2018

Self-Organising Maps

Challenging the Scottish Index of Multiple Deprivation (SIMD) with the SOM technique

Author: B125260

Table of Contents

Table of Figures

Figures

Tables

Appendix

1 Introduction

Reducing social disadvantage is of fundamental importance for the positive development of a society and up to now, a plethora of research has attempted to represent deprivation by indices. Understanding the whole phenomenon of social deprivation is evident for efficient policy decision-making (Lucchini & Assi 2013). However, several researchers argue that social deprivation can't adequately be represented by linear indices as it doesn't manifest on a linear scale (Pisati *et al.* 2010, Lucchini & Assi 2013, Lucchini *et al.* 2014).
This paper will continue this research and challenge the linear Scottish Index of Multiple Deprivation (SIMD) with the Self-Organising maps technique, an unsupervised clustering method for data exploration. Patterns of social deprivation in the capital of Scotland – Edinburgh – will be analysed and compared with the SIMD rankings.

2 The Scottish Index of Multiple Deprivation (SIMD)

The Scottish Index of Multiple Deprivation ranks areas in Scotland from the most deprived to the least deprived data zone (see Scottish Government 2016). The SIMD is widely used in Scotland for guiding policy decisions and allocating resources to tackle the most deprived areas. To determine social disadvantage, it ranks each data zone within the seven themes; income, employment, education, health, access to services, crime and housing, then combines them to an overall ranking (Scottish Government 2016). Each theme is composed of one or several variables, overall 26 variables are included.

2.1 Limitations of the SIMD

The SIMD is undoubtedly a valuable guide for targeting deprivation, however, several limitations can be pointed out. Firstly, the weightings attached to themes influence the ranking massively, yet they are not grounded in any theoretical or statistical justification (Deas *et al.* 2003). Secondly, the SIMD provides a relative ranking over the whole country, however, data zones with completely different characteristics (e.g. rural and urban data zones) make comparisons challenging (Fischbacher 2014). Thirdly, the SIMD cannot represent multi-dimensional data as its ranking is linear. To guide interventions, a measure that only describes how disadvantaged an area is compared to other regions, is not sufficient. Moreover, disadvantaged areas should be characterised to determine the kind of intervention needed, which allows for an alternative technique.

3 Self-Organising Maps (SOMs)

Self-Organising Maps (SOMs) are a type of unsupervised Artificial Neural Networks (ANNs). They were developed by Teuvo Kohonen (1982) and are mostly used for clustering, visualisation and data exploration. SOMs reduce n-dimensional data and display it on the two-dimensional map[1] where similar data is placed into the same grid cells, hereinafter referred to as neurons or nodes.

SOMs have been used in a variety of fields, including image analysis (Laaksonen *et al.* 2002), text classification (Merkl 1998) and WEBSOM for internet exploration (Lagus *et al.* 1999), behavioural patterns (Liu *et al.* 2011) and organising collections of music (Risi *et al.* 2007). Recently, attempts have been made to use SOMs for investigating social deprivation (Lucchini *et al.* 2014 for Switzerland, Pisati *et al.* 2010 for Ireland, Lucchini *et al.* 2007 for Switzerland). SOMs have further been described as a valuable addition to conventional indices of social exclusion (Pisati *et al.* 2010).

[1] Note that the word "map" here doesn't refer to geographical space.

4 Overview of the Analysis

The following section will give an overview on the SOM analysis, justify variables, grid size and training parameter used. The workflow can be viewed in Figure 1.

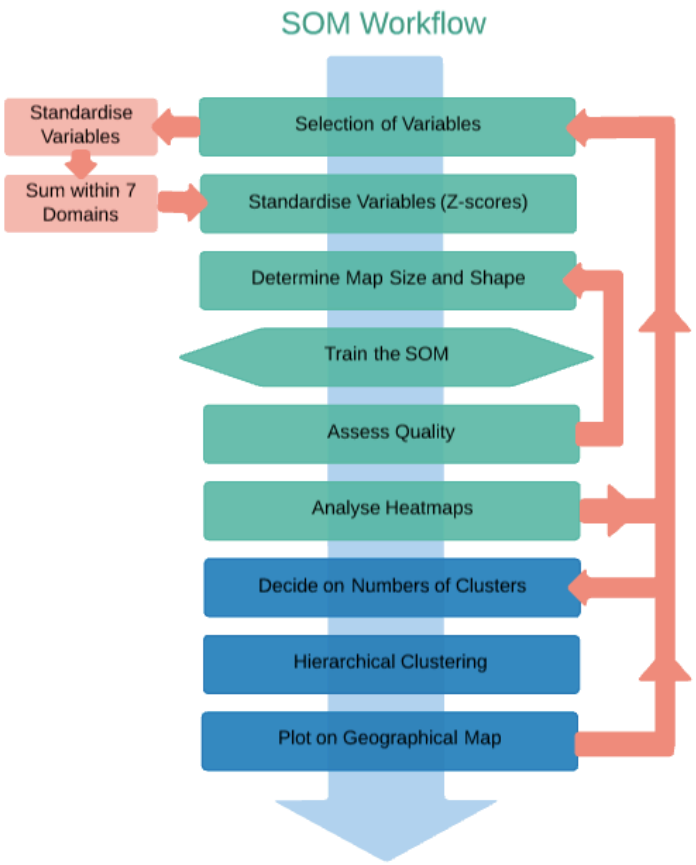


Figure 1: This diagram shows the workflow of this study. SOMs are an explorative technique carried out in iterations to improve the variable selection, the grid size and other parameter.

4.1 Data and Software

The study uses the 2016 SIMD dataset containing 26 deprivation variables (excluding absolute measures), an overall rank and rankings over each of the 7 domains. For an overview of all the variables see Scottish Government (2016). To closely monitor deprivation patterns, the study area was reduced to the 597 data zones in Edinburgh.
For creating the SOMs the "kohonen package" within the software RStudio was used[2]. The code is attached in the Appendix 3.

[2] For a good overview on the "kohonen package" see Wehrens & Buydens (2007).

4.2 Selection of Variables and Data Preparation

The selection of variables was an ongoing process (see Figure 1) and before selecting the final variables different combinations were tested. To make the model as comparable as possible to the SIMD rankings it was decided to use all the 26 variables. However, as many of the variables are correlated within the domains (see Appendix 1) when training the SOM with the 26 variables, domains with more variables (e.g. access has 9 variables) had too much impact on the clustering outcome. For this reason, the variables were summed up within each of the themes.[3] The SOM was then trained with seven variables representing the seven themes; income, employment, health, education, access, crime and housing. For training the SOM, the variables were normalised using Z-Scores, as proposed by Skupin & Agarwal (2008) and Tian *et al.* (2014).

4.3 Definition of Size, Shape and Topology of the SOM Grid

To determine the number of neurons within the SOM grid the formula,

$$Msize = 5 * \sqrt[2]{N}$$

described by Tian *et al.* (2014), was used, with *Msize* as the number of neurons and *N* as the number of observations. With 597 observations, this results in a SOM grid with approximately 122 neurons for this study. Kohonen (2001) suggests an asymmetrical grid to reduce edge effects.[4] This was also empirically tested by Wendel & Buttenfield (2010) who found more edge effects with square shaped SOM grids. The grid used in this study with 9 x 13 neurons fits into Kohonen's (2001) *One-Half rule*, where the shorter side should at least be half of the longer side. This grid size resulted in a distribution of 5 to 10 observations per neuron. A hexagonal lattice was used as it has 6 adjacencies, which is beneficial for SOM training and data visualisation (Kohonen 2001). In addition, a comparison between hexagonal and rectangular lattice was conducted with this data set, whereas the hexagonal lattice showed significantly less edge effects.

[3] Note that only percentage values and relative values (e.g. crime incidents per 10'000 inhabitants) and no absolute values were used. To sum up, within a theme each variable was standardised and some variables were reverted so that high values always represent disadvantage.

[4] Skupin & Agarwal (2008) describe edge effects as compression of observations towards the edges.

4.4 SOM Training

A Gaussian neighbourhood for training the SOM was chosen, based on Wendel & Buttenfield (2010) who encountered best results with a Gaussian kernel. Iterations, learning rate and neighbourhood radius were adjusted so that a plateau in the training process was achieved (see Appendix 2). This study experimented with different values until an overall good quality was achieved. Figure 2 shows that the quality is very good, except within one node on the left edge, which is due to the few very high crime rates. This issue is further described in section 6.3.

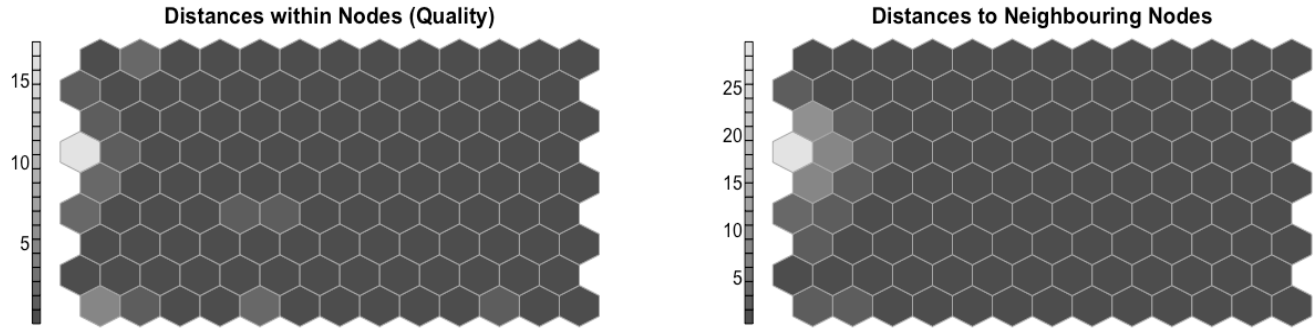


Figure 2: The left diagram shows the quality of the nodes, calculated by mean distance to the codebook vector within a node. Dark values represent low distances (good quality), whilst bright nodes indicated high distances (lower quality). The right diagram, also called U-matrix, displays the distances to neighbouring nodes. It can be used to identify clusters. Again, bright values represent higher distances.

5 Clustering

Clustering within SOMs was firstly conducted by Ritter & Kohonen (1989) developing their "semantic bird maps". Since then, it has become a widely-used technique in the field of SOMs.

In this study, the number of clusters was chosen based on the *Within Clusters Sum of Squares (WCSS*) metric, a rough indicator for the ideal number of clusters. In addition, the fact that clusters are spatially continuous within the SOM and do not display divided clusters or islands, indicates that the clustering was successful (Pisati *et al.* 2010). For this reason, a hierarchical clustering method (see Skupin & Agarwal 2008) with seven clusters was performed. Figure 3 displays the clusters within the component planes. The component planes are useful to understand the clustering and to identify variable relationships and correlations (Koua & Kraak 2005). By means of visual exploration and statistical measures 7-dimensional disadvantage could be evaluated for the seven clusters (Table 1).

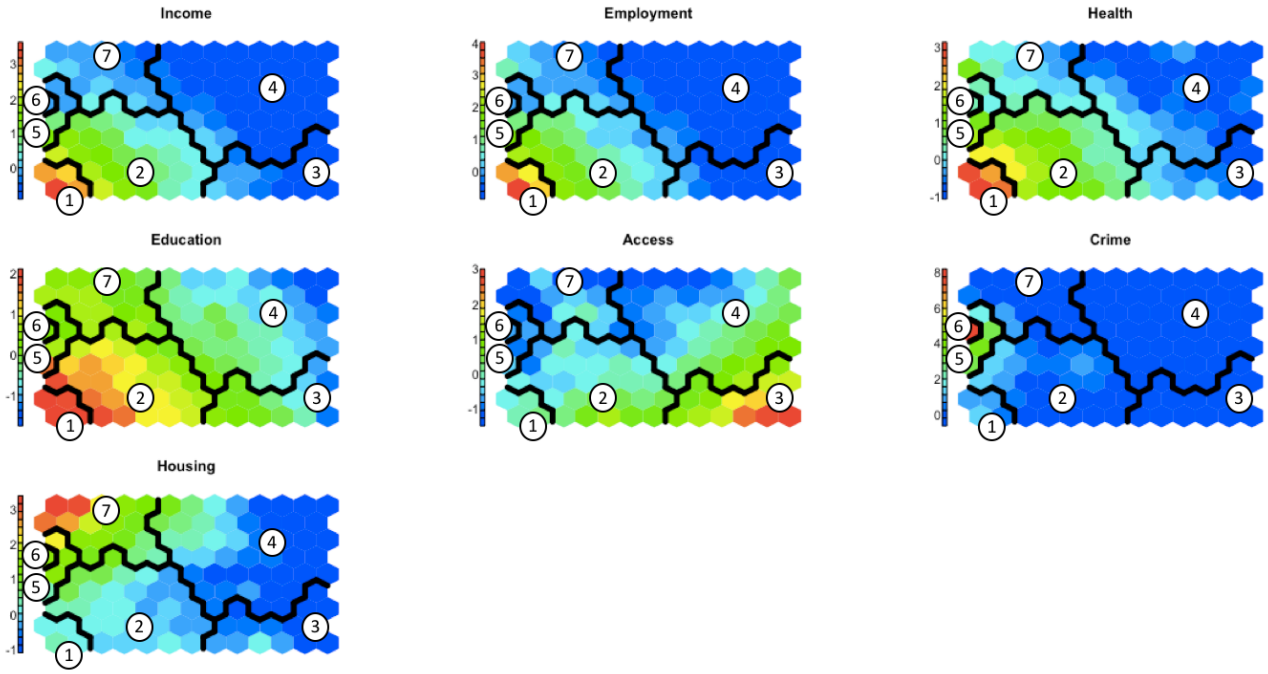


Figure 3: The component planes show the distribution of each variable in the SOM. Red values indicate high disadvantage, whilst blue values represent low disadvantaged. The planes further show visually that income, health, employment and education are correlated. The numbers represent the seven clusters.

	Income	Employment	Health	Education	Access	Crime	Housing
Cluster 1	--	--	--	--	=	=	=
Cluster 2	-	-	-	-	=	=	=
Cluster 3	++	++	++	+	--	++	++
Cluster 4	++	++	++	++	+	++	++
Cluster 5	=	=	=	=	++	-	-
Cluster 6	+	+	=	=	++	--	-
Cluster 7	+	+	=	=	++	+	--

Table 1: Disadvantage of the seven themes per cluster, (--) indicates very high disadvantage, (++) indicates very low disadvantage. The estimations are based on the visual analysis of component planes and statistical queries. The colours represent the cluster colours used in Figure 4 and Figure 5.

5.1 Cluster Interpretation

By means of component planes analysis and display of the results on a geographic map, the clusters were characterised to facilitate the interpretation. Note that the cluster descriptions are not purely based on scientific knowledge, moreover they aim to tell a narrative backed by common knowledge about the city.

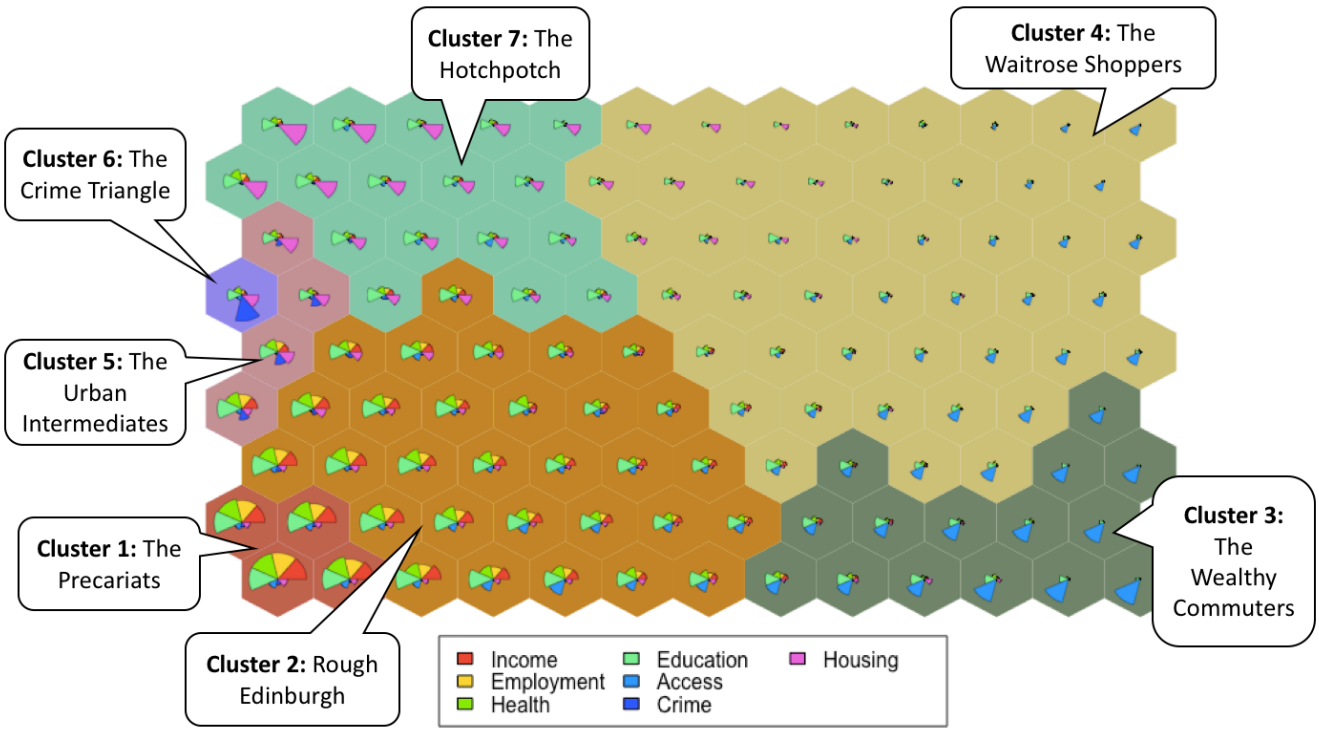


Figure 4: Characteristics of the seven clusters. The node background colours represent the seven clusters. The codes show the seven variable properties for each neuron, with larger symbols indicating higher disadvantage. They are displayed to visualise similarity and differences between adjacent neurons.

Cluster 1 (7%) – The Precariats

The Precariats is a cluster defined by very high disadvantage within the income and employment domain, low education and health issues, whilst access, crime and housing do not seem to be a major issue. The Precariats are the poorest and most deprived cluster.

Cluster 2 (23%) – Rough Edinburgh

The inhabitants of "Rough Edinburgh" share similarities with "The Precariats" but are less vulnerable. However, they still score very low within the domains income, employment, education and health. Typical "Rough Edinburgh" districts are areas with social housing such as Dumbiedykes.

Cluster 3 (13%) – The Wealthy Commuters

The Wealthy Commuters are the cluster most disadvantaged by access – but by choice. Apart from the access domain they show very low disadvantage amongst all the themes. They typically live on the outskirts or in suburban areas where they are house owners and commute to work every day.

Cluster 4 (41%) – The Waitrose Shoppers (Edinburgh Posh)

This cluster is defined by a uniformly low disadvantage across the seven themes. Typical middle class families and professionals with high income and education living in urban and suburban areas belong to this cluster. Areas such as Marchmont and Stockbridge are typical for this cluster.

Cluster 5 (3%) – The Urban Intermediates

This cluster is defined by excellent access and intermediate characteristics amongst the domains income, employment, health and education. It is noticeable that there is a relatively high crime rate and rather low housing conditions.

Cluster 6 (1%) – The Crime Triangle (Edinburgh Nightlife)

The Crime Triangle is defined by very high crime and bad housing. It represents only a very small area in the city centre nearby Princess Street. Data zones within this cluster are not a typical living area but rather an area where people congregate and where young people enjoy the nightlife, which explains the high proportions of crime rates per inhabitants.

Cluster 7 (12%) – The Hotchpotch

The Hotchpotch is characterised by very bad housing and excellent access. It encompasses areas in the city centre with a high proportion of students and presumably flat shares, but also more ethnic areas near to the city centre.

6 Interpretation of Results

6.1 Comparison to the SIMD

Figure 5 displays the SIMD rankings and the SOM clusters on geographical maps. Overall, the SOM clusters show a similar pattern as the SIMD. Data zones within both clusters, “The Precariats” (Cluster 1) and “Rough Edinburgh” (Cluster 2), are ranked very high disadvantaged within the SIMD. However, a few differences can be pointed out:

(1) The SIMD does not show the difference between “The Waitrose Shoppers” (Cluster 4) and “The Wealthy commuters” (Cluster 3), since the SIMD is not very sensitive towards differences in one domain (here access domain)
(2) Very high values within one domain (e.g. “The Crime Triangle”) do not manifest in the linear SIMD rankings. The SIMD rankings are relative and there is no representation of absolute differences between the data zones (see also ISD Scotland & National Services Scotland 2016). In contrary, the SOM technique clusters prominent values together.
(3) The four domains – income, employment, health and education – are highly correlated (see Figure 3) and represent 84% of the SIMD weightings (Scottish Government 2016). This results in that areas with bad scores in the three other domains are not highlighted within the SIMD. One advantage of the SOM technique is that correlations of variables can be easily detected visually using component planes (Pölzlbauer *et al.* 2005).

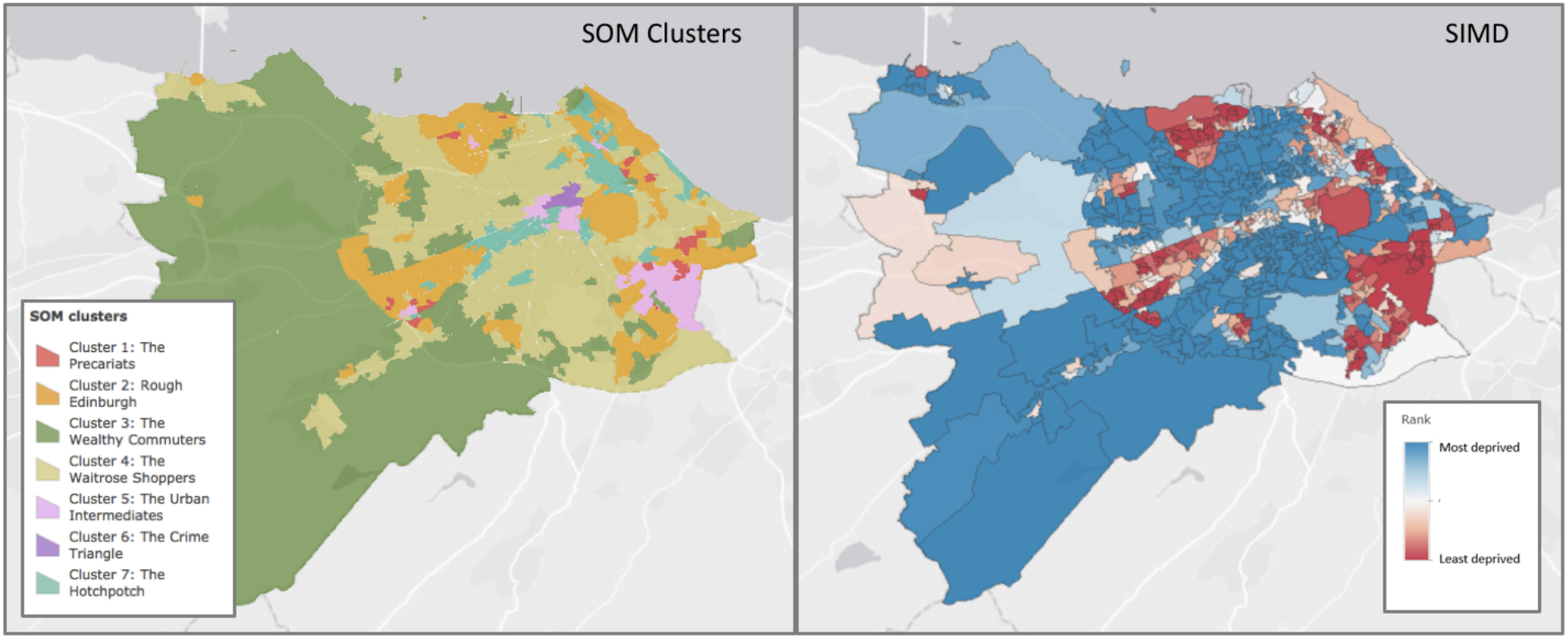


Figure 5: SOM clusters and SIMD rankings displayed on geographic maps.

6.2 Identified SIMD Data Limitations

During the SOM process several data limitations of the SIMD data were discovered. The housing domain composed out of (1) overcrowded living and (2) no central heating, does not seem to represent disadvantaged housing adequately. Areas with bad housing show a mix of student areas where young people live in flat shares and ethnic areas (Cluster 7, The Hotchpotch). House and flat prices could be a better measure for disadvantaged housing. Moreover, measuring crime incident per inhabitants seems to be a problematic approach in city centres where only few people live but many people congregate. To be aware of these data limitations is essential as variables and data composing an index are key to gather valuable outcome (Lucchini & Assi 2013).

6.3 SOM Limitations

The SOM bypasses the weighting problem of the SIMD, however, the SOM technique is very sensitive towards the characteristic of presented variables. For instance, the crime rate data for this study contains a few extremely high values which have a high impact on the clustering outcome. When the crime data was log-transferred to approximate a normal distribution, the SOM showed a completely different clustering. Another limitation of the SOM technique is the potentially inconsistent solutions and the reliability on the interpreter. Clustering outcome and cluster interpretation can be different between runs and different interpreters. However, it can be argued, that this represents the reality more adequately. If possible, the best practice is to train the SOM several times and to work in teams to discuss outcomes. Finally, the plain SOM technique has limited applications, it is best to be used in conjunction with other visualisation techniques such as geographic maps (Skupin & Agarwal 2008).

7 Conclusion

As described above, SOMs offers insights that can't be explored or displayed with the linear SIMD ranking. The technique can provide a different picture on social deprivation for allocating resources. However, SOMs show several drawbacks such as that there is a heavy reliance on the interpreter. Using the SOMs and the SIMD ranking complementary offers a great opportunity to explore and understand the whole phenomenon of social deprivation. Overall, it can be said that the two methods pursue different purposes. Whilst the SIMD gives a measure of how disadvantaged an area is based on the chosen variables, the SOM technique can be used to get to know the variables and the areas of a region. In short, the SIMD assesses, the SOM method offers a platform to tell a story.

Word count (without captions, tables and sources): 1'991

Sources

DEAS, I., ROBSON, B., WONG, C. & BRADFORD, M. 2003. Measuring Neighbourhood Deprivation: A Critique of the Index of Multiple Deprivation. *Environment and Planning C: Government and Policy*, **21**, 883–903, 10.1068/c0240.

FISCHBACHER, C.M. 2014. Identifying "deprived individuals": are there better alternatives to the Scottish Index of Multiple Deprivation (SIMD) for socioeconomic targeting in individually based programmes addressing health inequalities in Scotland. *Glasgow: Scottish Public Health Observatory*.

ISD SCOTLAND & NATIONAL SERVICES SCOTLAND. 2016. The Scottish Index of Multiple Deprivation (SIMD) Available at: http://www.isdscotland.org/Products-and-Services/GPD-Support/Deprivation/SIMD/ [Accessed April 2, 2018].

KOHONEN, T. 1982. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, **43**, 59–69, 10.1007/BF00337288.

KOHONEN, T. 2001. *Self-Organizing Maps*. 3rd ed. Berlin Heidelberg: Springer-Verlag Available at: //www.springer.com/gb/book/9783540679219 [Accessed March 29, 2018].

KOUA, E.L. & KRAAK, M.-J. 2005. Integrating computational and visual analysis for the exploration of health statistics. *In Developments in Spatial Data Handling*. Springer, Berlin, Heidelberg, 653–664., 10.1007/3-540-26772-7_49.

LAAKSONEN, J., KOSKELA, M. & OJA, E. 2002. PicSOM-self-organizing Image Retrieval with MPEG-7 Content Descriptors. *Trans. Neur. Netw.*, **13**, 841–853, 10.1109/TNN.2002.1021885.

LAGUS, K., HONKELA, T., KASKI, S. & KOHONEN, T. 1999. Websom for Textual Data Mining. *Artificial Intelligence Review*, **13**, 345–364, 10.1023/A:1006586221250.

LIU, Y., LEE, S.-H. & CHON, T.-S. 2011. Analysis of behavioral changes of zebrafish (Danio rerio) in response to formaldehyde using Self-organizing map and a hidden Markov model. *Ecological Modelling*, **222**, 2191–2201, 10.1016/j.ecolmodel.2011.02.010.

LUCCHINI, M. & ASSI, J. 2013. Mapping Patterns of Multiple Deprivation and Well-Being using Self-Organizing Maps: An Application to Swiss Household Panel Data. *Social Indicators Research*, **112**, 129–149.

LUCCHINI, M., BUTTI, C., ASSI, J., SPINI, D. & BERNARDI, L. 2014. Multidimensional Deprivation in Contemporary Switzerland Across Social Groups and Time. *Sociological Research Online*, **19**, 3.

LUCCHINI, M., PISATI, M. & SCHIZZEROTTO, A. 2007. Stati di deprivazione e di benessere nell'Italia contemporanea. Un'analisi multidimensionale.

MERKL, D. 1998. Text classification with self-organizing maps: Some lessons learned. *Neurocomputing*, **21**, 61–77, 10.1016/S0925-2312(98)00032-0.

PISATI, M., WHELAN, C.T., LUCCHINI, M. & MAÎTRE, B. 2010. Mapping patterns of multiple deprivation using self-organising maps: An application to EU-SILC data for Ireland. *Social Science Research*, **39**, 405–418, 10.1016/j.ssresearch.2009.11.004.

PÖLZLBAUER, G., DITTENBACH, M. & RAUBER, A. 2005. *Gradient Visualization of Grouped Component Planes on the SOM lattice*.

RISI, S., MÖRCHEN, F., ULTSCH, A. & LEWARK, P. 2007. Visual mining in music collections with emergent SOM. *In Proceedings Workshop on Self-Organizing Maps (WSOM)*.

RITTER, H. & KOHONEN, T. 1989. Self-organizing semantic maps. *Biological Cybernetics*, **61**, 241–254, 10.1007/BF00203171.

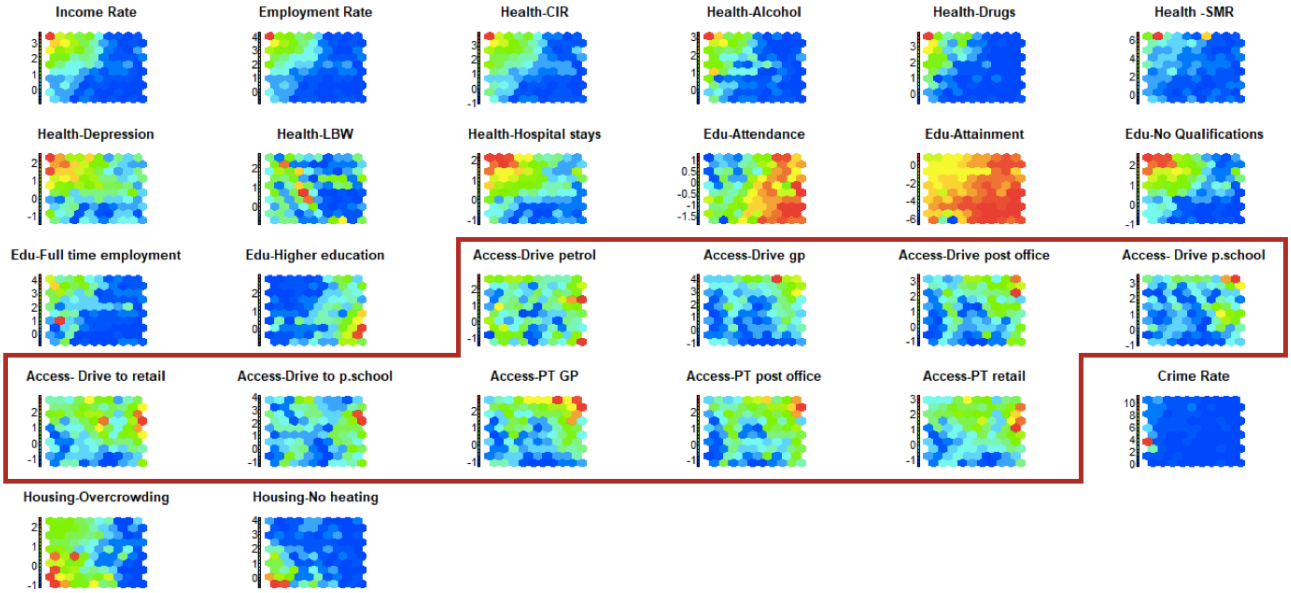
SKUPIN, A. & AGARWAL, P. 2008. Introduction: What is a Self-Organizing Map? *In* Agarwal, P. & Skupin, A., *eds. Self-Organising Maps*. Chichester, UK: John Wiley & Sons, Ltd, 1–20., 10.1002/9780470021699.ch1.

TIAN, J., AZARIAN, M.H. & PECHT, M. 2014. Anomaly Detection Using Self-Organizing Maps-Based K-Nearest Neighbor Algorithm. *In Proceedings of the European Conference of the Prognostics and Health Management Society*. Citeseer.

WEHRENS, R. & BUYDENS, L.M.C. 2007. Self- and Super-organizing Maps in R: The kohonen Package. *Journal of Statistical Software*, **21** Available at: https://econpapers.repec.org/article/jssjstsof/v_3a021_3ai05.htm.

WENDEL, J. & BUTTENFIELD, B.P. 2010. Formalizing Guidelines for Building Meaningful Self-Organizing Maps in GIScience 2010—Sixth International Conference on Geographic Information Science. *Zurich, Switzerland*, 6.

Appendix



Appendix 1: Component planes when running all the 26 variables within one SOM. Red values represent high social disadvantage, whilst blue values describe low disadvantage. Correlations between the variables can be visually detected. The red box captures the nine access variables.

Appendix 2: Training progress. The mean distance to the closest unit should reach a minimum plateau. Once the plateau has been reached after approximately 450 iterations the SOM has stabilised and continuing iterations don't not improve the quality significantly.

```r
# Set working directory
setwd("/Users/livia/R/soms")

library(kohonen)
library(ggplot2)
library(rgdal)
library(rgeos)
library(gridExtra)
library(grid)


#read in the boundary data for the edinburgh area, already matched up by row with the census data
edinburgh_map <- readOGR(dsn="AssessmentData/SG_SIMD_2016_EDINBURGH.shp", layer="SG_SIMD_2016_EDINBURGH")

#######CONVERT PROJECTION TO LAT LONG
#convert the object into latitude and longitude for easier use with ggmap
edinburgh_map <- spTransform(edinburgh_map, CRS("+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs"))


#convert to data frame
edin_data <- as.data.frame(edinburgh_map)
#rename
names(edin_data) <- c("Datazone", "Area Name","Total Population","Working Age Population","SIMD Domain Rank","Quantile SIMD Rank","Decile SIMD Rank","Vigintile SIMD Rank","Percentile SIMD Ranking","Income Rate","Income Count","Income Domain Rank","Employment Rate","Employment Count","Employment Domain Ranking","Comparative Illness Factor","Hospital-Alcohol","Hospital-Drugs","Standard Mortality Rate","Depression","Low Birth Weight","Emergancy stays in hospital","Health Domain Ranking","Education Attendance","Education Attainment", "Education- No Qualifications","Education-In full time employment or ed.","Education-Entering higher education", "Education Domain Rank","Access-Drive to petrol", "Access- Drive to gp","Access- Drive to post office","Access- Drive to primary school","Access- Drive to retail","Access-Drive to secondary school","Access-Public transport to GP","Access-Public transport to post office","Access-Public transport to retail","Geographical Access Domain Ranking","Crime Count","Crime Rate","Crime Domain Rank", "Housing- Overcrowded count","Housing-No central heating count","Housing - Overcrowding rate","Housing- No central heating rate","Housing Domain Ranking","Shape Length","Shape Area","Intermediary Name" )

##FORTIFY
#convert spatial polygon to dataframe including columns of spatial information
edinburgh_fort <- fortify(edinburgh_map, region= "DataZone")
#merge the new dataframe with the edinburgh simd data using their shared column
edinburgh_fort <- merge(edinburgh_fort, edin_data, by.x="id", by.y="DataZone")

##PREPARE HEALTH DATA
health <- edin_data[, c(16,17,18,19,20,21,22)]
## standardise (between 0 and 1)
health_st<-apply(health, MARGIN = 2, FUN=function(X) (X-min(X))/diff(range(X))) #Margin=2 re-scales column wise
## sum standardised values
health_sum<-rowSums(health_st[,c(1,2,3,4,5,6,7)])
health_sum

##PREPARE ACCESS DATA
access <- edin_data[, c(30,31,32,33,34,35,36,37,38)]
access_st<-apply(access, MARGIN = 2, FUN=function(X) (X-min(X))/diff(range(X)))
access_sum <- rowSums(access_st[,c(1,2,3,4,5,6,7,8,9)]) #sum the colums

##PREPARE EDUCATION DATA
education <- edin_data[, c(24,25,26,27,28)]
education_st1 <- apply(education[,c(1,2,5)], MARGIN = 2, FUN=function(X) (max(X)-X)/diff(range(X))) #invert
```

```r
education_st2 <- apply(education[,c(3,4)], MARGIN = 2, FUN=function(X) (X-min(X))/diff(range(X)))
education_sum <- rowSums(cbind(education_st1, education_st2))
#check if standardisation was right
education_check<-cbind(education_st1, education_st2, education_sum,edin_data[,29])

##PREPARE HOUSING DATA
housing <- edin_data[, c(45,46)]
housing_st <-apply(housing, MARGIN = 2, FUN=function(X) (X-min(X))/diff(range(X)))
housing_sum <- rowSums(housing_st[,c(1,2)])

#INCOME
income <- edin_data[,10]

#EMPLOYMENT
employment <- edin_data[,13]

#CRIME
crime <- edin_data[,41]

##Bringing all the variables into a data frame
summed_vars<-as.data.frame(matrix(c(income,employment,health_sum,education_sum,access_sum,crime,housing_sum),nrow=length(access_sum)))
names(summed_vars) <- c("Income", "Employment", "Health","Education", "Access", "Crime", "Housing")


########## SOM TRAINING

#choose the variables with which to train the SOM by subsetting the dataframe called data
# this was used to create Appendix 1
#data_train <- edin_data[, c(10,13,16,17,18,19,20,21,22,24,25,26,27,28,30,31,32,33,34,35,36,37,38,41,45,46)]

#standardise the data creating z-scores and convert to a matrix
data_train_matrix <- as.matrix(scale(summed_vars))
#keep the column names of data_train as names in new matrix
names(data_train_matrix) <- names(summed_vars)

########## SOM GRID
#define the size, shape and topology of the som grid
som_grid <- somgrid(xdim = 13, ydim=9, topo="hexagonal", neighbourhood.fct="gaussian")

########## TRAIN
# Train the SOM model, alpha is learning rate, rlen is number of iterations
som_model <- som(data_train_matrix,
                 grid=som_grid,
                 rlen=500,
                 alpha=c(0.1,0.01),
                 keep.data = TRUE )

# Plot of the training progress - how the node distances have stabilised over time.
# mean distance to closes codebook vector during training
plot(som_model, type = "changes")

## load custom palette, created by Shane Lynn
source('coolBlueHotRed.R')

###PLOT COUNT
#counts within nodes
plot(som_model, type = "counts", main="Node Counts per Node", palette.name=coolBlueHotRed, shape="straight", border="transparent")

```

```
###CREATE PLOTS OF QUALITY AND NEIGHBOUR DISTANCES
par(mfrow = c(1,2)) #create both plots next to each other
#map quality
plot(som_model, type = "quality", main="Distances within Nodes (Quality)", pal-
ette.name=grey.colors, shape="straight", border="darkgrey")
#neighbour distances
plot(som_model, type="dist.neighbours", main = "Distances to Neighbouring Nodes",
shape="straight", palette.name=grey.colors, border="darkgrey")

dev.off() #mfrow off

#code spread, plot codebook vectors
plot(som_model, main="Codebook Vectors", type = "codes", shape="straight", bgcol="lightgrey",
palette.name=rainbow, border="darkgrey")

#plot codebook vectors of crime
plot(som_model, type ="property", property=getCodes(som_model)[,6], main="Crime", pal-
ette.name=coolBlueHotRed,shape="straight", border="transparent")

##plot all the component planes into the same image
par(mfrow = c(3,3)) # 3 x 3 grid
for (i in 1:7) { # loop through all of them and plot
  plot(som_model, type = "property", property = getCodes(som_model)[,i],
       main=colnames(getCodes(som_model))[i], palette.name=coolBlueHotRed, shape="straight",
border="transparent")
  add.cluster.boundaries(som_model, som_cluster)
  }
dev.off()

#reset margins
par(mar=c(5,5,4,2))

# show the WCSS metric for kmeans for different clustering sizes.
# Can be used as a "rough" indicator of the ideal number of clusters
mydata <- getCodes(som_model) #extract codebook vectors
wss <- (nrow(mydata)-1)*sum(apply(mydata,2,var))
#calculate sums of squares for 2-15 clusters
for (i in 2:15) wss[i] <- sum(kmeans(mydata,
                                     centers=i)$withinss)
#plot wcss
plot(1:15, wss, type="b", xlab="Number of Clusters",
     ylab="Within groups sum of squares", main="Within cluster sum of squares (WCSS)")


############################### CLUSTERS
# Form clusters on grid
## use hierarchical clustering to cluster the codebook vectors
som_cluster <- cutree(hclust(dist(getCodes(som_model))), 7)

# Colour palette definition
class_colors <- c("coral3","orange3", "darkseagreen4", "khaki3", "lightpink3","mediumpurple2",
"aquamarine3","peru")

# plot codes with cluster colours as background
plot(som_model, type="codes", bgcol = class_colors[som_cluster], main = "Clusters",
shape="straight", palette.name=rainbow, border="transparent")
add.cluster.boundaries(som_model, som_cluster)


#######CLUSTER VARIABLE ANALYSIS
cluster<-data.frame(id=edin_data$DataZone, cluster=som_cluster[som_model$unit.classif],
data_train_matrix)
agg<-aggregate(cluster[,3:9], list(cluster$cluster), mean)
```

```
160
161 result<-c()#empty variables
162 #create same colours as in som maps
163 colour_scheme <- c("#FF0000FF", "#FFDB00FF", "#49FF00FF", "#00FF92FF", "#0092FFFF", "#4900FFFF",
"#FF00DBFF")
164
165 for (i in 2:8) {
166   aggtemp<-agg[,c(1,i)]
167   aggtemp["Variable"]=names(agg)[i]
168   names(aggtemp)<-c("Cluster","Value")
169   result<-rbind(result,aggtemp)
170 }
171 names(result)<-c("Cluster","Value", "Variable")
172
173 ##PLOT GRAPH
174 par(mar=c(5,5,4,2), bg="grey99")
175 plot(result[,1], result[,2], main = "Eruptions of Old Faithful", xlab = "Cluster", ylab = "Z-
Scores", col=colour_scheme[factor(result$Variable)],type="p", pch = 16)
176 rect(par("usr")[1], par("usr")[3], par("usr")[2], par("usr")[4], col =
177        "grey90") #plot a grey rect
178 points(result[,1], result[,2], col=colour_scheme, type="p", pch = 16) #plot points over rect
179 # Add a legend
180 par(bg="white")
181 legend(5, 11, legend=names(cluster)[3:9],
182        col=colour_scheme, pch = 16, horiz=FALSE, cex=0.8)
183
184
185 ##### MAKE GEOGRAPHIC MAP
186
187 #create dataframe of the small area id and of the cluster unit
188 cluster_details <- data.frame(id=edin_data$DataZone, cluster=som_cluster[som_model$unit.clas-
sif])
189
190 #we can just merge our cluster details onto the fortified spatial polygon dataframe we created
earlier
191 mappoints <- merge(edinburgh_fort, cluster_details, by.x="id", by.y="id")
192
193 # Finally map the areas and colour by cluster
194 ggplot(data=mappoints, aes(x=long, y=lat, group=group, fill=factor(cluster))) +
195   geom_polygon(colour="transparent")  +
196   coord_equal() +
197   scale_fill_manual(values = class_colors)
198
199
200 # combine map with cluster details
201 ed_map <- merge(edinburgh_map, cluster_details, by.x="DataZone", by.y="id")
202
203 # save as an esri shapefile
204 writeOGR(obj=ed_map, dsn="edinburgh_map_clustered_fin", layer="edinburgh_map_clustered2",
driver="ESRI Shapefile")
```

Appendix 3: R code.