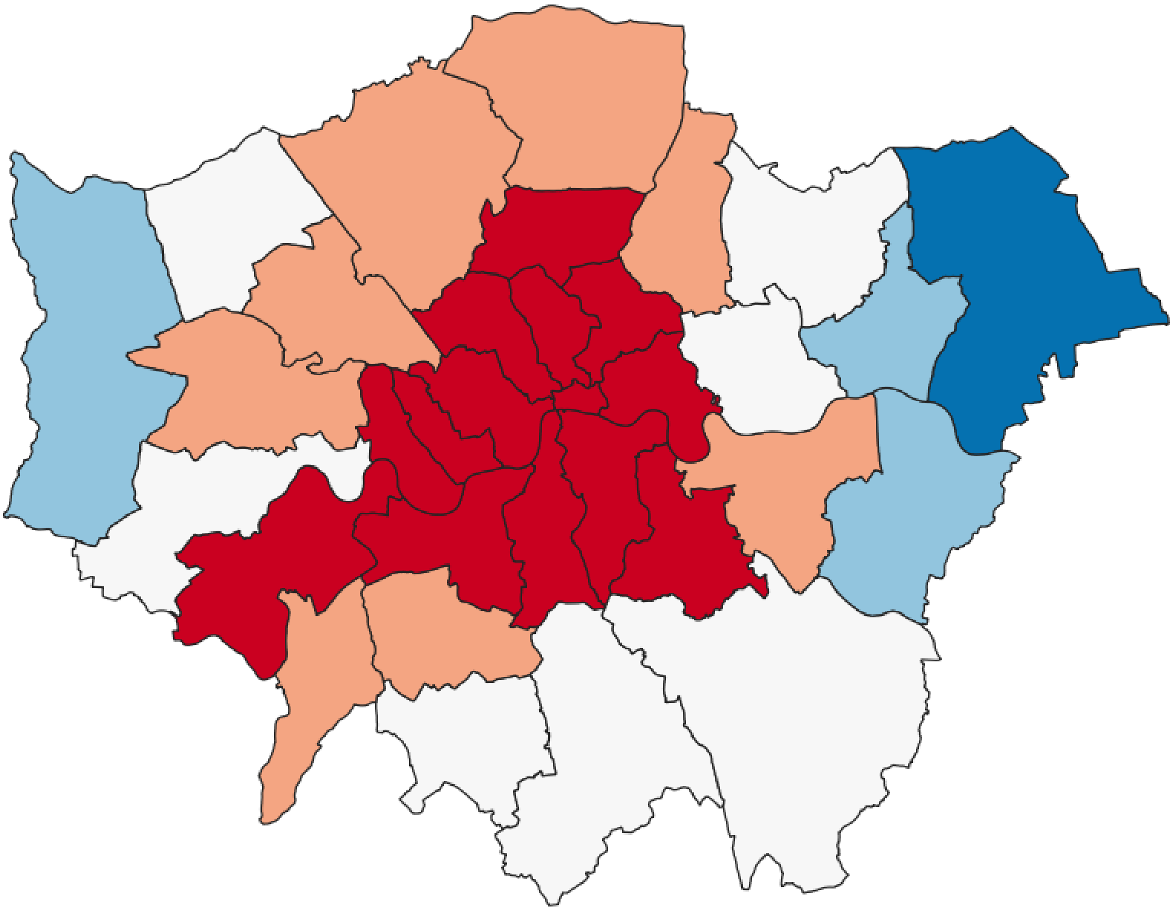
Symbol	Values	Legend
☑ 🟥	20.000000 - 30.000000	20 - 30
☑ 🟥	30.000000 - 40.000000	30 - 40
☑ 🟧	40.000000 - 50.000000	40 - 50
☑ 🟦	50.000000 - 60.000000	50 - 60
☑ 🟦	60.000000 - 70.000000	60 - 70
☑ 🟦	70.000000 - 80.000000	70 - 80

Mode: Equal Interval Classes: 6

□ Symmetric Classification

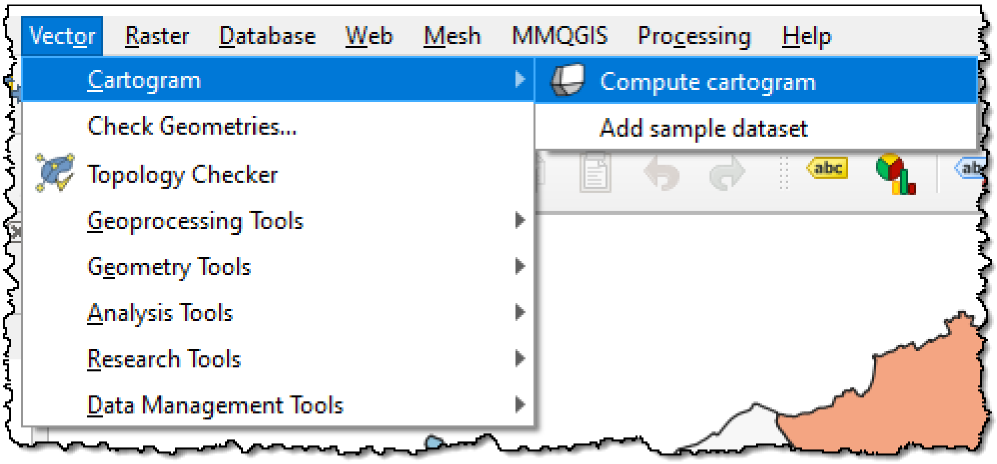
☑ Link class boundaries

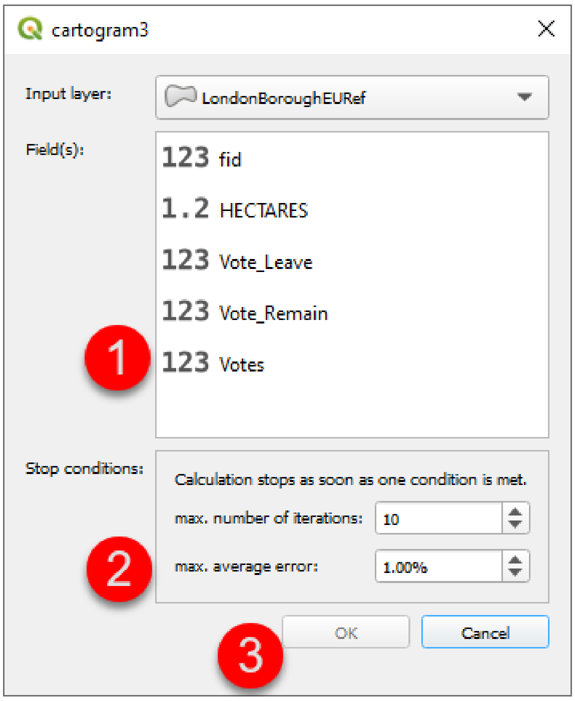
The maps should look like this now:



We can now begin to turn this into a cartogram:

Open the plugin from the menus at the top of the screen **Vector → Cartogram → Compute cartogram**





cartogram3

Input layer: LondonBoroughEURef

Field(s):

- 123 fid
- 1.2 HECTARES
- 123 Vote_Leave
- 123 Vote_Remain
- 123 Votes

Stop conditions: Calculation stops as soon as one condition is met.

max. number of iterations: 10

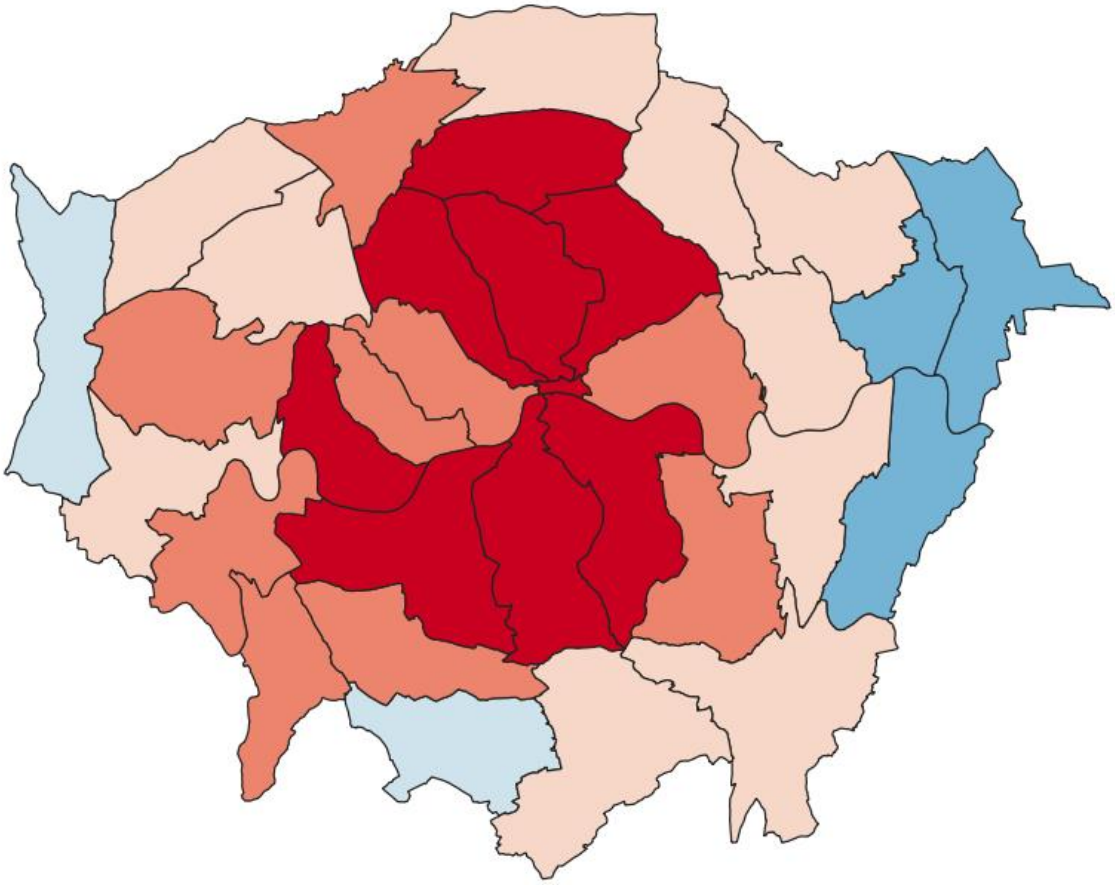
max. average error: 1.00%

Once the options open set it up as in the image here:

- Choose the **Votes** attribute to scale the polygons by.
- Leave the **max. number of iterations** at **10** but decrease the **max. average error** down to **1%**
- Click **OK**

The plugin will now iterate changes to the area of each borough each time getting closer to the proportional change. The plugin keeps going until it has done this for the max number of iterations or until it gets within the max average % error. This represents a difference between the area it has calculated and the actual area, once proportionally adjusted. Each iteration gets further away from the original shape and closer towards the perfect cartogram.

The Result should look something like this:



Cartograms work when they don't warp the boundaries into unrecognisable shapes. If this does happen to your data then one solution is to animate your cartogram. You can use the number iterations to create frames of an animation, each time capturing the image with 1, 2, 3, 4… iterations. You can then stitch these together to create an animation in your favourites GIF making package.

ELECTION MAPPING: DOT DENSITY MAPS

These maps display the density and distribution of a phenomena over a geographic area and are a good alternative to cartograms especially where they would really warp beyond all recognition. The markers, usually a dot or cross, represent the occurrence or an aggregation of occurrences which are then randomly distributed across distinct regions of the map. Colours can be used to represent different classifications to add an extra dimension to the map.

In this exercise we are going to use this to have a different look at the votes in the EU referendum results for London. Each dot represents 100 votes and these have been randomly scattered within each borough to represent the density of votes.

THE DATA

For this exercise we are going to use a geopackage which contains the outlines of the London Boroughs. The attributes of the boroughs include the number of votes to remain and the votes to leave the EU. The data is in the folder you have been provided with.

- Add the **LondonBoroughEURef.gpkg** to a new map.
- If it doesn't automatically update change the **CRS** to be **British National Grid EPSG: 27700**

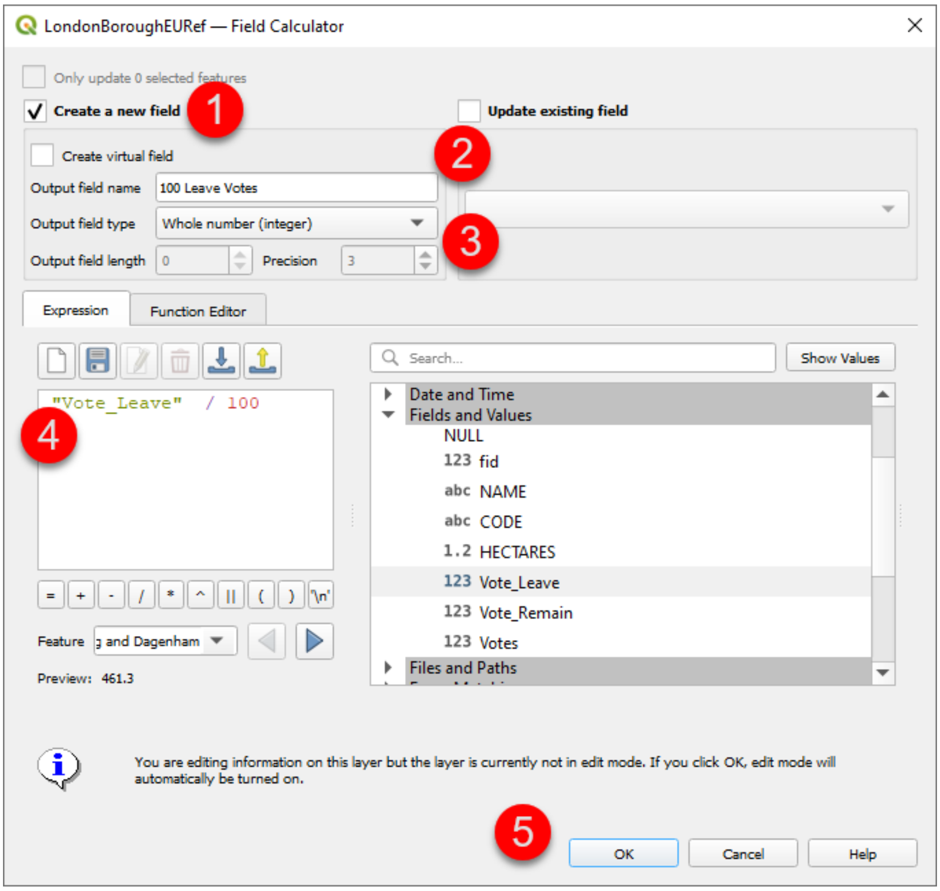
FIELD CALCULATOR

Before we can start generating the dots on the map we need to create a column with a reduced number, as there were too many votes cast to give each one its own dot. To do this we will use the field calculator to create new attribute columns with the numbers of votes each way divided by 100.

- **Right-Click** on **LondonBoroughEURef** in the table of contents and select **Open Attribute Table**

- Click on the **Open Field Calculator** button above the table:

- You will need to set up the Field Calculator as follows:



LondonBoroughEURef — Field Calculator

☐ Only update 0 selected features

☑ Create a new field

☐ Update existing field

☐ Create virtual field

Output field name: 100 Leave Votes

Output field type: Whole number (integer)

Output field length: 0 Precision: 3

```
"Vote_Leave"  / 100
```

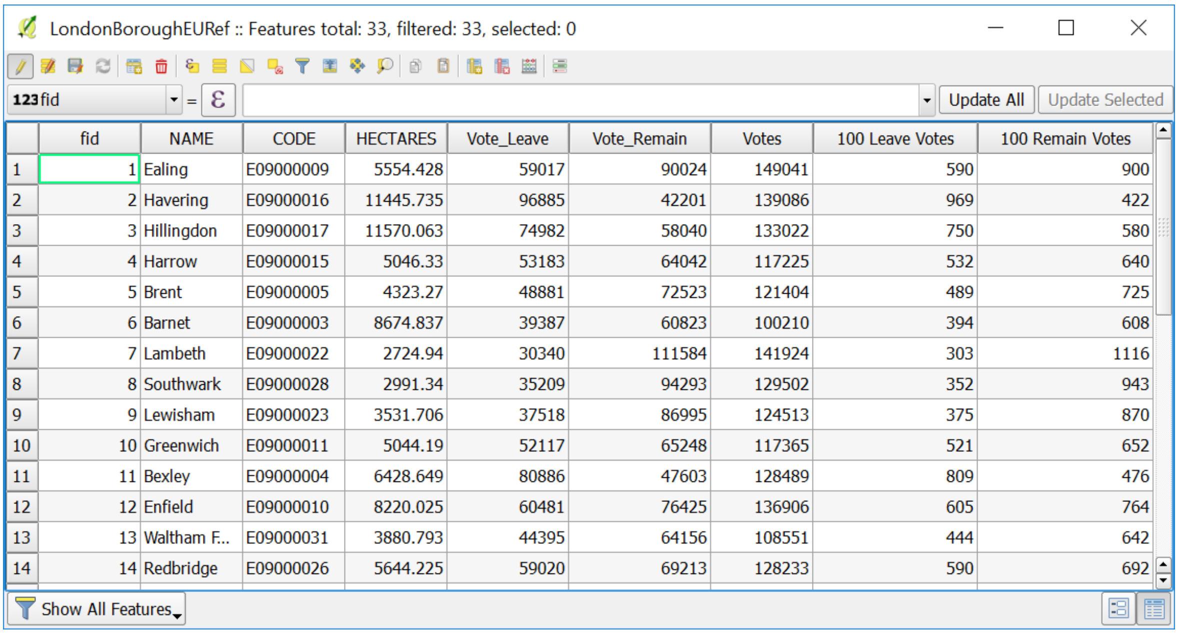
Feature: g and Dagenham

Preview: 461.3

You are editing information on this layer but the layer is currently not in edit mode. If you click OK, edit mode will automatically be turned on.

1. Make sure the **Create a new field** option is selected.
2. Enter the name **100 Leave Votes.**
3. Make sure the **Output field type** is set to **Whole number (integer)**.
4. The expression needs to be: ***"Vote_Leave" / 100*** you can user the **Fields and Values** section to the right to make sure you get the right field names and quote marks; just double click on the field name you want.
5. Click **OK.**

Now repeat this process to create a similar column for the remain votes, call the column 100 Remain Votes. The table should now look like this:



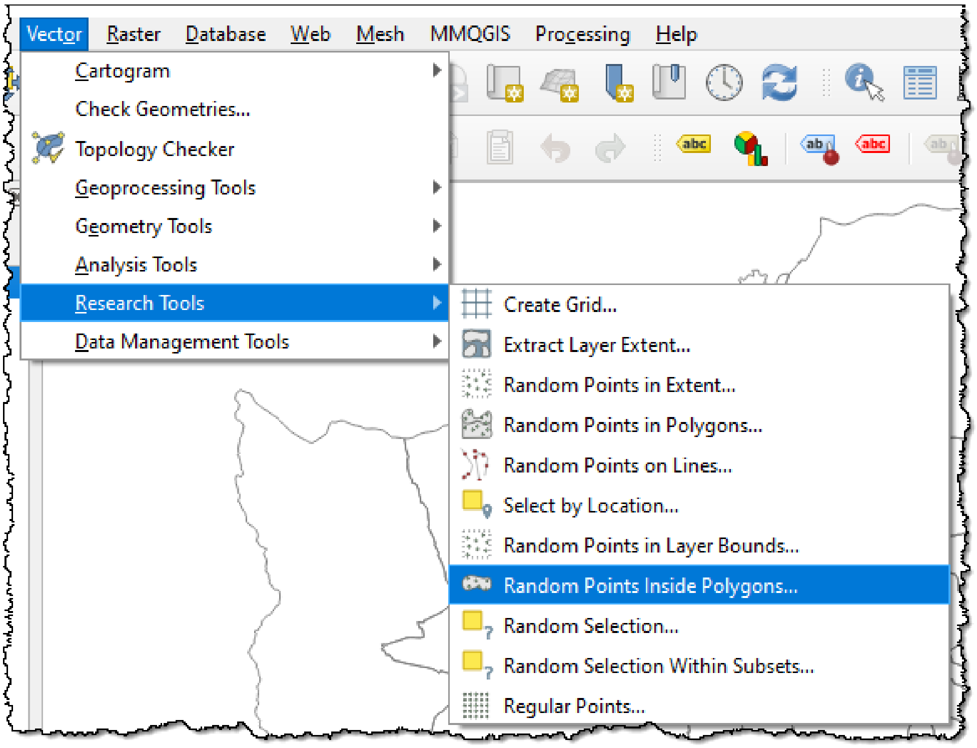
LondonBoroughEURef :: Features total: 33, filtered: 33, selected: 0

	fid	NAME	CODE	HECTARES	Vote_Leave	Vote_Remain	Votes	100 Leave Votes	100 Remain Votes
1	1	Ealing	E09000009	5554.428	59017	90024	149041	590	900
2	2	Havering	E09000016	11445.735	96885	42201	139086	969	422
3	3	Hillingdon	E09000017	11570.063	74982	58040	133022	750	580
4	4	Harrow	E09000015	5046.33	53183	64042	117225	532	640
5	5	Brent	E09000005	4323.27	48881	72523	121404	489	725
6	6	Barnet	E09000003	8674.837	39387	60823	100210	394	608
7	7	Lambeth	E09000022	2724.94	30340	111584	141924	303	1116
8	8	Southwark	E09000028	2991.34	35209	94293	129502	352	943
9	9	Lewisham	E09000023	3531.706	37518	86995	124513	375	870
10	10	Greenwich	E09000011	5044.19	52117	65248	117365	521	652
11	11	Bexley	E09000004	6428.649	80886	47603	128489	809	476
12	12	Enfield	E09000010	8220.025	60481	76425	136906	605	764
13	13	Waltham F...	E09000031	3880.793	44395	64156	108551	444	642
14	14	Redbridge	E09000026	5644.225	59020	69213	128233	590	692

Click **Save edits** then toggle out of **Edit mode**

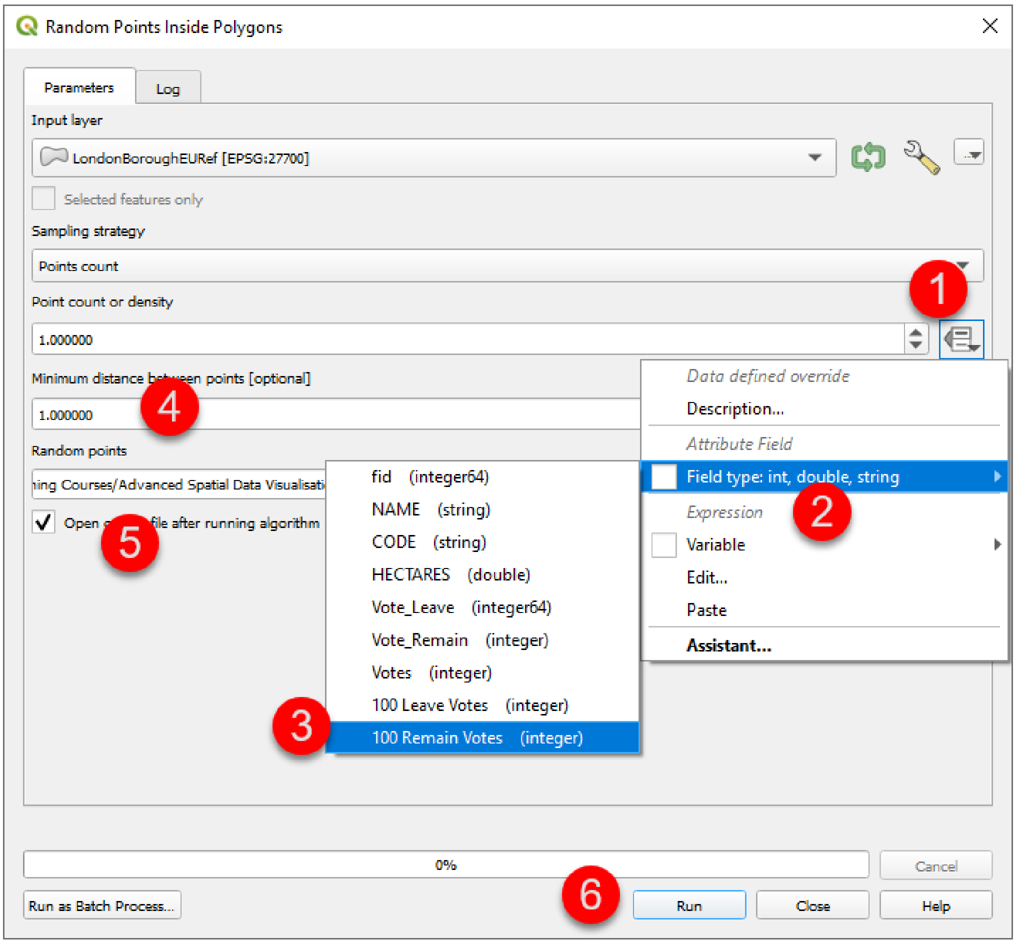
Random points

Now we have the reduced quantities we can use QGIS to randomly assign points across each area.



Select the **Random Points Inside Polygons** from the **Vector → Research Tools** menu.

Fill in the values as follows:



Random Points Inside Polygons

Input layer: LondonBoroughEURef [EPSG:27700]

□ Selected features only

Sampling strategy: Points count

Point count or density: 1.000000

Minimum distance between points [optional]: 1.000000

Random points: ning Courses/Advanced Spatial Data Visualisati

☑ Open output file after running algorithm

1. Make sure the Sampling strategy is set to **Points count** and where it says Point count or density, click on the **filing cabinet** icon at the end.
2. Click on **Field type int, double, string.**
3. Choose **100 Remain Votes**
4. Set a **Minimum distance** of **1 metre** It is good to not have the points overlapping but not essential as we will mitigate this at a later stage. If you set this too high there may not be room for all the dots in the smaller boroughs.
5. **Save** your file as a new Geopackage (or shapefile if you want to). These files will be quite big so it is best not to create a temporary layer. Call this one **Remain Dots** and set the **Output Layer** as **Remain Dots** when the small box pops up.
6. Click **Run**. It will take a while as the algorithm has to cycle through the 33 boroughs.

Now rerun the Random points inside polygons (variable) for the remain votes. Use the same process as above, but use **100 Leave Votes** as the **Number Value**, call the new geopackage and output layer **Leave Dots.**

You will now have a map with three layers, smothered in dots!

STYLING THE MAP

As it stands this map is rather useless, the points are all overlapping, and you can't see the leave votes as the remain all lie over the top of them. Thankfully, we can use the power of QGIS to style the points in a way that makes gives equal emphasis to dots from each layer.

First, we will style the London Borough polygons so that they don't interfere with the dots.

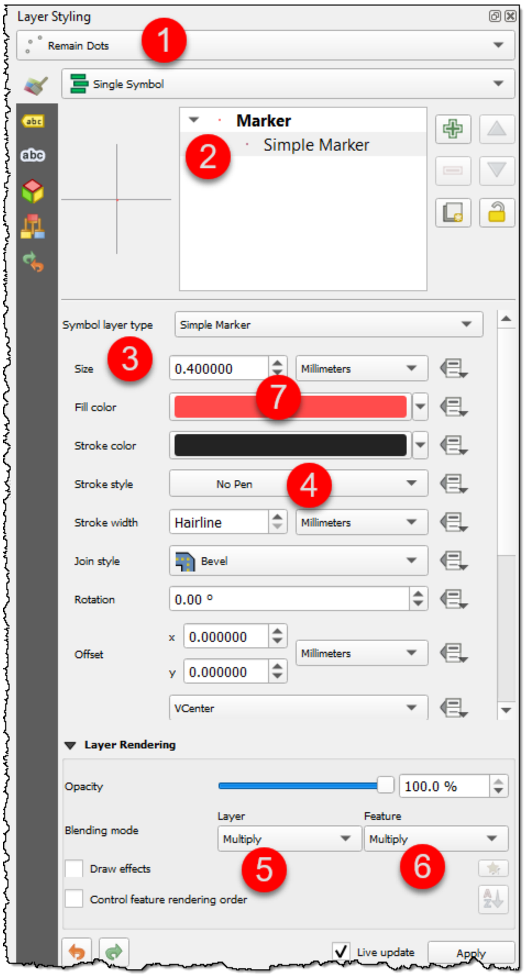
- Click on the **LondonBoroughsEURef** in the layers panel to highlight it and then press **F7** on your keyboard to open the **Live Style Dock** on the right side of the map.
- Click **Simple Fill** at the top of the **Layer Styling** panel.
- Change the **Fill** to **Transparent.**
- Change the **Stroke colour** to be **dark grey,** its less harsh than black**.**
- Change the **Stroke width** to be **0.4 millimetres**

The Outlines will now be clear but there will be no fill colour to interfere with the dots. We can now move on to styling the points. We need to reduce the size of the points so the map isn't too crowded and give them a good colour. We also need to set a blend mode to account for any over lapping points.

1. Use the drop-down menu at the top of the **Layer Styling** panel to change the layer to **Remain Dots**.
2. Click **Simple Marker** to reveal more options.
3. Set the **Size** to be **0.4 (Millimetres).**
4. Set the **Stroke Style** to be **No Pen.**
5. Set the **Layer lending mode** to be **Multiply**
6. Set the **Feature blending mode** to be **Multiply**
7. For **Fill color**, click on the colour swatch.

We want to set the colour as red but we want to take down the saturation. This is to allow overlapping features to have a stronger colour; either a strong red where two features in the same layer are blended or a purple where there is overlap with the blue leave dots.

By picking colours of the same intensity we have also made sure that the map has minimal bias to one colour or the other. This is also maintained by the use of the Blend modes which mix into purple and so minimise the bias where one dot may have overlapped another of a different colour.



Remain Dots

Single Symbol

Marker

Simple Marker

Symbol layer type: Simple Marker

Size: 0.400000 Millimeters

Fill color: 🟥

Stroke color: ⬛

Stroke style: No Pen

Stroke width: Hairline Millimeters

Join style: Bevel

Rotation: 0.00 °

Offset: x 0.000000, y 0.000000 Millimeters

VCenter

Layer Rendering

Opacity: 100.0 %

Blending mode: Layer Multiply, Feature Multiply

□ Draw effects

□ Control feature rendering order

☑ Live update

On the colour chart:

1. Set **R**(ed) to be **255** (Full)

   G(reen) to be **0**.

   B(lue) to be **0**.

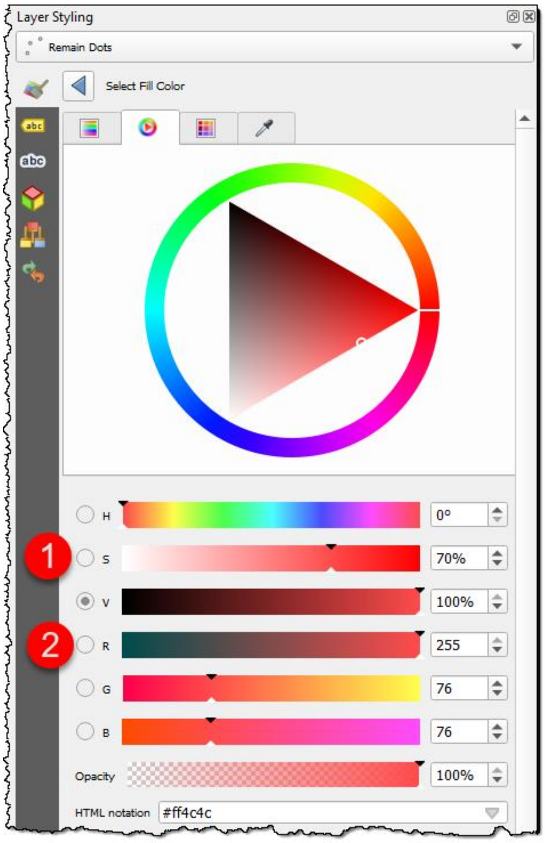
2. Go up to **S**(aturation) and bring this down to **70%** making the red lighter.

   You will notice that the **G** and **B** values increase automatically to **76**.

The colour now has scope to increase its intensity if it overlaps with another dot.

Repeat the process above with the **Leave Dots** layer except here we are going to use blue. Just make sure you increase **B**(lue) to **255**rather than **R**(ed) when selecting the colour.

The map will now look like the one below, it works at scales between 1:200,000 and 1:400,000. You will need to have smaller dots or reduce the saturation (or both) if you want to zoom out further.



Layer Styling

Remain Dots

Select Fill Color

H 0°

S 70%

V 100%

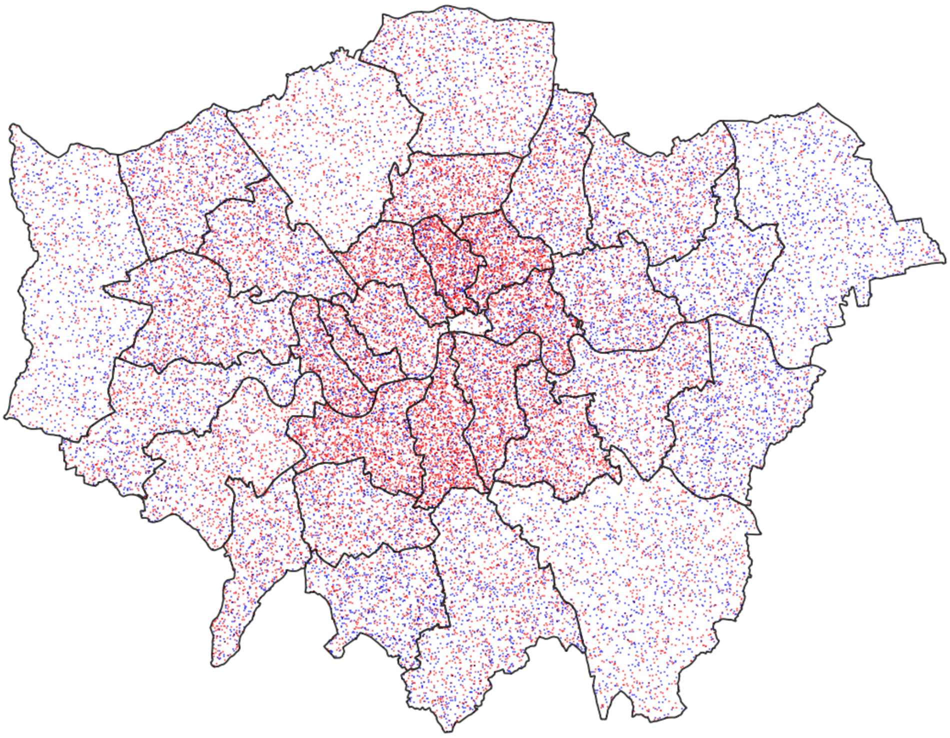
R 255

G 76

B 76

Opacity 100%

HTML notation #ff4c4c



Contains OS data © Crown copyright and database right (2017)

You should now be able to make the distinction between the largely remain voting central London compared to the periphery where voting leave was more popular, particularly in the West. You can also see which boroughs have higher and lower densities of votes

Animating your Heatmap

Now we are going to return to the **Heatmap** of Listed Buildings in Edinburgh. The data has the dates on which each building was listed. We can use this date to break up the data into time periods and then create a map for each one. These are then stitched together to create an animation.

- First, remove the **Kernel Density** surface we produced as we are going to use the rendered points.
- You can now resave the project as **Animated Heatmaps**

To begin we need to set up the data so that QGIS knows which attribute field is the one that has the temporal data.

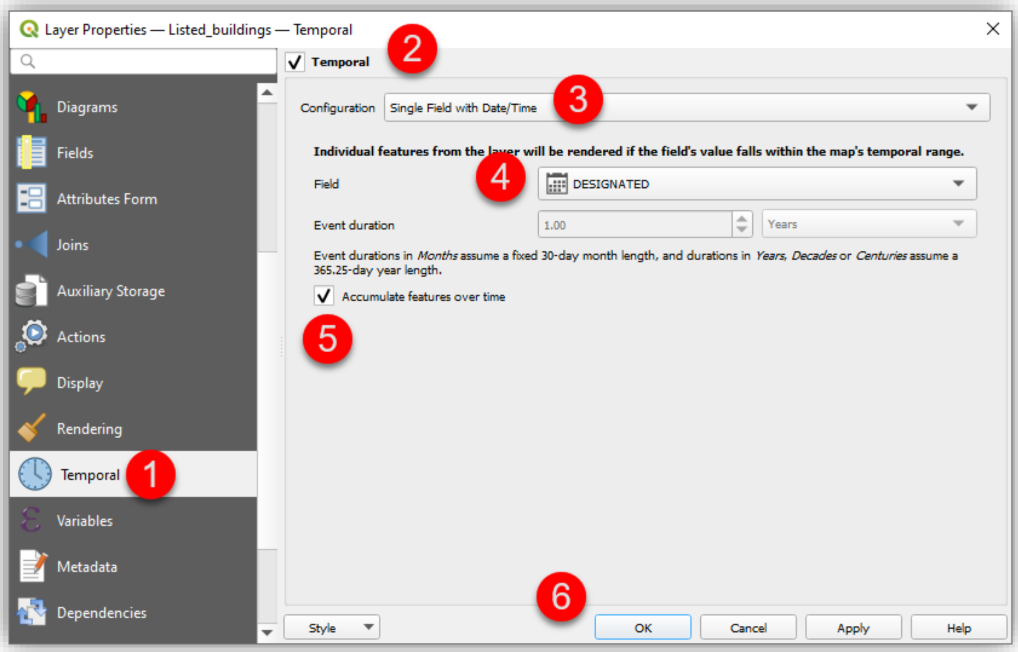
Now we can get on with activating the temporal properties of the dataset. If you open the attributre table you can see there are two columns of data that are dates, Created and Designated. It is the designated one that we are interested in as this is when the Buildings were actually listed

- Right click on the **Listed Buildings** dataset in the left-hand layers pane and click on **properties**.

1. Go to the **Temporal** tab.
2. Tick the **Temporal** box at the top of the screen to enable the options.
3. Select the **Single Field with Date/Time** for the **Configuration.**
4. Set the **Field** to be **Designated.**

Ignore **Event duration.** In this example we are going to accumulate the features over time as the listing of buildings remain, they are not discrete events.

5. Tick the box for **Accumulate features over time**.
6. Click **OK**



Layer Properties — Listed_buildings — Temporal

☑ Temporal

Configuration: Single Field with Date/Time

Individual features from the layer will be rendered if the field's value falls within the map's temporal range.

Field: DESIGNATED

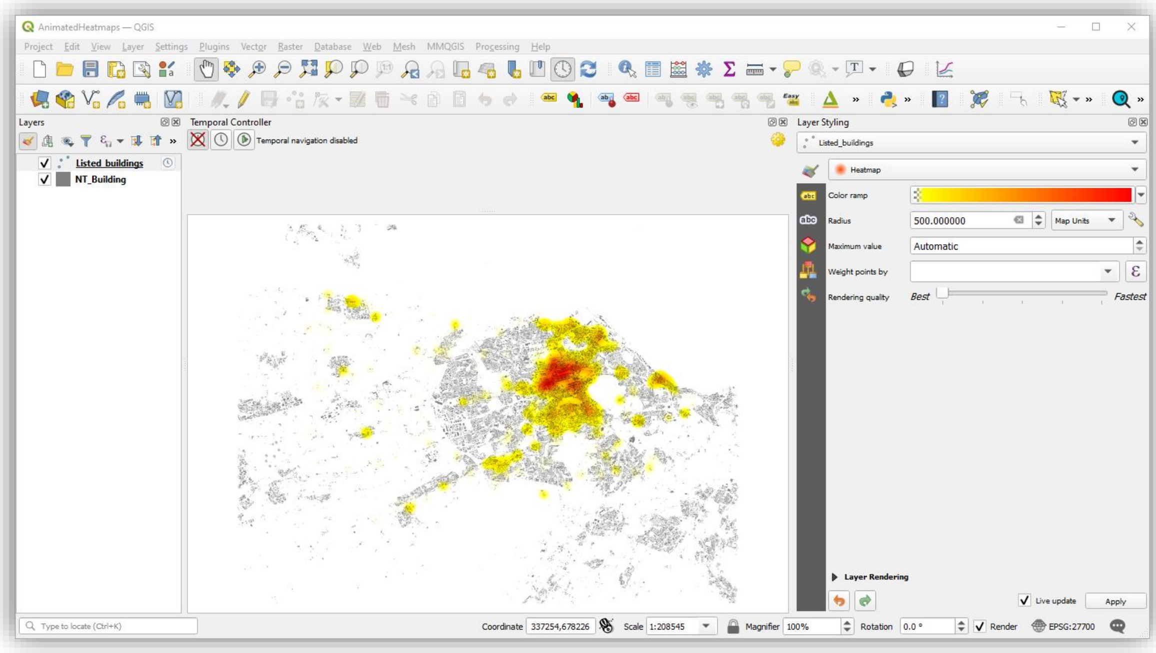
Event duration: 1.00 Years

Event durations in *Months* assume a fixed 30-day month length, and durations in *Years, Decades* or *Centuries* assume a 365.25-day year length.

☑ Accumulate features over time

We now need to activate the **Temporal Controller Panel** to get the controls for the animation.

Right click on the panel full of buttons at the top and select tick the Temporal Controller Panel in the long list that appears. It is in the Top Section for Panels, the lower section is for toolbars. Your screen should look something like this now:



AnimatedHeatmaps — QGIS

Layers: ☑ Listed_buildings, ☑ NT_Building

Temporal Controller: Temporal navigation disabled

Layer Styling: Listed_buildings — Heatmap

Color ramp

Radius 500.000000 Map Units

Maximum value Automatic

Weight points by

Rendering quality Best — Fastest

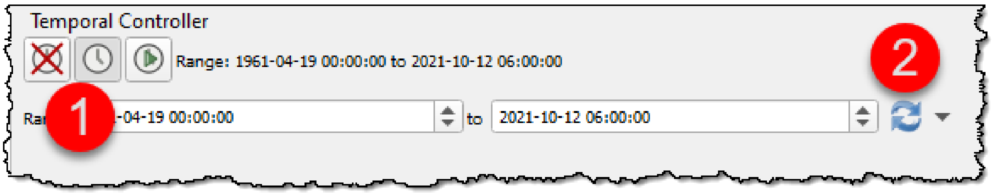
Layer Rendering

☑ Live update Apply

Coordinate 337254,678226 Scale 1:208545 Magnifier 100% Rotation 0.0 ° ☑ Render EPSG:27700

Project Edit View Layer Settings Plugins Vector Raster Database Web Mesh MMQGIS Processing Help
Type to locate (Ctrl+K)

- Click on the **Clock** 🕓 button to set up the date range for the animation.

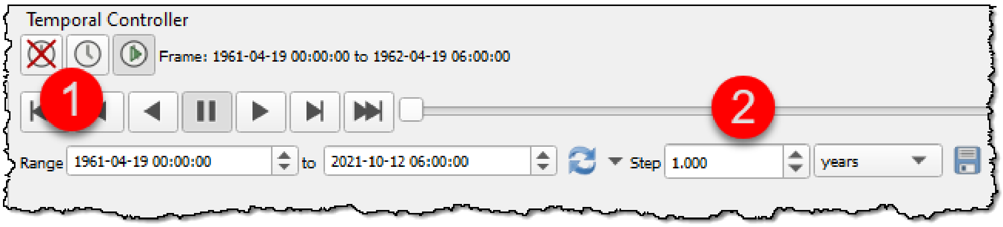


Temporal Controller

Range: 1961-04-19 00:00:00 to 2021-10-12 06:00:00

Ra[nge] -04-19 00:00:00 to 2021-10-12 06:00:00

- Click on the **Blue Arrows** 🔄 to pull in the first and last dates from the dataset.
- Click on the **Green Arrow** ▶ button to open the player controls.



Temporal Controller

Frame: 1961-04-19 00:00:00 to 1962-04-19 06:00:00

Range 1961-04-19 00:00:00 to 2021-10-12 06:00:00 Step 1.000 years

- Change the **Step** to be **1 year**.
- Click on the **Settings** button ⚙
- In the new panel change the **Frame rate** to be **4 per second**.
- Click the Blue back arrow ◀ to go back to the player controls.

- Click the **play button** ▶

The map sequence will play in the map window, but as the amount of data it needs to process increases you will notice some white flashes between frames.

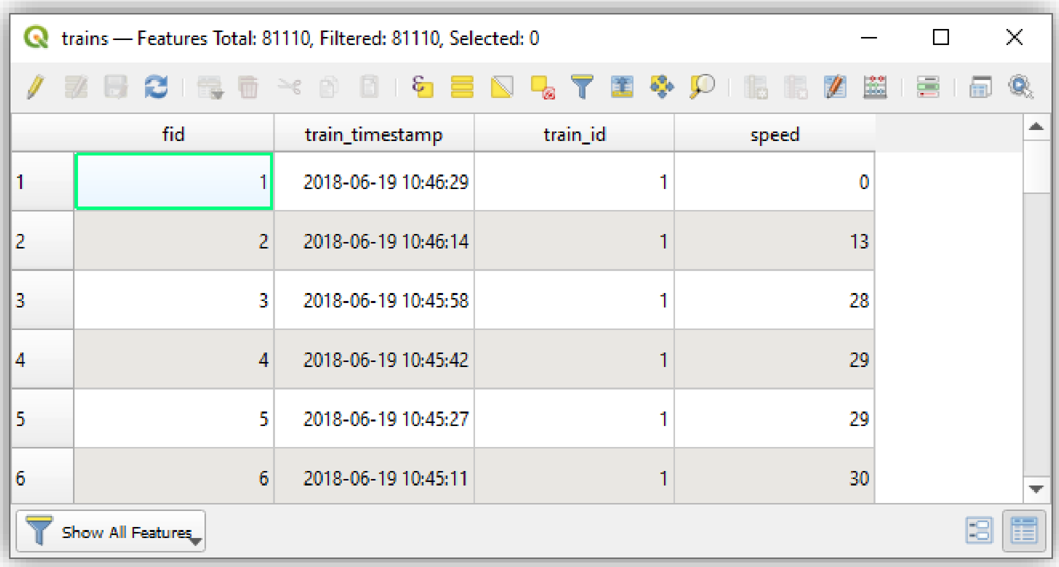
If you click the **Export Video** 💾 button it will output a PNG file for each map frame. These can be stitched together very easily into a video in a range of different software: Windows Video Editor, Blender etc.

Save your Map Project and we can move on to the next section.

Animating Point Data

Similar to the process for animating heatmaps we are now going to look at animating some point data. IN this example we are going to look at some tracking data from trains moving around Finland. This will give you some exposure on how to deal with movement data in the form of points. This data is a common output from GPS tracking, but may also be generated from journals or other locations from literature or research where people or items are having their movements logged.

- Create a new map project and save it as **Animated Points.**
- Add in the **trains.gpkg** data from the **Animated Points** folder.
- Right click on trains in the Layers panel on the left side of the screen and **Open the Attribute Table.**



trains — Features Total: 81110, Filtered: 81110, Selected: 0

	fid	train_timestamp	train_id	speed
1	1	2018-06-19 10:46:29	1	0
2	2	2018-06-19 10:46:14	1	13
3	3	2018-06-19 10:45:58	1	28
4	4	2018-06-19 10:45:42	1	29
5	5	2018-06-19 10:45:27	1	29
6	6	2018-06-19 10:45:11	1	30

You can see this is very basic data with just the date and time of the location, the train ID and it's speed at the time. Note that each train has its location recorded at 15-16 second intervals.

The map just shows a huge number of points following the railway tracks. What we want to achieve this time is an animation showing the locations of the trains as they move along the tracks. We want to be able to distinguish different trains and it would be good to get an idea of their speed as they move along.

Let’s start setting up the map so that we can see what is going on. We’ll begin with the animation and then move on to styling the points.

- Right click on the **trains** dataset in the left-hand layers pane and click on **properties**.

7. Go to the **Temporal** tab.
8. Tick the **Temporal** box at the top of the screen to enable the options.
9. Select the **Single Field with Date/Time** for the **Configuration.**
10. Set the **Field** to be **train_timestamp.**

    This time we need an **Event duration** as the points are discrete events in time. Set this to be **10 seconds** so each event is over before the next starts**.**

    Ignore the **Accumulate features over time** for this example.

11. Click **OK**

Now we can adjust the settings in the temporal controller panel. This is a day’s worth of data but it changes every 15 seconds so we need to break it up into chunks that are manageable and speed up the frame rate to get through the whole thing quickly!

- Click on the **Clock** button to set up the date range for the animation.
- Click on the **Blue Arrows** to pull in the first and last dates from the dataset.
- Click on the **Green Arrow** button to open the player controls.
- Change the **Step** to be **1 minute**.
- Click on the **Settings** button
- In the new panel change the **Frame rate** to be **20 per second**.
- Click the Blue back arrow to go back to the player controls.
- Click the **play button**

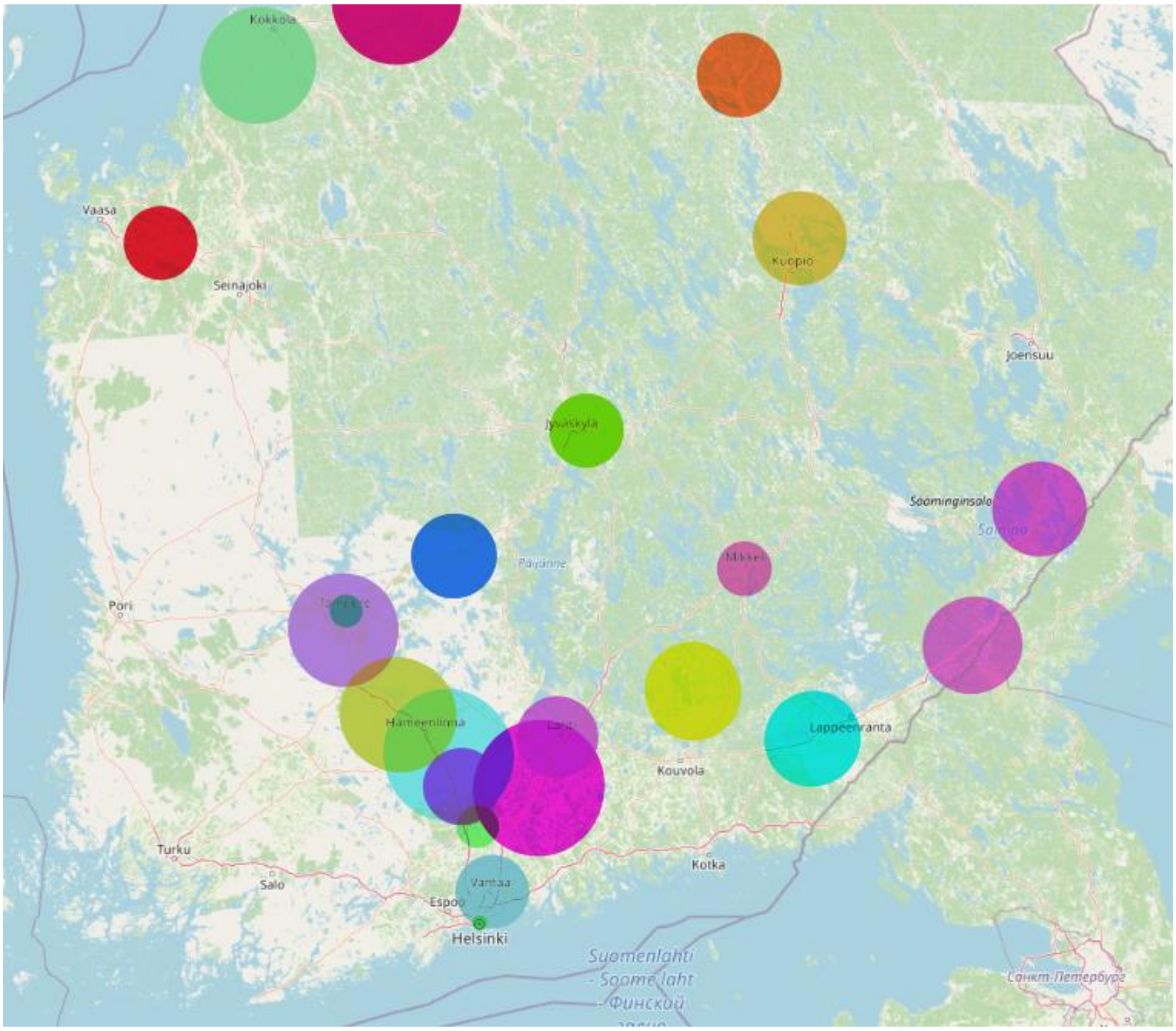
The animation starts off with a single dot but as the day progresses more trains start to move around. To make this a more engaging visualisation we can style the points to be different colours, have a size related to speed and blending mode to see where the trains are heading.

If you zoom in you will notice that there are around 4 dots displayed at anyone time for each train, this is because the Steps are 1 minute.

Now it is over to you to see what you can do with what you have learned so far to:

- Bring in a web basemap (Use QuickMapServices and try OSM Standard)
- Change the **CRS** to match the Basemap (EPSG 3857)
- Assign a random colour to each train. (Use **Categorized** instead of single symbol)

- Change the Size of the point based on the train speed. (Click on Symbol and then the Data Defined Override for Size)
- Add a blend mode so that you can see both trains when they pass each other and the base map through the dots.



This is what I managed with a few changes to the defaults.

Congratulations!

Well done for working through these exercises, please get in touch if you want to know more:

Email: tom.armitage@ed.ac.uk

Twitter: @MapNav_Tom

Data Sources:

Natural Earth Land outlines and Populated Places:

http://www.naturalearthdata.com/downloads/

London Boroughs Election Data

OS BoundaryLine:
https://www.ordnancesurvey.co.uk/opendatadownload/products.html#BDLINE

For Higher Education:

http://digimap.edina.ac.uk/

Election Results:

https://www.electoralcommission.org.uk/

Train Data For Finland:

https://github.com/tjukanovt/tjukanovt.github.io/blob/master/data2share/trainGPS.csv

Other Recommended Plugins:

Qgis2web:

Converts your QGIS Map into a webmap in either leaflet OpenLayers or Mapbox

Data Plotly:

Allows you to put interactive charts alongside your maps! Highly recommended for data visualisation.

Terrain Shading:

Gives you extra power for creating nice hillshades for your maps, including open sky and ambient occlusion… essentially it makes them more realistic.

QGIS Resource Sharing:

Access to styling and SVG point markers created by other people.

Other Resources and People:

Alasdair Rae:

Highly recommended blog, and YouTube beautiful data visualisations and maps in QGIS:

http://www.statsmapsnpix.com/

https://www.youtube.com/@automaticknowledge/videos

Topi Tjukanov:

Great blog for really pushing what is capable with QGIS, some amazing styles and resources too:

https://tjukanov.org/

Klas Karlsson:

More traditional analysis but again more styles to download and reuse and he explains things really well:

http://geosupportsystem.se/

https://www.youtube.com/channel/UCxs7cfMwzgGZhtUuwhny4-Q

Ujaval Ghandi:

A huge amount of information, some out of date but he is very active so it usually gets updated. He has covered a huge range of functionality in QGIS:

https://www.qgistutorials.com/en/