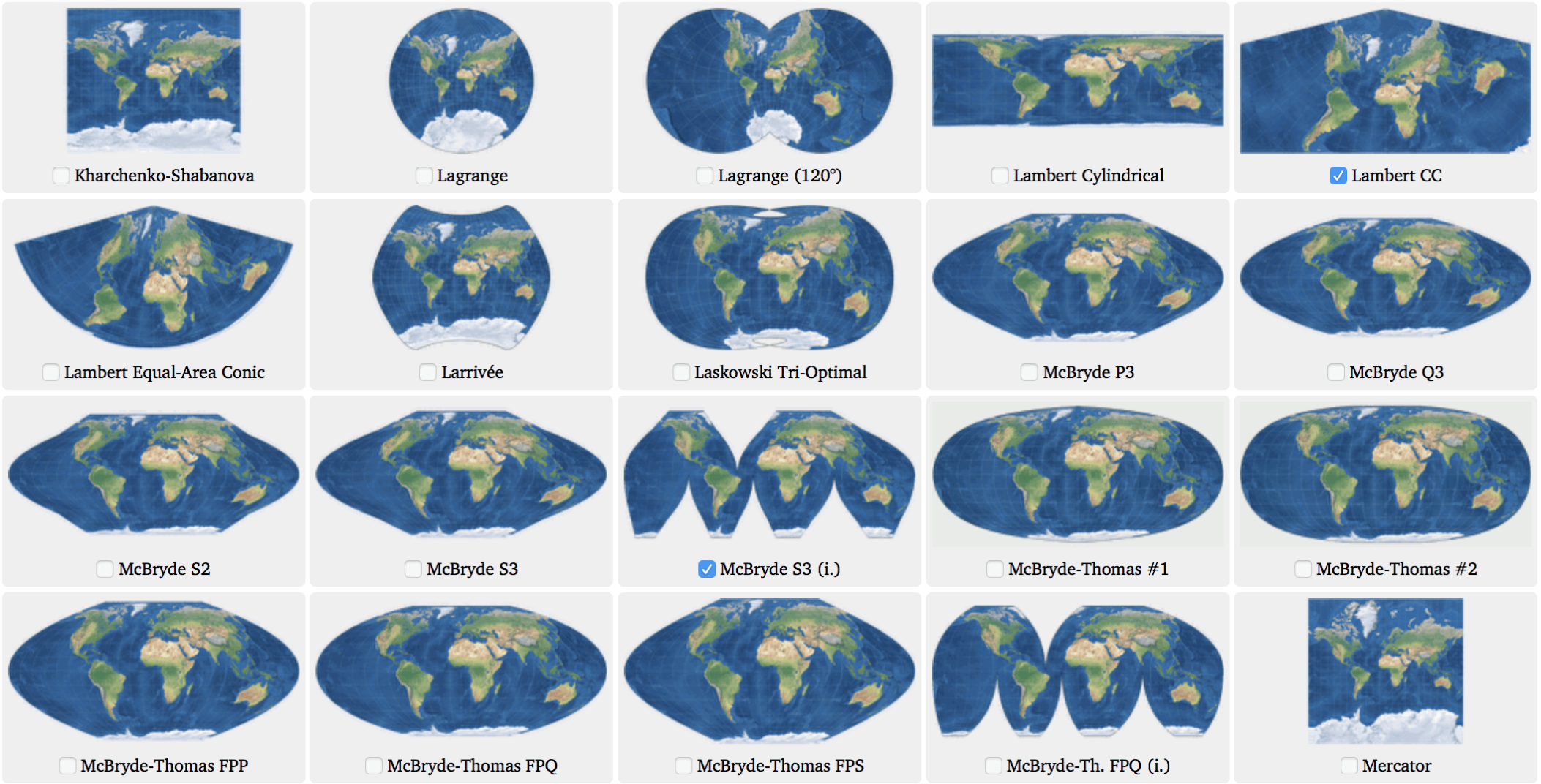
Getting started with spatial data

Mike Spencer

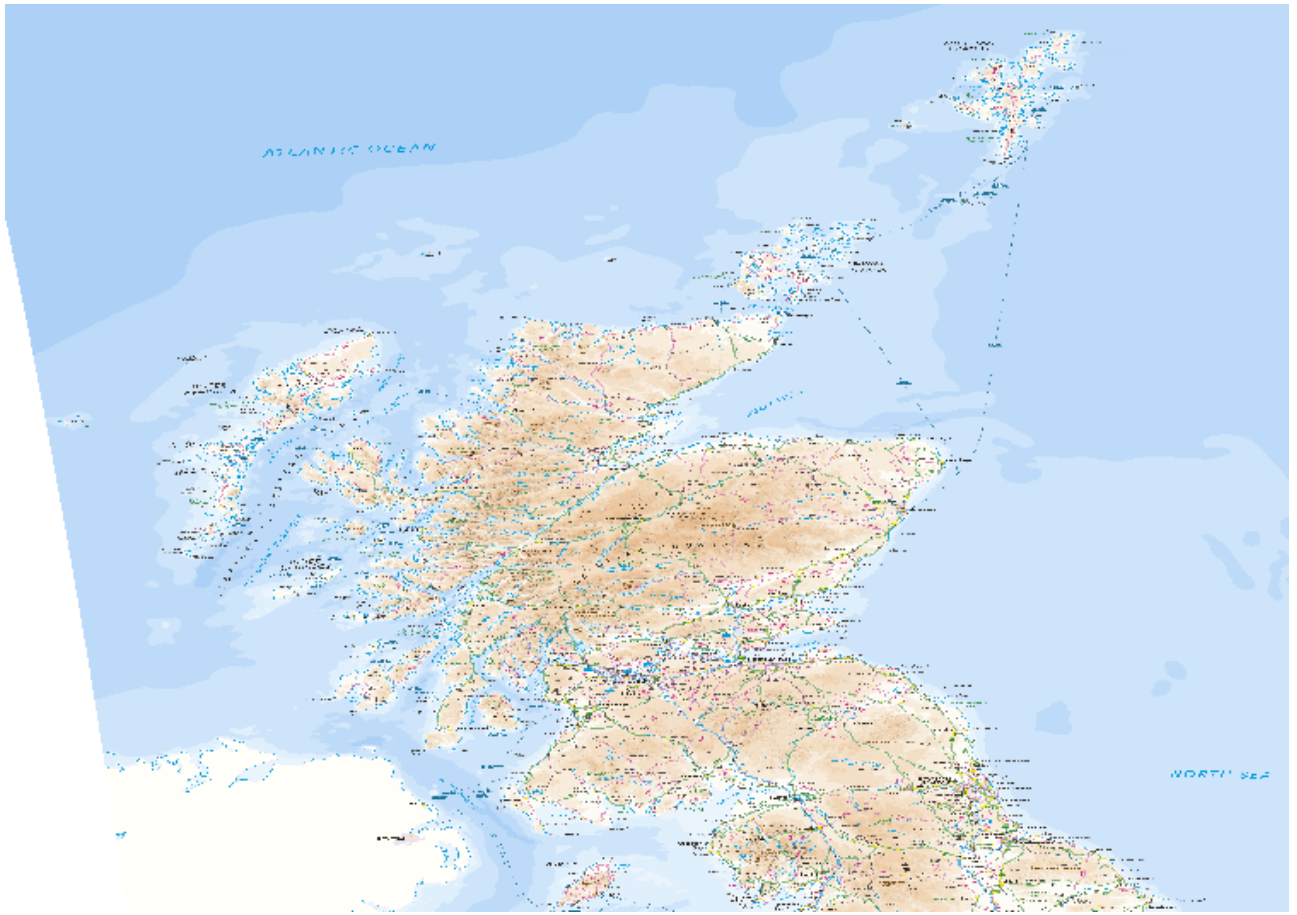
Projections

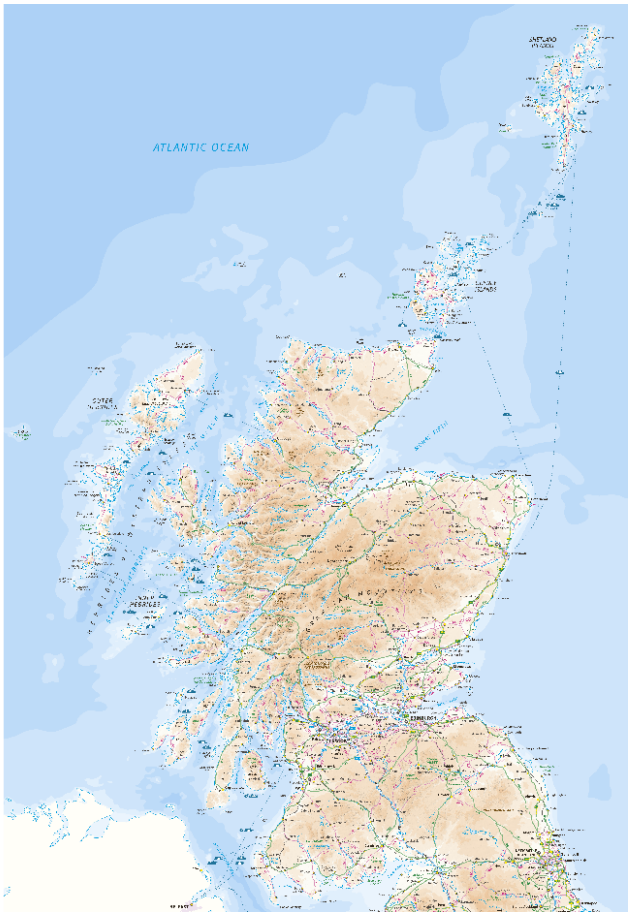


https://i0.wp.com/geoawesomeness.com/wp-content/uploads/2017/04/Map-projections-net-comparison-1.png

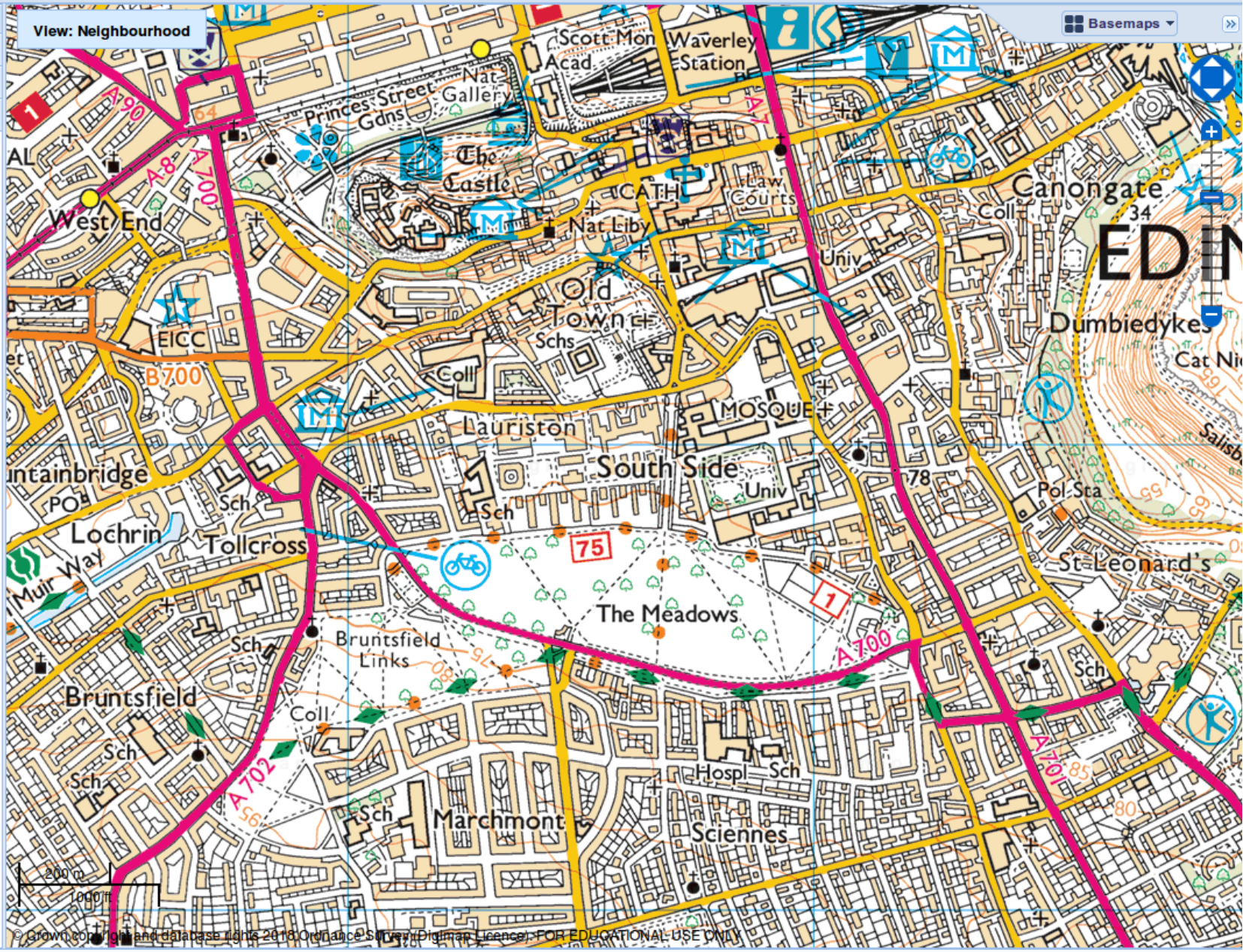
Projections

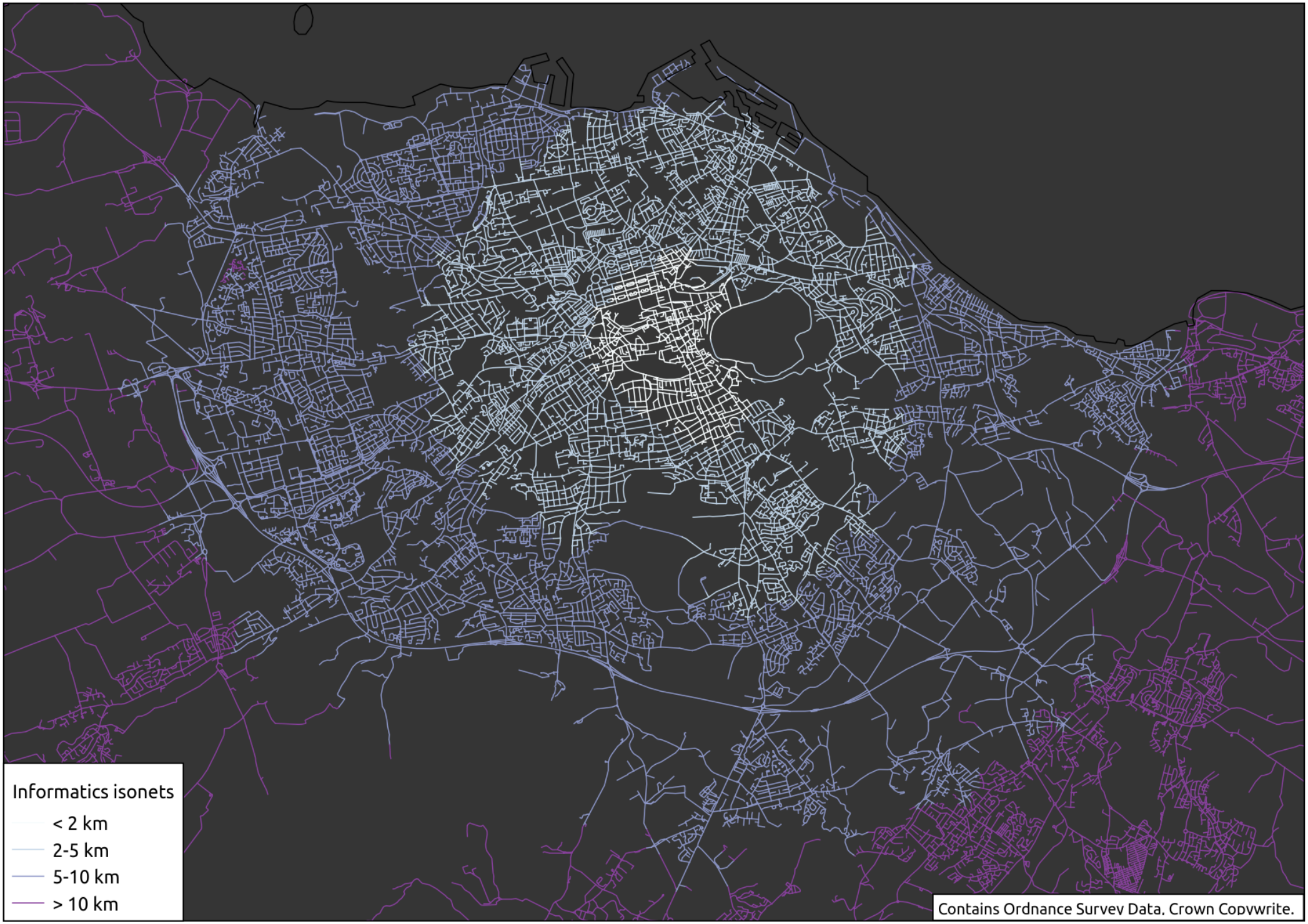
- British national grid EPSG:27700
- World geodetic system EPSG:WGS84





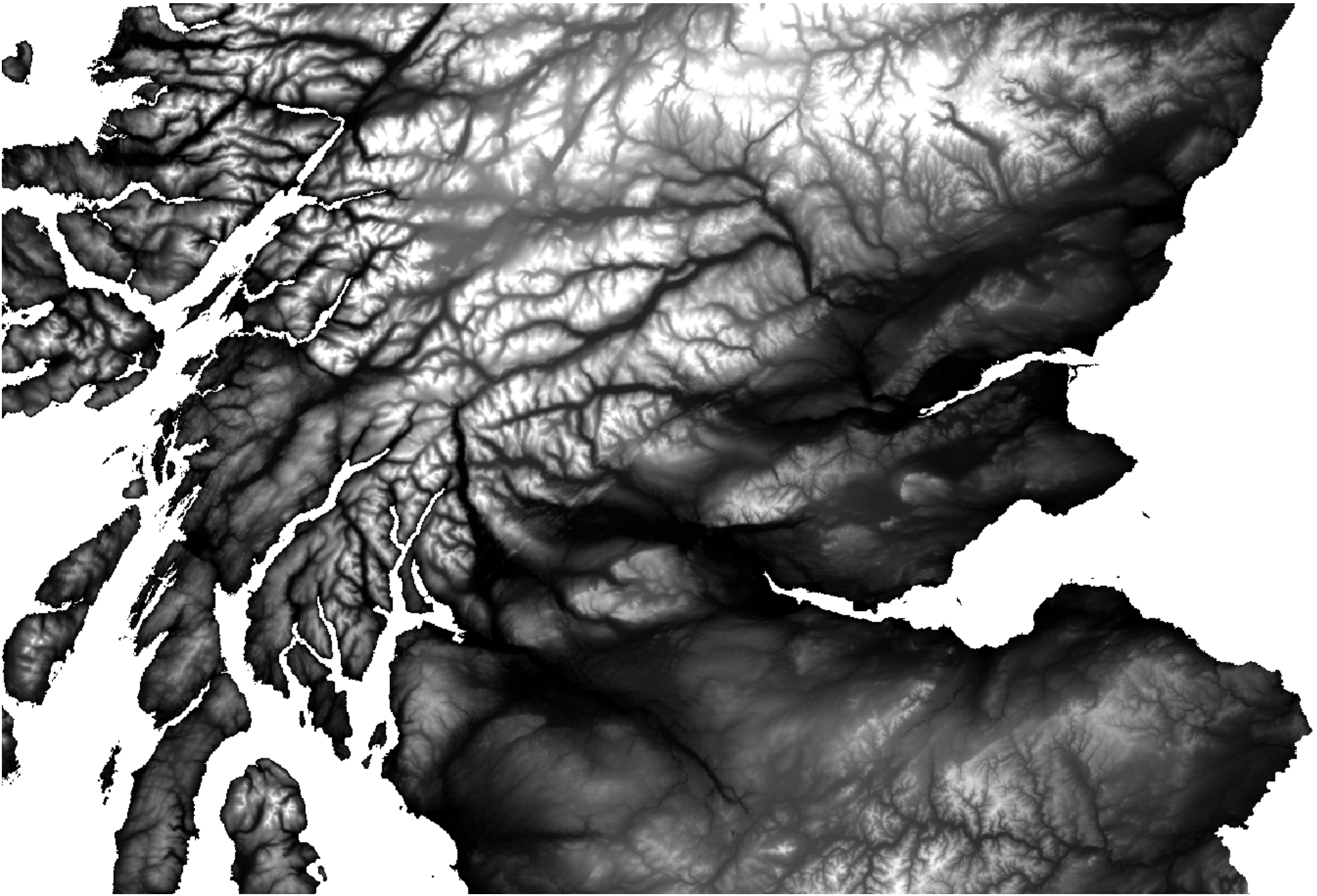
https://www.ordnancesurvey.co.uk/business-and-government/products/miniscale.html



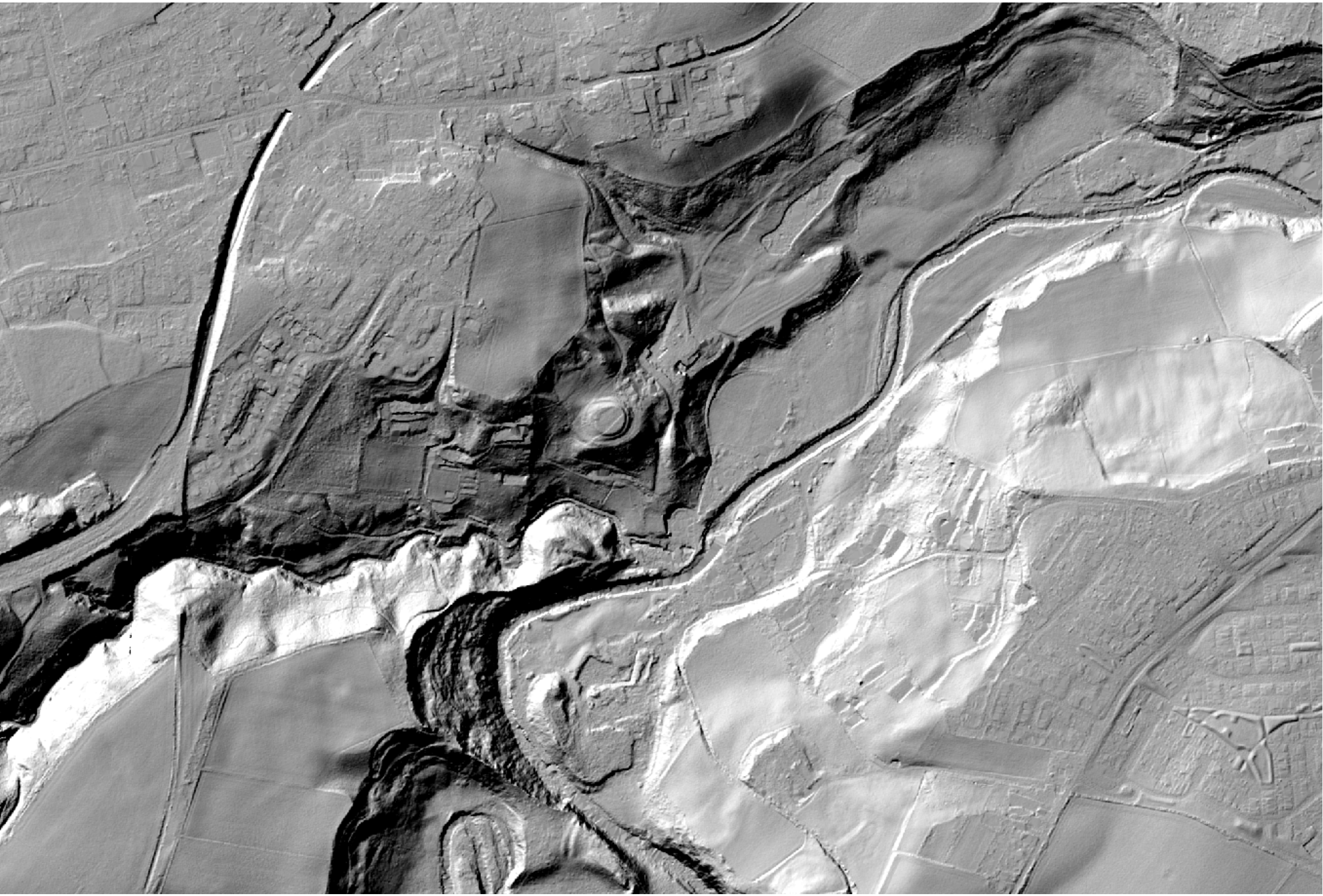


https://scottishsnow.wordpress.com/2017/11/17/network-analysis-qgis-ug/

Raster

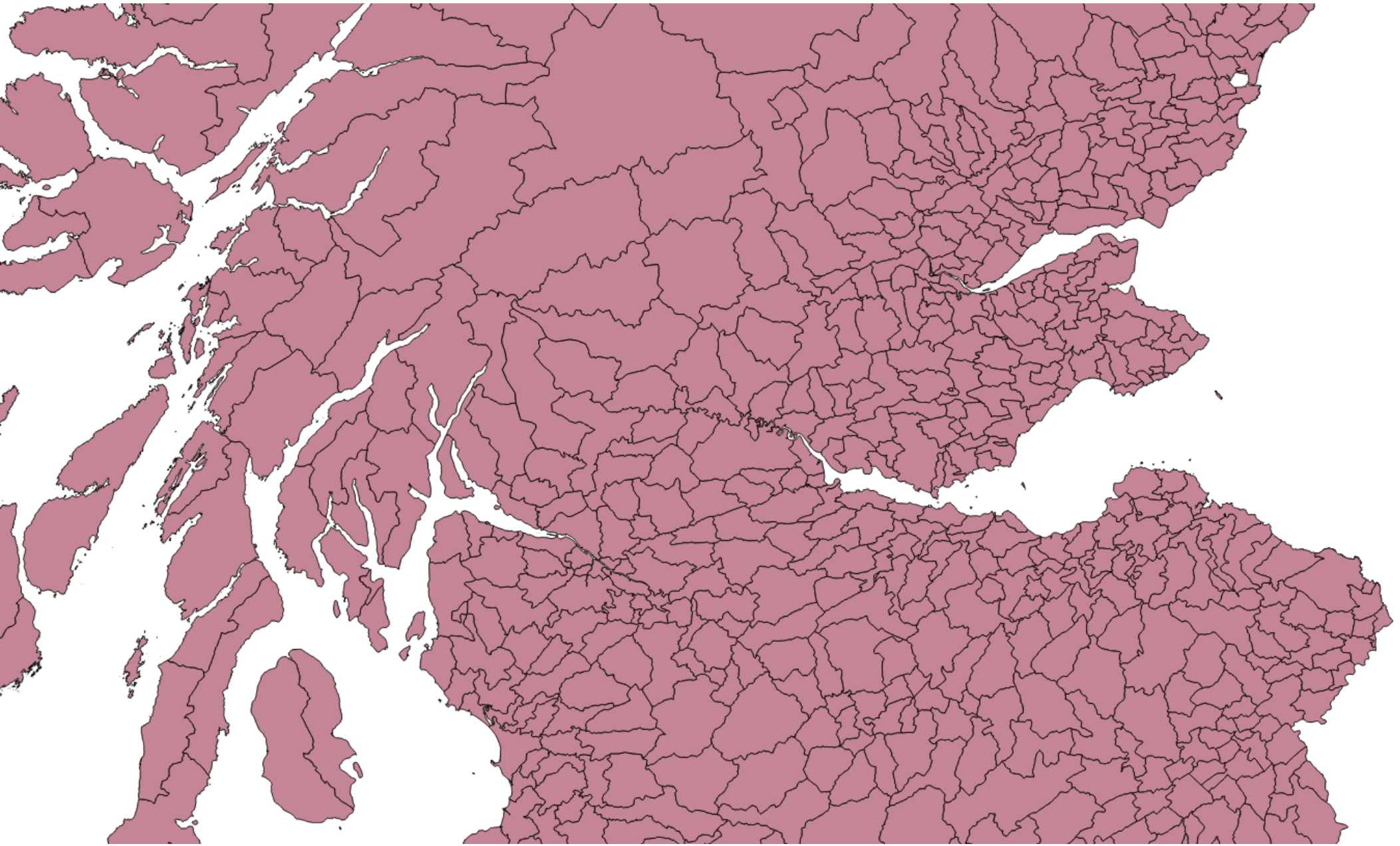


https://www.ordnancesurvey.co.uk/business-and-government/products/terrain-50.html



https://data.gov.uk/data/search?q=lidar+scotland

Vector

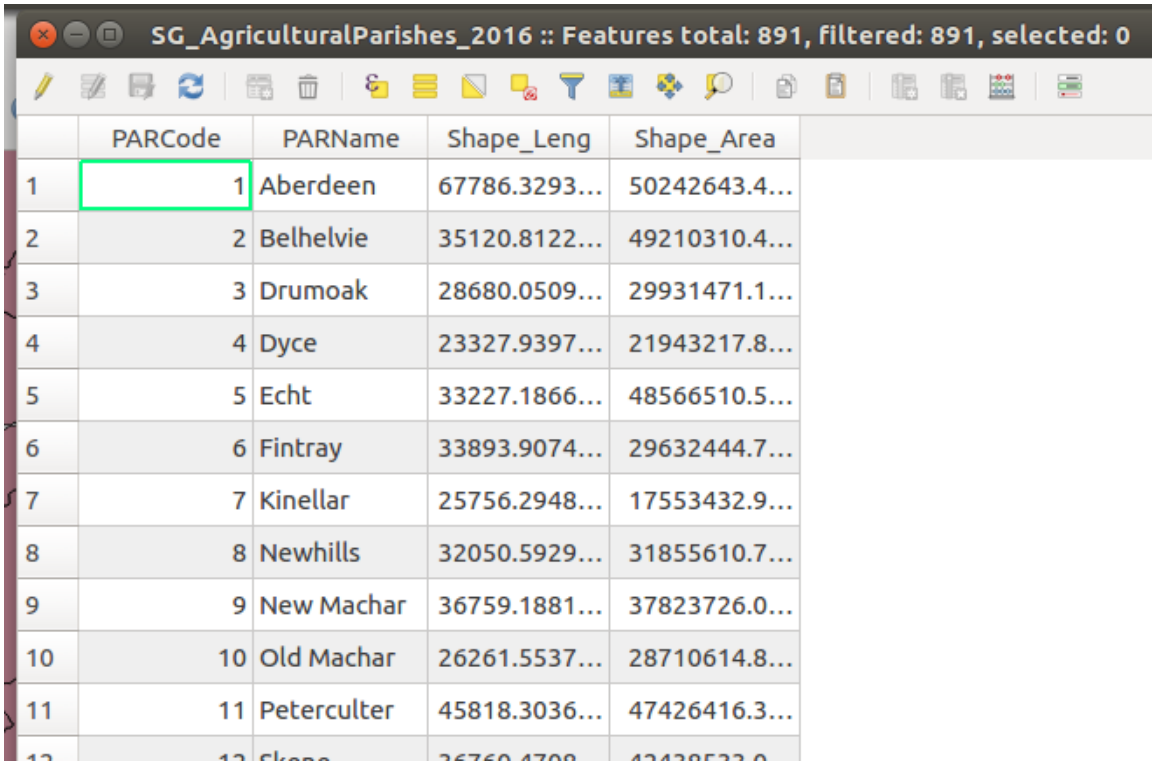


https://data.gov.uk/dataset/agricultural-parishes

Vector

- Polygons
- Polylines
- Points

...but also flat data tables!

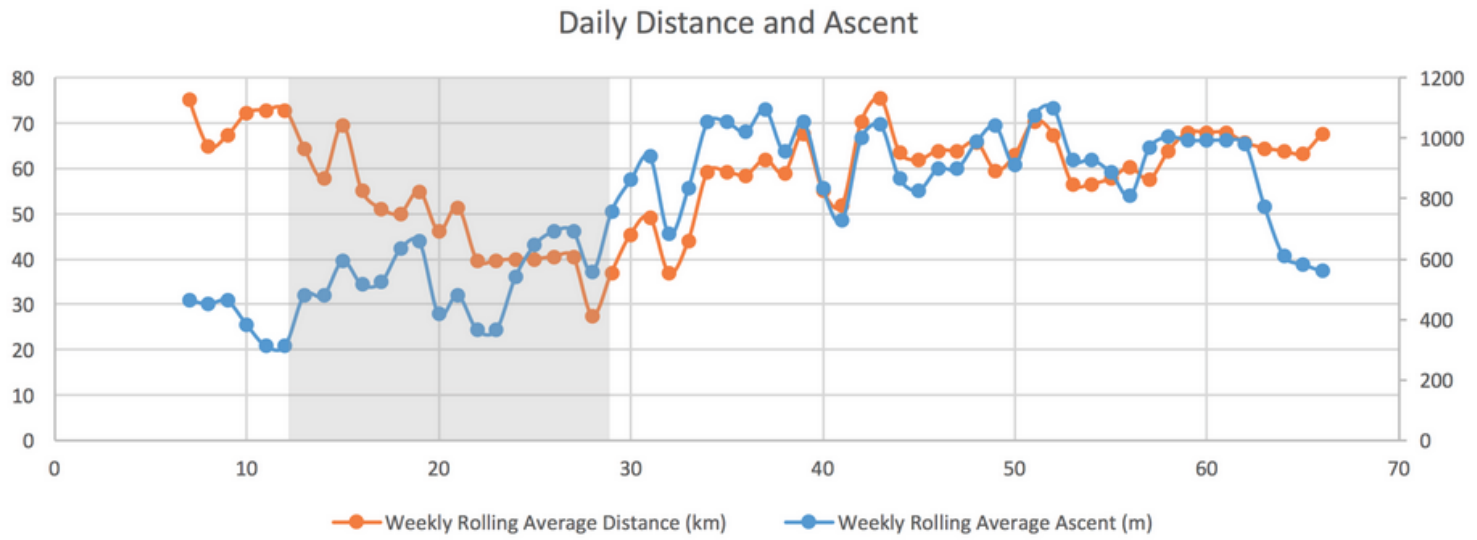


	PARCode	PARName	Shape_Leng	Shape_Area
1	1	Aberdeen	67786.3293…	50242643.4…
2	2	Belhelvie	35120.8122…	49210310.4…
3	3	Drumoak	28680.0509…	29931471.1…
4	4	Dyce	23327.9397…	21943217.8…
5	5	Echt	33227.1866…	48566510.5…
6	6	Fintray	33893.9074…	29632444.7…
7	7	Kinellar	25756.2948…	17553432.9…
8	8	Newhills	32050.5929…	31855610.7…
9	9	New Machar	36759.1881…	37823726.0…
10	10	Old Machar	26261.5537…	28710614.8…
11	11	Peterculter	45818.3036…	47426416.3…

R Example

- Distance travelled: 3966km
- Elevation ascended: 49,219m
- Hours of riding: 334hrs
- Maximum speed: 56.8km/h
- Countries visited: 15
- Crashes: 2 (plus one emergency departure from the road to avoid an oncoming lorry)

Perhaps slightly more interesting is to see how my rate of progress changed over the 10 weeks. To do that, I've calculated a rolling weekly average of daily distance travelled and elevation ascended. You can see it in the graph below:



http://willonwheels.co.uk/the-stats

Process

- Translate files
- Read files
- Map files

Translate files

- https://www.gpsbabel.org/
- For translating gps data

- .fit files to .gpx

```
# Convert from .fit
dir = "~/Cloud/Michael/Projects/Will/data/"
f = list.files("~/Cloud/Michael/Projects/Will/data", pattern="fit")

lapply(f, function(i){
   system(paste0("gpsbabel -i garmin_fit -f ", dir, i, " -o gpx -F ", dir, substr
(i, 1, 8), ".gpx"))
})
```

Read files

- rgdal package – translates most things
- readOGR() for vector data
- Must use full path

```
counties = readOGR(paste0(normalizePath("~"), "/dir/dir"),  "file")
```

- In our case can use list.files(full.names=T)

```
# Read files
f = list.files("~/Cloud/Michael/Projects/Will/data", pattern="gpx", full.names=T)
dat = lapply(f, function(i){
   readOGR(i, layer="track_points")
})
```

Interrogate

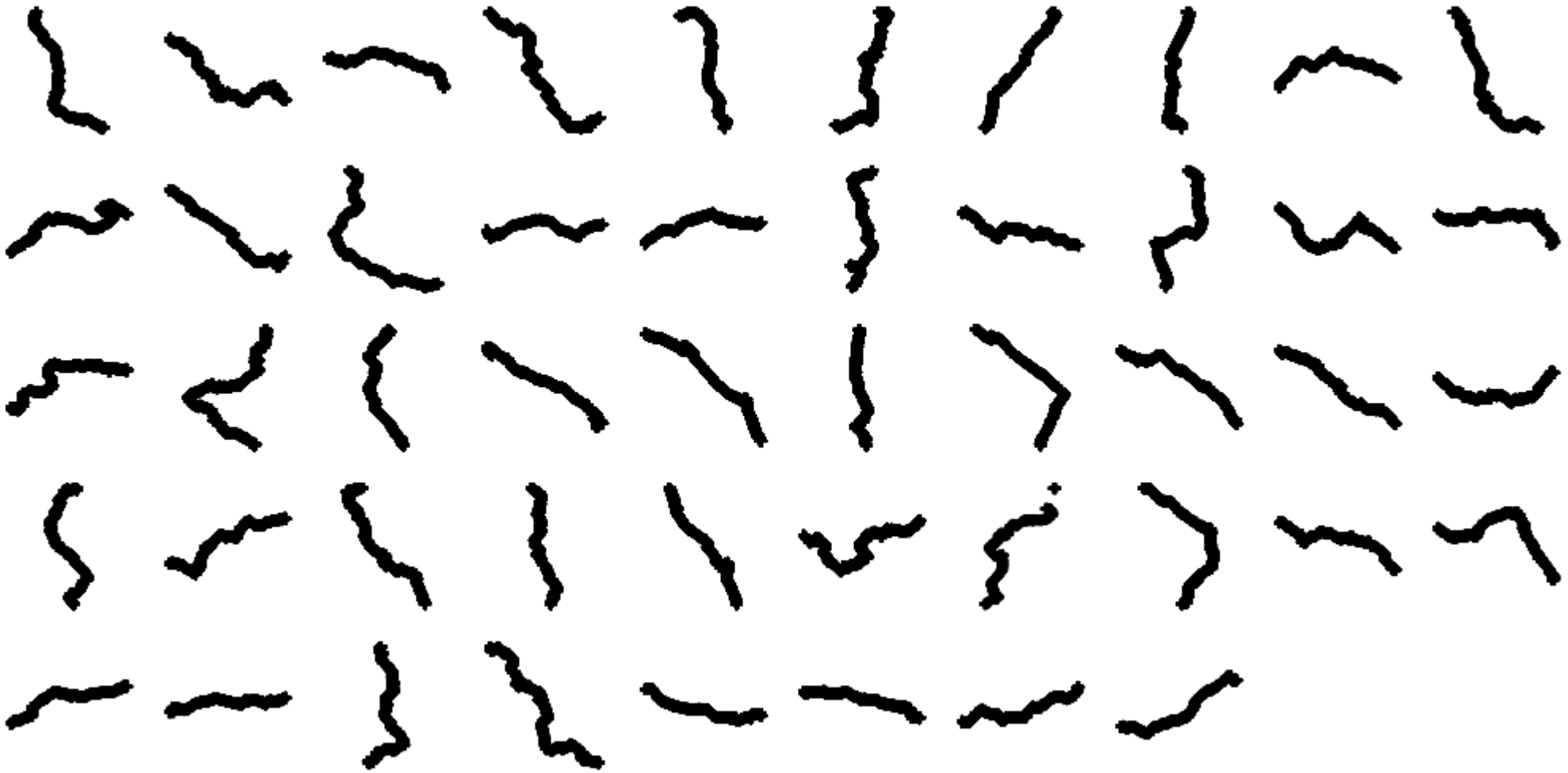
- head(data.frame(spatial_object))
- head(spatial_object@data)
- spatial_object$new_column
- merge(spatial_object, a_dataframe)

- Tidyverse, not so much, probably @data works

Plot

```
png("~/track.png", width=1000, height=500)
par(mfrow=c(5, 10), mar=c(1, 1, 1, 1) + 0.1)
lapply(dat, function(i){
    plot(i)
})
dev.off()
```

Plot



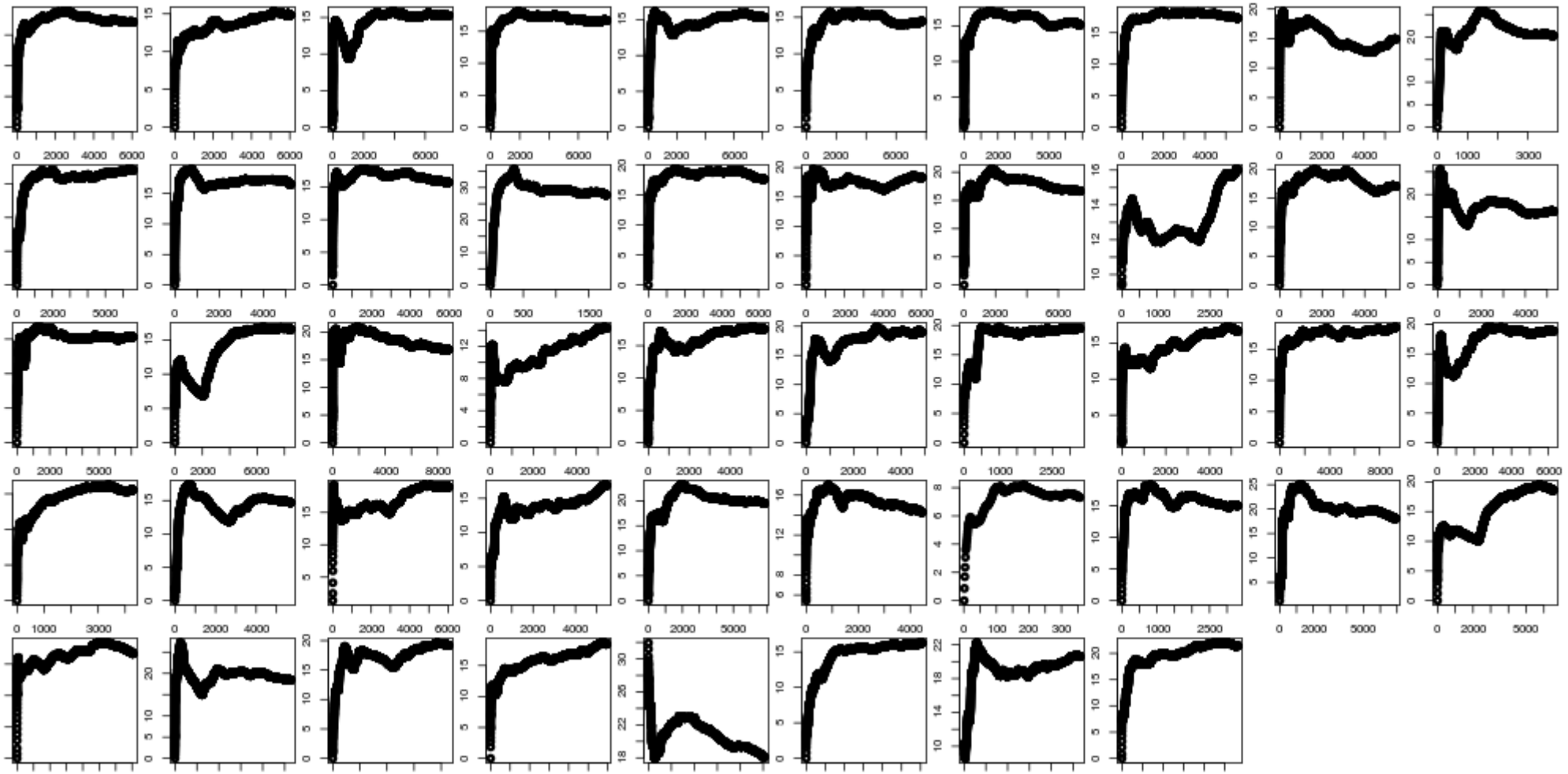
New variables

```r
# New variables
dat = lapply(dat, function(i){
    # Speed in km/h instead of m/s
    i$speed_kmh = i$speed * 3.6

    # Return
    i
})
```

```r
# Evolution of daily averages
daily.average = mclapply(dat, mc.cores=4, function(i){
    sapply(2:nrow(i), function(j){
        mean(i$speed_kmh[1:j])
    })
})
```

parallel package in Linux/Mac only

```r
png("~/daily_average.png", width=1000, height=500)
par(mfrow=c(5, 10), mar=c(1, 1, 1, 1) + 0.1)
lapply(daily.average, function(i){
    plot(i)
})
dev.off()
```



Other (FOSS) tools

- QGIS https://qgis.org
- GRASS https://grass.osgeo.org/

NAME

v.net.distance - Computes shortest distance via the network between the given sets of features.
Finds the shortest paths from each 'from' point to the nearest 'to' feature and various information about this relation are uploaded to the attribute table.

KEYWORDS

vector, network, shortest path

SYNOPSIS

v.net.distance
v.net.distance --help
v.net.distance [**-gl**] **input**=*name* **output**=*name* [**arc_layer**=*string*] [**arc_type**=*string*[,*string*,...]] [**node_layer**=*string*] [**from_layer**=*string*] [**from_cats**=*range*] [**from_where**=*sql_query*] [**to_layer**=*string*] [**to_type**=*string*[,*string*,...]] [**to_cats**=*range*] [**to_where**=*sql_query*] [**arc_column**=*name*] [**arc_backward_column**=*name*] [**node_column**=*name*] [**--overwrite**] [**--help**] [**--verbose**] [**--quiet**] [**--ui**]

Flags:

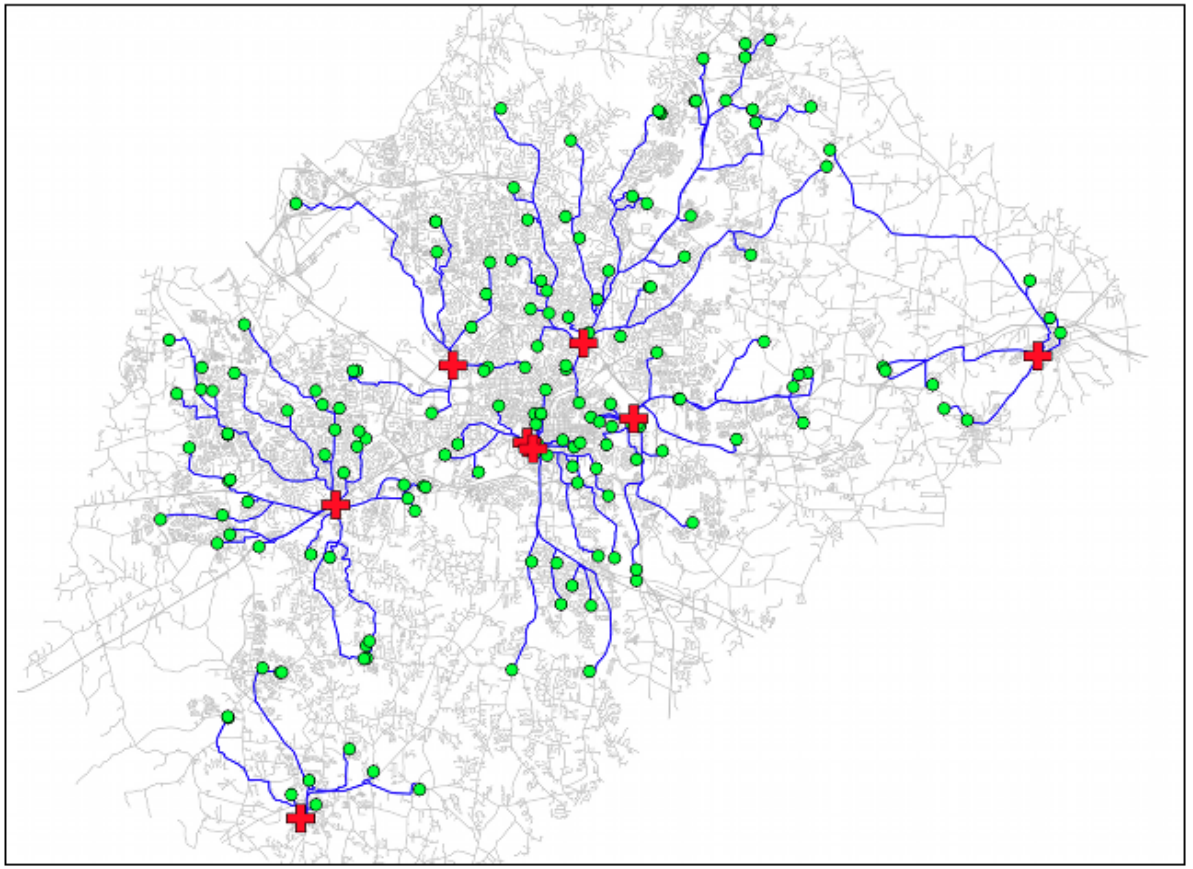
-g
Use geodesic calculation for longitude-latitude locations

-l
Write each output path as one line, not as original input segments.

--overwrite
Allow output files to overwrite existing files

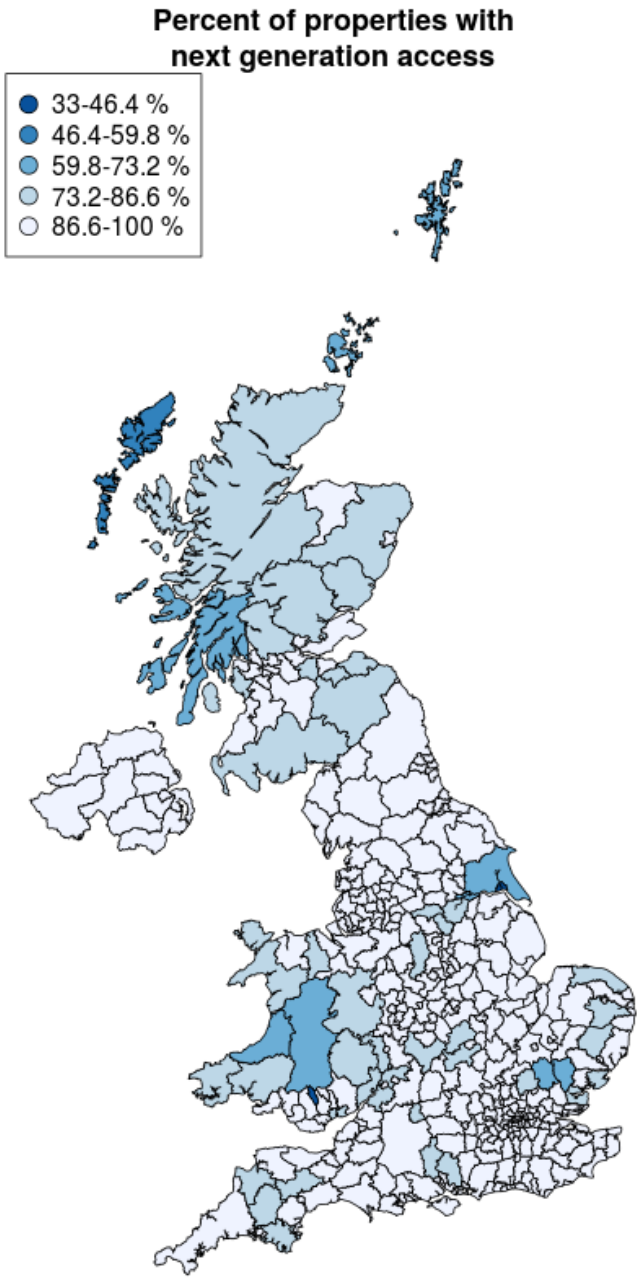
--help
Print usage summary

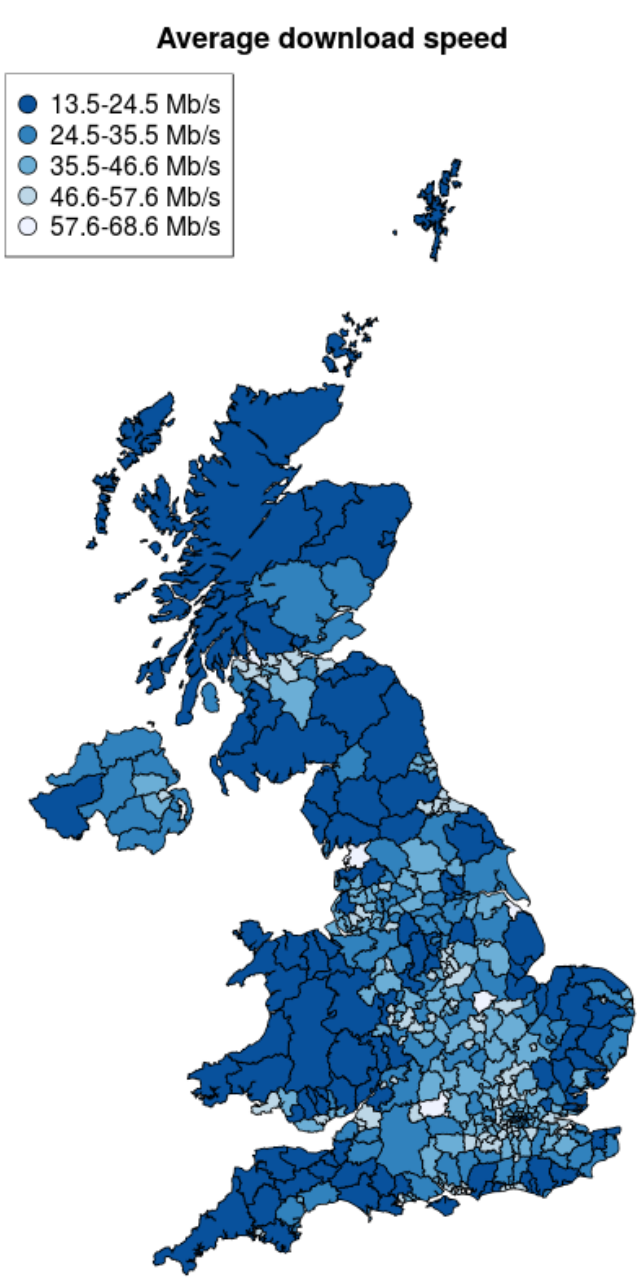
--verbose

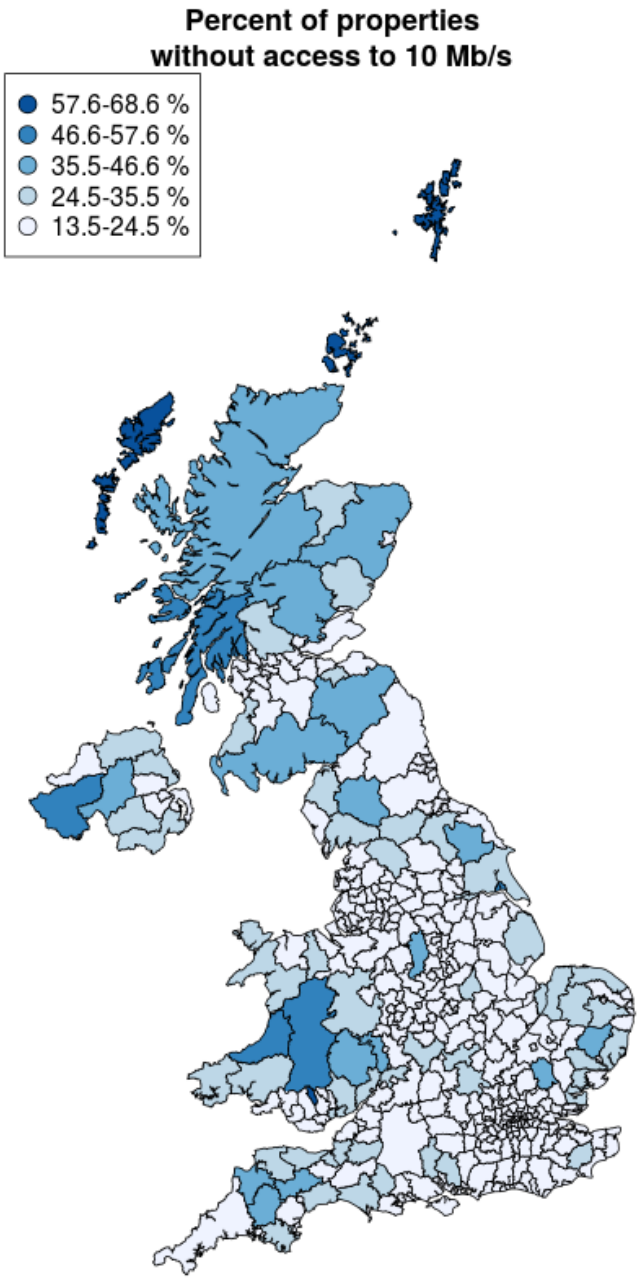


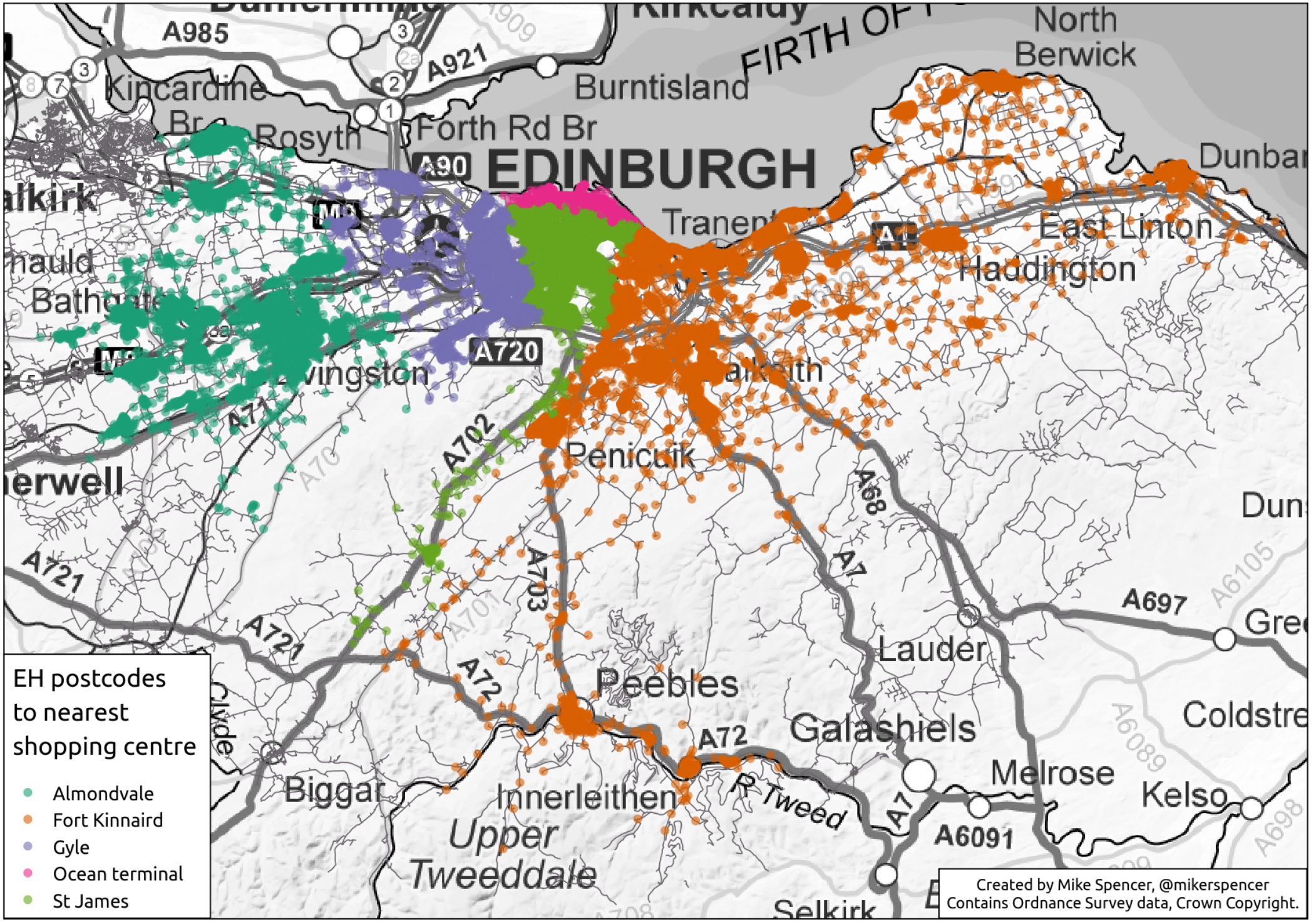
```
# connect schools to streets as layer 2
v.net input=streets_wake points=schools_wake output=streets_net1 \
      operation=connect thresh=400 arc_layer=1 node_layer=2

# connect hospitals to streets as layer 3
v.net input=streets_net1 points=hospitals output=streets_net2 \
      operation=connect thresh=400 arc_layer=1 node_layer=3

# inspect the result
v.category in=streets_net2 op=report

# shortest paths from schools (points in layer 2) to nearest hospitals (points in layer 3)
v.net.distance in=streets_net2 out=schools_to_hospitals flayer=2 to_layer=3

# visualization
g.region vector=streets_wake
d.mon wx0
d.vect streets_wake color=220:220:220
d.vect schools_wake color=green size=10
d.vect map=hospitals icon=basic/cross3 size=15 color=black fcolor=red
d.vect schools_to_hospitals
```

Examples



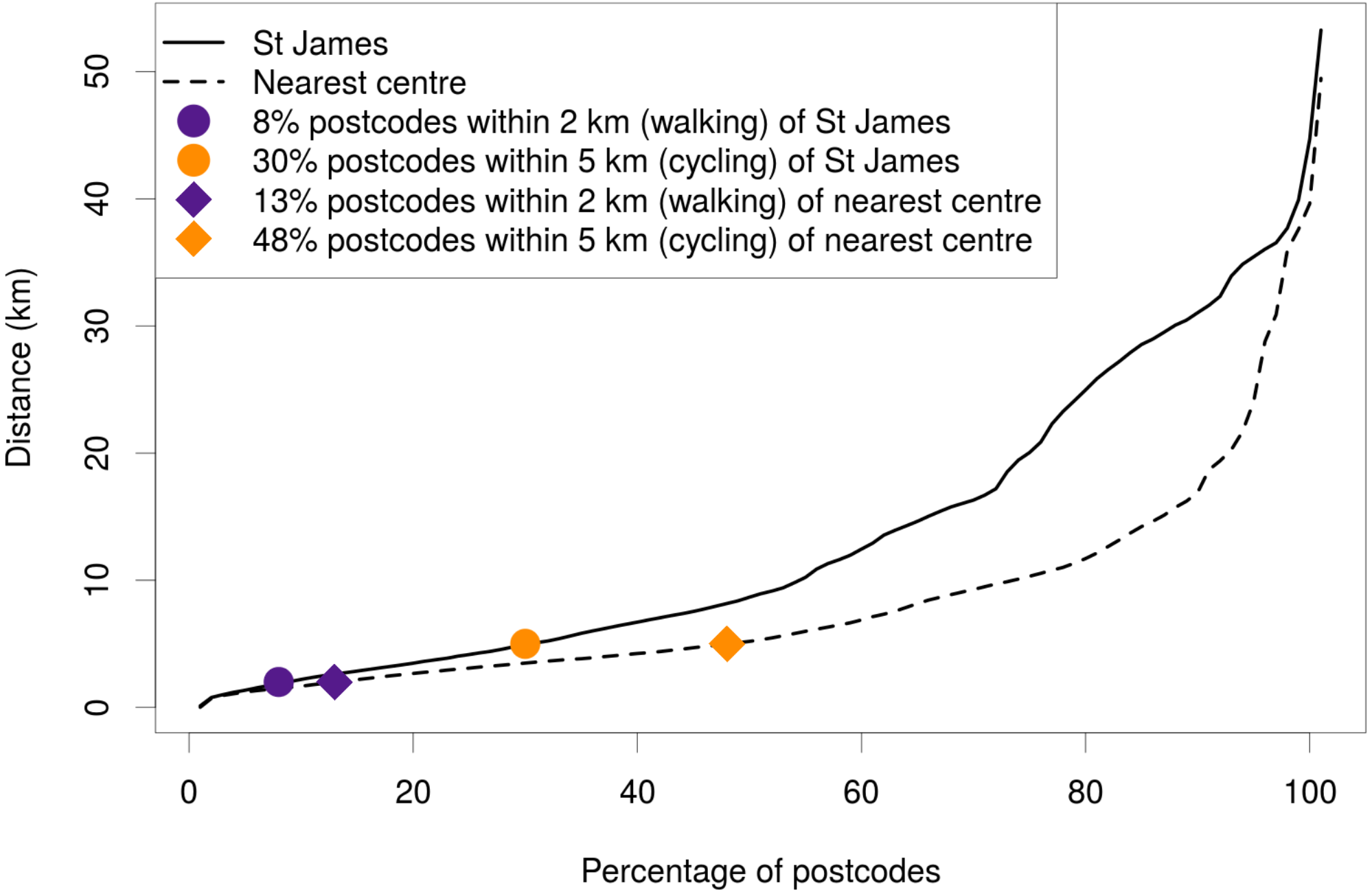




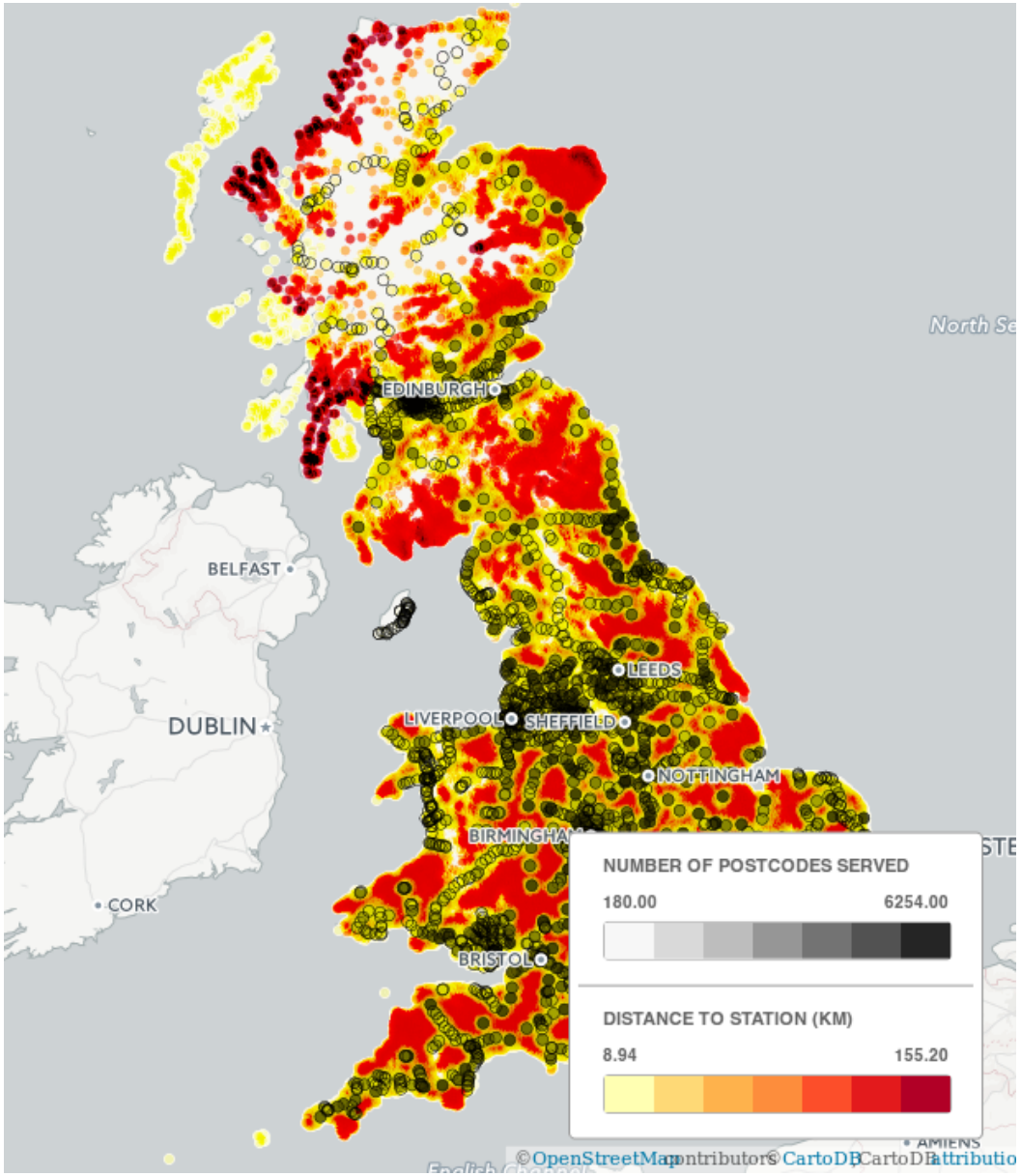


https://scottishsnow.wordpress.com/2017/11/13/spatial-networks-stjames-2/

EH postcode: shortest road distances to EH shopping centres



https://scottishsnow.wordpress.com/2017/11/13/spatial-networks-stjames-2/



https://mikerspencer.carto.com/viz/c7cf448e-2727-11e6-8d2f-0ecd1babdde5/public_map

Great resources

- http://www.datacarpentry.org/R-spatial-raster-vector-lesson/
- http://www.maths.lancs.ac.uk/~rowlings/Teaching/UseR2012/cheatsheet.html
- https://scottishsnow.wordpress.com/
- https://docs.qgis.org/2.18/en/docs/training_manual/
- https://gis.stackexchange.com

- Scottish QGIS UG: http://planet.qgis.org/planet/tag/scotland/
- FOSS4G UK: http://uk.osgeo.org/foss4guk2018/

Mike Spencer

@mikerspencer

scottishsnow.wordpress.com