Spatial Analysis Techniques in R

Lesson 3: Area objects, spatial autocorrelation and local statistics

3.1 Introduction

In this lesson we look at methods for handling so-called 'lattice data' relating to natural or arbitrary regions of the earth's surface. Such data occur fairly frequently and in a number of different guises. Task 3.1 takes you to some readings that explore some of the major issues:

> ### Task 3.1: What have we learnt so far about lattices?
>
> 1) Go to O'Sullivan and Unwin (2010) and read Section 7.2 Types of Area Object, pages 188-191 (The First Edition has similar materials, also Section 7.2, pages 169-172)
>
> 2) Spatial autocorrelation and the MAUP problem, the ecological fallacy, Section 2.2, pages 34-39 (Pages 28-33 of First Edition)
>
> 3) Section 3.7, pages 72-79 Mapping and Exploring Areas
>
> It will help if at this point you install the spdep package from your chosen CRAN mirror.
>
> Understandably Brunsdon and Comber (2005) make use of Chris Brunsdon's own GISTiils package. Section 3.3 pages 59-71 and most of Chapter 7, pades 218-235 cover much of the same ground.

In this lesson we will explore:

- Methods by which some statistical ideas implemented in the R environment can help produce more sensible choropleth maps of these data;
- The detection and evaluation of global spatial autocorrelation;
- The important concept of the geographic structure matrix **W** as an hypothesis about any spatial structure;
- The idea of local statistics, as illustrated by the local version of Moran's *I*; and
- The fairly recent development of geographically weighted regression and related local statistics.

Just as in Lesson 2 we made use of the spatstat package, in this lesson use will be made of the spdep (**sp**atial **dep**endence) package which you should ensure you have acquired from your CRAN mirror site using the packages drop down menu in the R-GUI.

3.2 Displaying lattice data: can theory help?

The short answer is 'yes'. To illustrate alternative approaches to the display of lattice data, we will use some famous data relating to the incidence of a form of lip cancer in Scotland from work by Clayton and Kaldor (1987). These same data were used in their books by Waller and Gotway (2004) and Banerjee, Carlin and Gelfand (2004). Epidemiologists see this disease as being most common in males, in rural areas, amongst people who work outdoors and (independently) with smoking.

My data for this section of the lesson are the number of male lip-cancer cases in the lattice given by the 53 Districts and 3 Island Authorities of Scotland for a six year period from 1975-1980 (coded as CANCER) together with some possible covariates, the number of male population-years-at-risk (POP), an age- and population-standardized expected number of cases (CEXP), and the percentage of the population working in agriculture, fisheries and food (i.e. a surrogate variable for 'outdoors', coded as AFF). The latter two variables are from Lawson *et al.* (1999). These same data are used in Cressie (1993, page 535 *et seq*.) and my source was a *shapefile* (recall Lesson 1) prepared by Dr. Luc Anselin and downloaded from:

http://geodatacenter.asu.edu (checked 02-12-15)

Note that, as often happens with lattice data for irregular polygons, and especially those with 'island' outliers such as this one of Scotland, Dr Anselin had to work hard to get them into .shp format. The data he collated originally came from a file *Scotland.map* included with the *WinBugs* package, which was then exported to *S-Plus* format and edited to bring it into his *GeoDa*™ input format, eliminate duplicate coordinates, remove so-called 'sliver' polygons, and represent islands with multiple polygons by a single polygon. The result was output as a clean .shp file that can easily be incorporated into the R-environment. In addition to projected (*x*, *y*) co-ordinates there are nine other fields:

Variable	Description
CODENO	Code converted to numeric (drop w prefix)
RECORD_ID	Unique ID
DISTRICT	District number 1-56
NAME	Name of districts from Cressie
CODE	District code from WinBugs
CANCER	Lip cancer cases from Cressie
POP	Population years at risk from Cressie

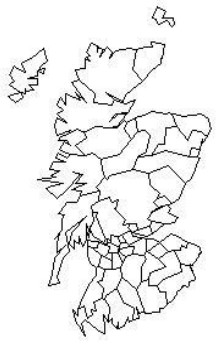
Variable	Description
CEXP	Expected cases from Lawson et al.
AFF	Outdoor industry from Lawson et al.

The data are accessed using the full path name, providing a key field and a coordinate reference system using readShapePoly in the maptools package and then the lattice outlines are plotted:

```
> library(maptools)
> library(sp)
> lips <- readShapePoly ("C:\\Users\\David\\Desktop\\scotlip\\scotlip",
IDvar="RECORD_ID",proj4string=CRS(as.character(NA)))
```

Note that on my machine the scotlip shapefile is in a folder of the same name. You don't need all of this for this exercise, but I've put it in to remind you that you might.

```
> plot(lips)
```



As usual plot knows about shapefiles and maps the area outlines correctly (or at least insofar as the projection used will allow this. No co-ordinate reference system is defined but I am sure that the projection used to record the (*x*, *y*) data is the UK Ordnance Survey, OSGB from the 1930s). It is useful at this point to have a look at the spatial polygons data frame called lips that we have just created to see how the .shp format is defined and the set of information that is needed:

```
> lips
An object of class "SpatialPolygonsDataFrame"
Slot "data":
  CODENO       AREA PERIMETER RECORD_ID DISTRICT          NAME  CODE
CANCER
1   6126  974002000  184951.0         1        1 Skye-Lochalsh w6126      9
```

```
2    6016 1461990000  178224.0        2       2  Banff-Buchan w6016     39
3    6121 1753090000  179177.0        3       3     Caithness w6121     11
Etc
56   5808 1597940000  172514.0       56      56     Annandale w5808      0
      POP  CEXP AFF
1    28324  1.38  16
2   231337  8.66  16
3    83190  3.04  10
Etc
56  103412  1.76  10
```

For example, the border of region 1 is coded in the file as:

```
Slot "polygons":
[[1]]
An object of class "Polygons"
Slot "Polygons":
[[1]]
An object of class "Polygon"
Slot "labpt":
[1] 197678.1 824339.8
Slot "area":
[1] 974001724
Slot "hole":
[1] FALSE
Slot "ringDir":
[1] 1
Slot "coords":
         [,1]     [,2]
 [1,] 214091.9 841215.2
 [2,] 218829.0 831090.0
 [3,] 217605.7 830865.8
 [4,] 201864.0 818968.0
 [5,] 216426.0 805179.0
 [6,] 192646.0 807178.0
 [7,] 187948.5 807454.3
 [8,] 176840.0 814731.0
 [9,] 183258.0 824522.0
[10,] 194413.0 820074.0
[11,] 187966.0 826737.0
[12,] 175233.0 827294.0
[13,] 179122.0 832281.0
[14,] 189365.0 836048.0
[15,] 205306.0 844697.0
[16,] 211804.0 840980.0
[17,] 214091.9 841215.2
```

etc

```
Slot "bbox":
     min     max
x  95631  454570
y 530297 1203008
Slot "proj4string":
CRS arguments: NA
```

We can of course also view the thematic data as a standard data frame which gives access to all the standard methods in R:

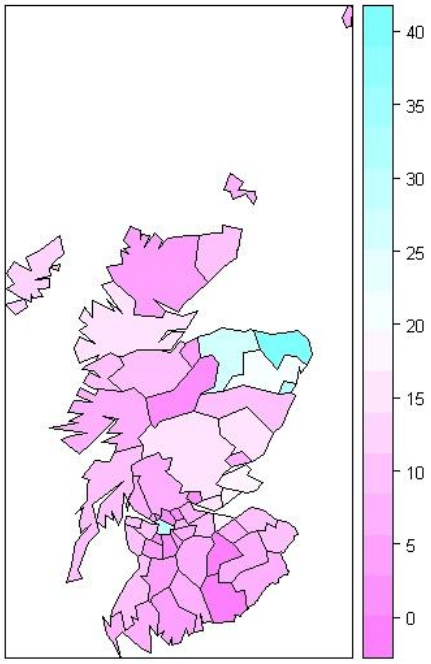
```
> as(lips,"data.frame")
```

```
  CODENO      AREA       PERIMETER RECORD_ID DISTRICT              NAME
                CODE CANCER

1    6126       974002000    184951.0       1                1                 Skye-
Lochalsh        w6126      9
2    6016       1461990000   178224.0       2                2                 Banff-
Buchan                  w6016     39
3    6121       1753090000   179177.0       3                3
          Caithness                 w6121     11
etc
56   5808       1597940000   172514.0      56                56
          Annandale                 w5808      0
      POP       CEXP  AFF
1    28324       1.38     16
2   231337       8.66     16
etc
```

A plot of the cancer data using spplot in the package sp is :

```
> spplot(lips,"CANCER")
```



This uses a standard color palette that looks correct when copied in black and white and a standard class interval scheme. I find this quite pleasing to look at, but this map not only isn't very useful, it is positively misleading and for the obvious reason that the highest district totals are in Moray (265, with POP=245513) Banff Buchan (39 with POP= 231,337), Aberdeen (22 with POP= 583,327) and Glasgow (28, POP=2,315,353) all of which have high number of male population years at risk (POP). (I know Scotland fairly well, but I had to consult an atlas at this point!). In fact, we might be tempted to conclude that the cancer is most prevalent in the urban areas of the country such as Aberdeen and Glasgow. So it is if all you are interested in is the absolute number of cases, but as a proportion of the at risk population we get a different picture.

As shown below, a simple standardization taking the ratio of observed cases to this base population expressed as a rate per thousand male population years at risk results in a very different picture:

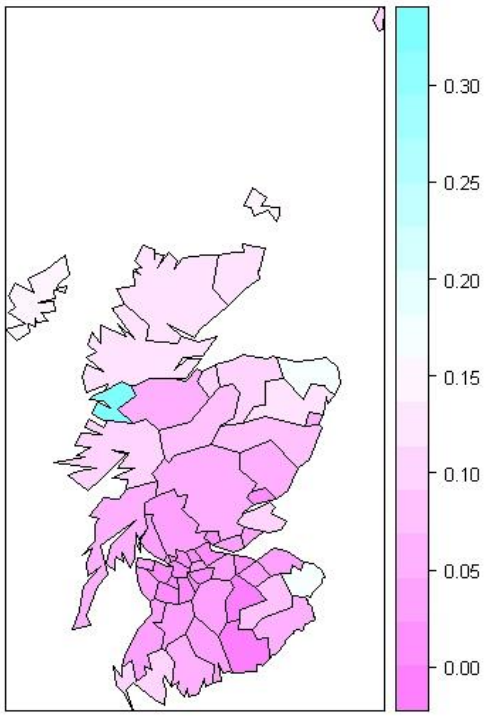
```
>rate <- (lips$CANCER/lips$POP) * 1000
> rate

 [1] 0.317751730 0.168585224 0.132227431 0.174047573 0.116035306 0.150378767
 [7] 0.105900706 0.111815728 0.101380464 0.120806504 0.148038490 0.133258708
[13] 0.102131136 0.092545463 0.091658040 0.080598218 0.073868883 0.074353391
[19] 0.055259813 0.068161680 0.060789119 0.053143434 0.057647157 0.042730347
[25] 0.043968047 0.039583476 0.042760365 0.043255402 0.046237296 0.028742991
[31] 0.035932963 0.045837917 0.028037346 0.034316354 0.034448634 0.030380977
[37] 0.028096130 0.025072711 0.025948527 0.025490046 0.016196614 0.018756492
```

[43] 0.014154883 0.013356158 0.014756582 0.012596045 0.008102119 0.009612211
[49] 0.012087968 0.010968601 0.005580544 0.009032852 0.006844065 0.004052783
[55] 0.000000000 0.000000000

We need to bind these data back into the spatial data frame before we can plot them:

```
> lips<-spCbind(lips,rate)
> spplot(lips,"rate")
```



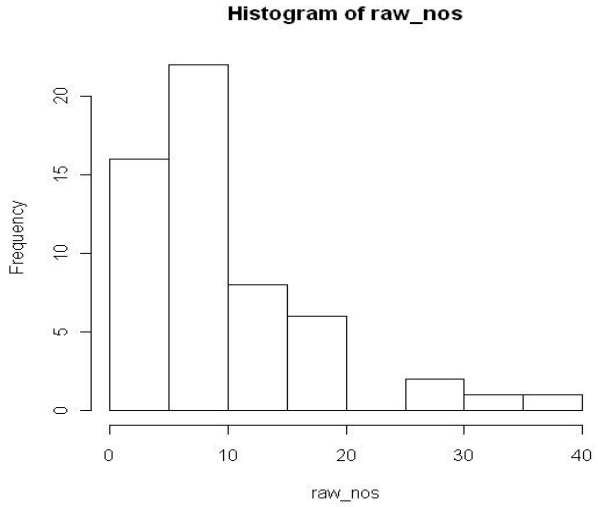
This is a totally different picture. Although Banff/Buchan remains moderately high (0.1686 per 1000 POP), the highest rates of incidence are now seen to be in the rural districts of Skye/Lochalsh (0.3178 per 1000 POP, the 'population years at risk') and Berwickshire (0.1740 per 1000 POP) and the cities of Glasgow and Aberdeen no longer figure.

Pause for Thought

Area-value or choropleth maps are a very common way of displaying lattice data that are aggregates over the individual areas that make up the map. Is there EVER a case when it is appropriate to map the absolute numbers rather than some population or area standardized ratio? Suggestions to the forum please …

Quite apart from the cartography, we should still be unhappy with this map for two further reasons. First, the frequency distribution of the raw data is basically Poisson:

```
>raw_nos<- lips$CANCER
> raw_nos
 [1]  9 39 11  9 15  8 26  7  6 20 13  5  3  8 17  9  2  7  9  7 16 31 11  7 19
[26] 15  7 10 16 11  5  3  7  8 11  9 11  8  6  4 10  8  2  6 19  3  2  3 28  6
[51]  1  1  1  1  0  0
> hist(raw_nos)
```



Second, we are dealing with small numbers, which means that the computed rates can be very unstable and this problem is almost always met when rare events are being considered or the base populations are highly variable. Two mapping solutions have been suggested to help solve these problems and both have been implemented in spdep.

a) Maps based on probabilities

A method that is often used in epidemiology is to map the *probability* of the observed count, given expected values computed according to some assumed process. This was originally suggested in a very early paper by Choynowksi (1959) when mapping the incidence of cancer tumours: *'It then occurred to me to construct a map showing not the incidence of tumors, but the probability of these incidences if in fact the true incidence were the same for the whole area'*. In this approach we first estimate the expected (mean) count in each area using

$$\hat{\mu} = total\ cases\left(population\ in\ zone \Big/ at\ risk\ population\ in\ zone \right)$$

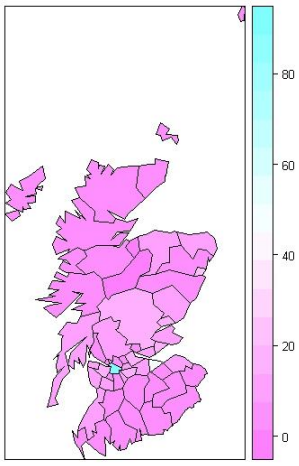
.

This was done by Lawson *et al*. (1999) based on not only the populations but also age distribution effects and so provides a measure of the male population at risk after such population effects are taken into account. This is the variable CEXP in the data. If you prefer it, you can do the same analysis using the available data in POP and the knowledge that in total over the entire country there were 526 cases. This enables a slightly different standardization to be accomplished for the male population at risk but not for its age distribution.

```
> expected_nos <- lips$CEXP
> expected_nos
 [1]  1.38  8.66  3.04  2.53  4.26  2.40  8.11  2.30  1.98  6.63  4.40  1.79
[13]  1.08  3.31  7.84  4.55  1.07  4.18  5.53  4.44 10.46 22.67  8.77  5.62
[25] 15.47 12.49  6.04  8.96 14.37 10.20  4.75  2.88  7.03  8.53 12.32 10.10
[37] 12.68  9.35  7.20  5.27 18.76 15.78  4.32 14.63 50.72  8.20  5.59  9.34
[49] 88.66 19.62  3.44  3.62  5.74  7.03  4.16  1.76
>sum(expected_nos)
[1] 536.01
```

It is useful to plot these expected numbers if only to make the point that, with the one exception of Glasgow city itself where we have over 88 cases expected, their distribution isn't too spatially variable:

```
> spplot(lips,"CEXP")
```



Instead of mapping the ratio of observed to expected counts, Choynowski suggested mapping the probability of getting a count more extreme that that observed under the assumption that the count in each area is Poisson with mean value μ_i. The zone means are estimated as above and for observed values, y_i , greater than the estimate of the mean $\hat{\mu}_i$ we map the probability:

$$p_i = \sum_{x \geq y_i} \frac{\hat{\mu}_i^{\,x} e^{-\hat{\mu}_i}}{x!}$$

In the package spdep this is implemented as choynowski:

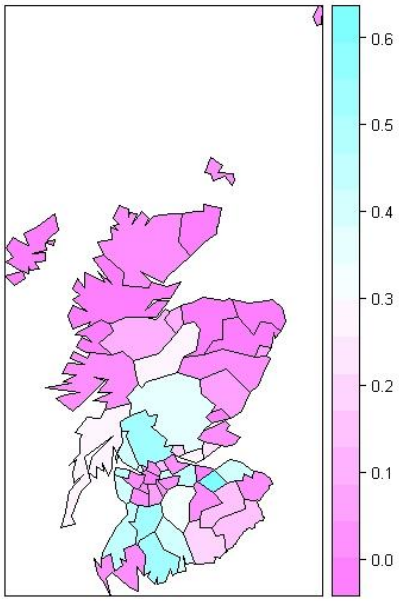
```
> library(spdep)
> ch  <- choynowski(lips$CANCER, lips$CEXP)
```

Ensure that you understand what is being passed to this method and why. The result is in ch:

```
>ch
        pmap  type
1  1.456447e-05 FALSE
2  3.963496e-14 FALSE
3  3.262428e-04 FALSE
Etc
56 1.720505e-01  TRUE
```

Note that by default choynowski returns an object called pmap and the spatial distribution of probabilities looks like:

```
>lips_extra<-spCbind(lips,ch$pmap)
>spplot(lips_extra,"ch.pmap")
```



Small value of p_i, say <0.05, would indicate a district with an unusually high *or* unusually low value. The basic approach folds both tails (high and low) of the distribution together. They can be separated using probmap. Probability maps have the disadvantage that if the underlying distribution of the data does not agree with our assumption, we can get several processes mixed up and so get misleading results but they are likely to be an improvement on basic choropleths using the rates.

b) Empirical Bayes

This is all very well, but the small number issue remains. An alternative is the Empirical Bayes (EMB) approach introduced by Clayton and Kaldor (1987), which is also implemented in spdep. The idea is simple, the computation less so. The idea is that our confidence in any mapped values might depend on many things, but, in the case of lattice data where the values are aggregated counts, the higher the count the more we are likely to have confidence in our mapped ratios. As we have seen, small numbers in the numerator make the ratios very unstable to quite minor changes, whereas large numbers give grounds for greater confidence. EMB takes advantage of this by shrinking small values towards the global mean by a relatively large amount but keeping large numbers close to the observed value. Our prior knowledge is given by the observed rate across all areas and we use Bayesian techniques to modify what we observe in a given area on the basis of this. If the unknown rate in each area, i, is θ_i and the observed rate is $r_i = y_i / n_i$ (as above). Without the shrinkage it is the r_i's that we would map as best estimates of the θ_i. But if we have a prior probability distribution for each θ_i with mean γ_i and variance ϕ_i, it can be shown that the best Bayes estimates of θ_i combining the observed rates and these prior distributions is:

$$\hat{\theta}_i = w_i r_i + (1 - w_i)\gamma_i$$

In which the weighting factors w_i are:

$$w_i = \theta_i \Big/ (\theta_i + \gamma_i / n_i)$$

It is worth thinking carefully about this. First, the weighting factors depend on both the population at risk in that zone and the mean and variance of the prior distribution. If for a given zone the weighting factor is close to 1, the effect in the first equation is to drag the estimate close to r_i the observed rate. If on the other hand it is close to 0, the second part of the first equation is dominant and will drag the estimate towards the prior mean. The issue thus resolves itself into how to obtain values for the prior means γ_i and variances ϕ_i. In classical Bayesian work we might have some prior belief, but in the EMB approach all we have are the data and we use them to estimate these means and variances as well. There

are several ways of doing this, of which the simplest is the method of moments which is implemented in spdep. In this we first estimate γ_i by the pooled mean of the observed rates:

$$\hat{\gamma} = \frac{\sum y_i}{\sum n}$$

Second, we estimate ϕ_i as the weighted sample variance of the observed rates about this mean:

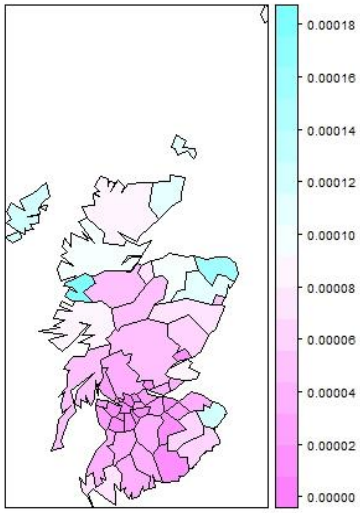
$$\hat{\theta} = \frac{\sum n_i (r_i - \hat{\gamma})^2}{\sum n} - \frac{\hat{\gamma}}{\bar{n}}$$

Substitution of these estimates into our original equations for w_i and θ_i gives the required values. The more complex approach assumes a particular form for the prior distribution, typically Gamma, and then estimates its scale and shape parameters by maximum likelihood. This alternative method is implemented in the R package DCluster, but in spdep all we do is:

```
> emp_bayes <- EBest(lips$CANCER,lips$POP)
> emp_bayes
          raw       estmm
1  3.177517e-04 1.752925e-04
2  1.685852e-04 1.538268e-04
3  1.322274e-04 1.073465e-04
4  1.740476e-04 1.244521e-04

ETC

56 0.000000e+00 7.820221e-06
>extract the parameter estimates ….
> unlist(attr(emp_bayes,"parameters"))
          a           b
1.237143e-09 3.578129e-05
> use spCbind to put emp_bayes back into the shapefile
> lips_emp <- spCbind(lips,emp_bayes$estmm)
> plot the map
> spplot(lips_emp,"emp_bayes.estmm")
```



Task 3.2: Producing ratio and a √(chi-square) map

It will be worthwhile taking a break here and completing the first part of the assignment associated with this week's lesson, which implements a third general alternative approach to mapping this kind of aggregate lattice data based on the chi-square *Pearsonian residual*.

c) Alternative choropleth mapping packages

If you come from a cartography/GIS background, I suspect that the default cartography used in spdep will not be to your taste. Where is the so-called 'base-detail, the title, scale bar, north arrow, and so on, that your cartography professor insisted should be on any map? If you are willing to 'fiddle' around a bit you can add this in spdep, and you certainly can use the RColorBrewer package to create nice looking color palettes for this type of map. Another option I have seen is the choropethr package, see:

- cran.r-project.org/web/packages/choroplethr/choroplethr.pdf

Best of all is probably Brunsdon and Chen's *GIStools* package:

- cran.r-project.org/web/packages/GISTools/index.html

3.3 Global spatial autocorrelation

Task 3.3: Global spatial autocorrelation

Read Sections 7.4, 7.5 and 7.6 pages 199 - 211 of O'Sullivan and Unwin (2010). This is probably the most important reading for this lesson.

In the First Edition, we approached the topic more gently, through the Joins Count statistics, also Section 7.4 - 7.6, pages 180 - 202 but with the benefit of 10 further years of hindsight I concede that this approach should perhaps be regarded as 'legacy'.

If you have a copy of Bivand *et al*. (2008) the same material is covered at much greater depth in Chapter 9 *Areal Data and Spatial Autocorrelation*, pages 237-268 Brunsdon and Comber (2015) have a very elegant introduction to the same materials on pages 220-235

Many scientific disciplines have discovered spatial autocorrelation in area objects, with the classic being an 1899 exchange between Tyler and Galton that has led to the name *Galton's problem* sometimes being used to refer to the difficulties it creates for conventional analysis. Galton questioned whether or not observations across area objects could be considered as truly independent observations, since they might all just reflect a general pattern from which they all had originated, with positive spatial dependence tending to reduce the amount of information contained in the observations. The simple way to understand this is to remind you that if, as is often the case with spatial data, nearby observations could be used in part to predict each other then we have some degree of spatial autocorrelation.

Moran's *I* is a perfectly logical measure of the phenomenon. It is calculated from what at first sight looks a formidable equation but, hopefully as the text shows, when you break it down into individual pieces it is entirely logical:

$$I = \left[\frac{n}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \right] \times \left[\frac{\sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} (y_i - \bar{y})(y_j - \bar{y})}{\sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij}} \right]$$

When dealing with area objects, the critical piece of information we need to supply are the w_{ij} values that switch on the calculation of the covariances between neighboring areas and together form a *geographic structure matrix*, **W**, with as many rows and columns as there are area objects:

$$\mathbf{W} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & & \vdots \\ \vdots & & \ddots & \vdots \\ w_{n1} & \cdots & \cdots & w_{nn} \end{bmatrix}$$

Each w_{ij} value expresses the assumed spatial relationship between locations *i* and *j*, including the entries on the diagonal that represent how we regard each area's relationship with itself. Note that while the order of the locations is arbitrary, the order must be the same for both the rows and columns of the matrix. In essence, what we are putting into the calculation by the **W** is some prior information, almost an hypothesis in fact, about how we expect the areas to be associated together, and for this reason it is extremely important to be sure that any autocorrelation measure we calculate is based on a sensible **W**. It is for this reason that, before dealing with Moran's *I* itself, almost all reasonably advanced texts in the field go into some considerable depth on the various options that are commonly used when creating a **W** matrix. As you might expect, the R package spdep has a number of methods for creating (or importing) a *neighborhood list* and the neighborhood list itself is an object of type nb.

3.4 Defining the Neighborhood and the Spatial Structure Matrix

The simplest basis for constructing a **W** matrix is one based on *contiguity* such that area objects are considered neighbors if they share one or more boundary points, with the slight added complication that we have to decide whether or not to include as neighbors areas that just touch at a single point. This is called *Queen's Case* contiguity after an analogy with the moves the Queen can make in chess. Using the same lip cancer data from Section 3.2, the appropriate R command is poly2nb. Assuming we have the spdep library loaded and have read in the Clayton and Kaldor shapefile into the R object lips, this is:

```
> lips_nb<-poly2nb(lips,row.names=NULL,queen=TRUE)
```

The neighborhood list object can be summarized:

```
> summary (lips_nb)
Neighbor list object:
Number of regions: 56
Number of nonzero links: 234
Percentage nonzero weights: 7.461735
Average number of links: 4.178571
3 regions with no links:
6 8 11
Link number distribution:
 0  1  2  3  4  5  6  7  8  9 11
 3  1  6 11 14  8  8  1  1  2  1
1 least connected region:
3 with 1 link
1 most connected region:
29 with 11 links
```

Notice that in this case we have three utterly disjoint areas (the Island Authorities) and that the link number distribution is what we (O'Sullivan and Unwin, pages 197-9) call the *contact number* distribution for this lattice of areas. We can plot the pattern of contiguities, provided we can create reference points inside each area to be taken as in some sense representative of each area as a matrix of co-ordinates:

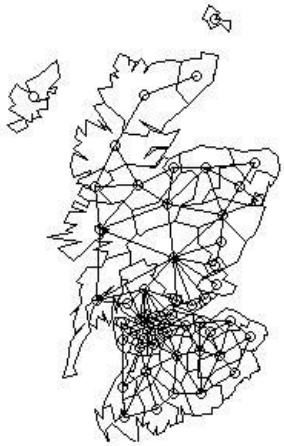
```
> matrix<-coordinates(lips)
> matrix
         [,1]     [,2]
 [1,] 197678.1  824339.8
 [2,] 383873.1  851587.0
 [3,] 314824.1  949563.9
```

Etc

```
[56,] 320644.3  589450.4
```

With these centroid reference points, we create our map of the **W** matrix and superimpose the zone boundaries as:

```
> plot (lips_nb, matrix)
> plot(lips, add =TRUE)
```



Simple contiguity may well be not what we want. As with these Scottish data, we might not want to allow some 'island' zones to have no neighbors at all and the number of other methods of creating a neighbor list is very large indeed. Each of course also generates a different **W** matrix. The table below attempts to summarize what is available in sdep:

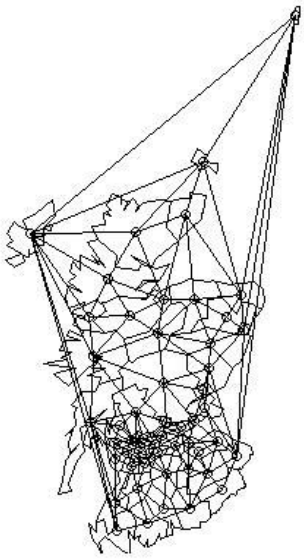
Approach	**R method**	**Comment**
Contiguity-based neighbors	poly2nb	As illustrated above
Graph-based neighbors	graph2nb	Uses a centroid inside each zone (as in matrix) from which a type of graph (Delauney neighbor, Gabriel or relative neighbor) is computed and saved as the nb object.
Distance-based neighbors	knn2nb	Chooses the *k* nearest neighbors based on the inter-centroid distances. Note that **W** can be asymmetric
Higher order neighbors	nblag	'Powering' any binary **W** matrix gives access to the list of second, third, etc order neighbors and so permits examination of the autocorrelation

		structure at various lags (distance or contiguity)
Grid-based neighbors	Cell2nb	Neighbor lists for a regular grid (raster)

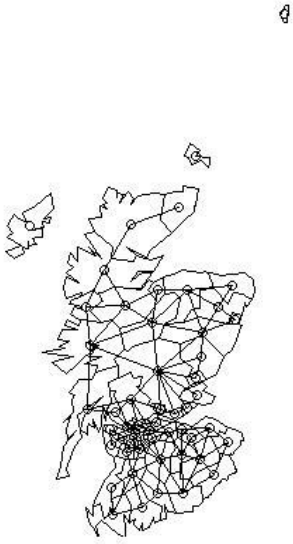
As an illustration, consider the following sequence to find and plot a series of graph-based neighbor lists in the Scottish districts system, all starting from the Delauney triangulation:

```
> delauney_Scot <- tri2nb(matrix)
> SOI_Scot<-graph2nb(soi.graph(delauney_Scot,matrix))
> Gabriel_Scot<-graph2nb(gabrielneigh(matrix))
> relative_neigh_Scot<-graph2nb(relativeneigh(matrix))
> plot.nb(delauney_Scot,matrix)
> plot(lips,add=TRUE)
> plot.nb(SOI_Scot,matrix)
> plot(lips,add=TRUE)
> plot.nb(Gabriel_Scot,matrix)
> plot(lips,add=TRUE)
> plot.nb(relative_neigh_Scot,matrix)
> plot(lips,add=TRUE)
```

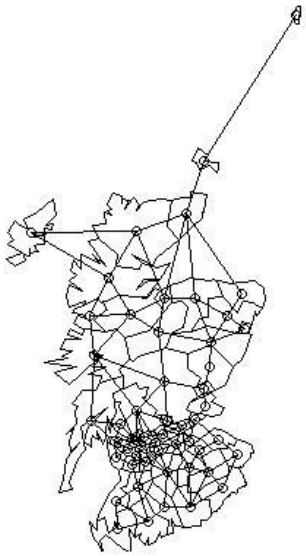
This produces a series of possible definitions of the concept of neighborhood and different **W** matrices:

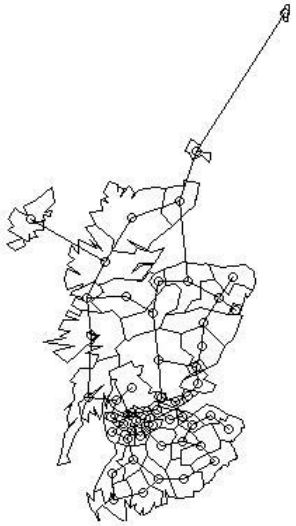


Delauney Triangulation neighbors



Sphere of influence neighbors



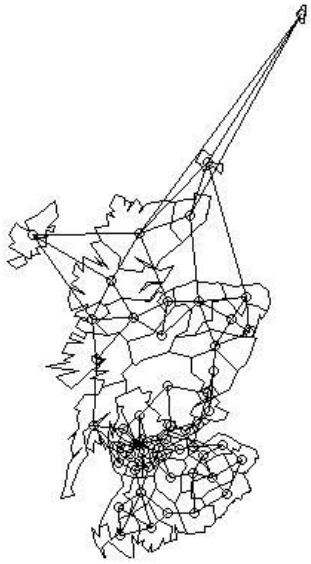


Gabriel graph neighbors **Relative graph neighbors**

Which, if any, of these schemes would be appropriate in the computation of a spatial autocorrelation index in any study of lip cancer totals across the same zones? I am not sure that I could provide an answer, and I hope you take the point that *each represents an hypothesis about the connection between the zones*. The designers of the spdep package were utterly clear and very sensible about all this, forcing their users to work through the choice of structure matrix rather than leaving a default contiguity formulation as is the case in almost all GIS that implement Moran's *I.*

In our case, and with the exception of some neighbors at the edges of the pattern, maybe the Delauney neighbors would be best, but I'd want to edit out some links (using edit.nb). A possible alternative might be based on the $k = 3$ nearest neighbors, again defined by the distances between centroids and using the command knn2nb:

```
> dist_3<- knn2nb(knearneigh(matrix, k=3))
> plot.nb(dist_3,matrix)
> plot(lips,add=TRUE)
```



The neighborhood lists defined above in essence say *which* row/column entries in the structure matrix are to be populated and so will switch on or off the covariance part of the calculation of a global Moran's *I*. The next step is to assign some numerical value to these cells and so create the **W** matrix to be entered into the arithmetic. Any of the methods listed in the table creates a binary 0/1 number set at 1 if the row/column combination of zones is to be considered part of a neighborhood and 0 if not. In most cases (but not all, be careful here!) the relationship will be symmetric. If zone A is a neighbor of zone B, then zone B will also be a neighbor of A and so **W** is symmetric about its principal diagonal (NB: if your math is a little rusty here: Appendix A of the course text, pages 373-393 has a quick primer on what is required). You might like to think through each of the options discussed above and see whether or not this symmetry is maintained. I guess you will have spotted that almost always **W** matrices are very sparse and are typically coded (for example in the .gal format that is almost the *de facto* standard) as a simple list of just the adjacent cases for each of the *n* areas in the system.

We can take things further and replace these binary numbers by some continuous scale that represent the *degree to which* we want the defined pair of zones to be considered as neighbors (see pages 204 and in Bivand *et al.*, 2008, pages 251-258). In general, I concur with Bivand *et al*. (2008) and others in advising that unless you really do know what you are doing, and have some strong motivations for doing it, it is best to stick to symmetric binary (0/1) **W** matrices. Doing this will avoid all the complications and additional *a priori* knowledge incorporated into the calculations and hence interpretations of the results obtained.

However, once again, spdep is very logical about this, so we have to create a list of weights for our neighborhood list object and store them in another class of object called a listw. Doing this uses nb2listw with obvious arguments consisting of:

a) An input object of class nb
b) glist a list of general weights,
c) style, which can be W,B, C, U or S. B gives basic binary coding, W is row-standardized summing over all links to n, C is for a global standardization, U is equal to C divided by the number of neighbors and S is a variance stabilizing coding. It follows from the remarks made above that in general we should specify version B.
d) zero.policy. If this is set TRUE weights are set to zero for regions that don't have any neighbors in the neighbors list (as in the case of our example with some neighborhood schemes). Alternatively, set to FALSE the command assigns the value NA and the default is NULL.

Examples of the use of this will be given in the next section.

3.5 And so back to Moran's *I*

Without even going into the issues of non-binary weights, we have six possible definitions of neighborhood with neighborhood objects called lips, dist_3, delauney_Scot, SOI_Scot, Gabriel_Scot, and relative_neigh_Scot. We also have several variables either supplied or computed of which the object rate is probably the most accessible, recall that:

```
> rate
 [1] 0.317751730 0.168585224 0.132227431 0.174047573 0.116035306 0.150378767
 etc
[55] 0.000000000 0.000000000
```

Recall also that in these example data we have some zones, the three Island Authorities, that in some list generation schemes are left without any neighbors at all. What in these circumstances should we do? Moran's *I* was originally defined assuming that all zones have at least one neighbor and in fact the spdep routines don't work if there are zones without neighbors, so either we only use neighborhood lists that by definition don't have any such areas or provide a software routine that gets around the problem. This is the purpose of the zero.policy argument in nb2listw (above) that is also carried through to the command moran.test.

To compute a global Moran's *I* the full command with its arguments is:

```
moran.test(x, listw, randomisation=TRUE, zero.policy=NULL,
 alternative="greater", rank = FALSE, na.action=na.fail, spChk=NULL,
adjust.n=TRUE)
```

with arguments:

x	a numeric vector the same length as the neighbors list in listw
listw	a listw object created for example by nb2listw
randomisation	variance of I calculated under the assumption of randomization, if FALSE normality is assumed
zero.policy	default NULL, use global option value; if TRUE assign zero to the lagged value of zones without neighbors, if FALSE assign NA
alternative	a character string specifying the alternative hypothesis, must be one of greater (default), less or two.sided.
rank	logical value - default FALSE for continuous variables, if TRUE, uses the adaptation of Moran's *I* for ranks suggested by Cliff and Ord (1981, p. 46)
na.action	a function (default na.fail), can also be na.omit or na.exclude - in these cases the weights list will be subsetted to remove NAs in the data. It may be necessary to set zero.policy to TRUE because this subsetting may create no-neighbor observations. Note that only weights lists created without using the glist argument to nb2listw may be subsetted. If na.pass is used, zero is substituted for NA values in calculating the spatial lag
spChk	should the data vector names be checked against the spatial objects for identity integrity, TRUE, or FALSE, default NULL to use get.spChkOption()
adjust.n	default TRUE, if FALSE the number of observations is not adjusted for no-neighbor observations, if TRUE, the number of observations is adjusted

First, we assign binary weights to all six neighbor lists, using nb2listw and taking care where necessary to set our zero.policy:

```
> contig_listw <- nb2listw(lips_nb, style="B", zero.policy=TRUE)
> dist_3_listw <- nb2listw(dist_3, style="B")
> SOI_listw <- nb2listw(SOI_Scot, style="B", zero.policy=TRUE)
> Gabriel_listw <- nb2listw(Gabriel_Scot, style="B", zero.policy=TRUE)
> rel_neigh_listw <-nb2listw(relative_neigh_Scot, style="B", zero.policy=TRUE)
> delaun_listw <- nb2listw(delauney_Scot, style="B")
```

Having assembled our **W** matrix, we can now compute global Moran's *I* for the six different neighborhood lists. Note that this is the first time we bring back the observed values in the zones, in this case the computed rates, but this could be any of the variables in lips.

```
> moran.test(rate, listw = contig_listw, zero.policy = TRUE)
> moran.test(rate, listw = delaun_listw)
```

```
> moran.test(rate, listw = dist_3_listw)
> moran.test(rate, listw = SOI_listw)
> moran.test(rate listw=Gabriel_listw,zero.policy = TRUE)
> moran.test(rate, listw=rel_neigh_listw,zero.policy = TRUE)
```

A summary of the results is given in the table below:

Scheme	**Moran's *I***	**Expected value**	**Variance of (E)**	***z*-score**
Simple contiguity	0.363263693	-0.019230769 (n=52)	0.006769752	4.6488
Delauney	0.519599336	-0.018181818	0.005068704	7.5537
Distance k=3	0.543587908	-0.018181818	0.008287442	6.1709
Sphere of influence	0.483547126	-0.018181818	0.006087487	6.4306
Gabriel graph	0.371846634	-0.022222222 (n=45)	0.007022745	4.7024
Relative neighbors	0.38126027	-0.02500000 (n=40)	0.01206414	3.6988

Under the null hypothesis of no autocorrelation, the expected value of I isn't quite 0, but is in fact:

$$E(I) = -\frac{1}{(n-1)}$$

There are two ways of computing the variances based on randomization (as in spdep) or normality. Moran's *I* can be shown to be asymptotically normal under both assumptions as long as *n* is moderately large.

Looking at these results three comments are necessary. First, I realize that some may find this range of values upsetting but this occurs fairly frequently when we use spatial data (recall Lesson 2 where we found different values for intensity estimates and for most of the point-pattern measures according to how we chose to define the region of interest). Second, and mercifully in this case, they all seem to point inexorably to the same conclusion. We have significant positive global spatial autocorrelation. Third, although Moran's *I* is relatively easy to calculate, many spatial analysts recognize that the assumptions made in assigning a *p*-value to such an observed value will almost never be sustained. As we have seen, the spatial structure of the zones used is in some sense also a parameter in the analysis, so a more usual approach is to use a Monte Carlo procedure in which the location attributes are randomly assigned to the zones a specified number of times (99, even 999) and a value for *I* calculated in each case. This enables to observed value to be ranked relative to these simulations and its statistical 'significance' assessed, at least informally. In spdep, this is easily accomplished using moran.mc, for example:

```
> #monte carlo values for delaun_listw
># Start the pseudo-random number generator differently each time you do this
> set.seed=(4567)
> moran.mc (rate, listw=delaun_listw,  nsim=99)

      Monte-Carlo simulation of Moran's I
data:  rate
weights: delaun_listw
number of simulations + 1: 100
statistic = 0.5196, observed rank = 100, p-value = 0.01
alternative hypothesis: greater
```

This confirms what we have already seen. We will continue to use the delaun_listw list in the rest of this lesson.

Task 3.4 : Global Spatial Autocorrelation

Now go to the assignment for this lesson and complete the next task, which asks you to select some data in shapefile format, incorporate into the R-environment and then use the spdep package to define and justify a **W** matrix and compute and interpret Moran's *I* for global spatial autocorrelation. Be sure to save your workspace for this Task: it will be used in the final part of the assignment.

The spdep package contains routines for computing Moran's *I* with several other inputs, for example if the zone values are estimates derived using the EMB approach (EBImoran) or are regression residuals (lm,moran.test). It also contains routines for other possible measures of spatial autocorrelation such as Geary's *C*, the global version of the Getis/Ord *G,* and the Cliff/Ord *joins count* for categorical variables.

3.5 Local Statistics: the Moran Scatterplot (MSP)

Task 3.5: Local Statistics

Now read Chapter 8 of O'Sullivan and Unwin (2010) Local Statistics, pages 215 – 226. One of the problems with this topic is that much of what passes for 'global' can in fact be interpreted as a 'local' statistic. Examples include KDE, where we estimate the local value of the intensity, and

spatial interpolation, where we estimate the local mean of some geostatistical data. If you have a copy of the First Edition of O'Sullivan and Unwin, the index will direct you to several points at which we discuss the idea. If you have a copy of Bivand *et al.* (2008) similar comments apply, but in the context of area objects, see pages 268 – 272

Brunsdon and Comber (2015) have Sections 6.1 to 8.6 pages 253-278 on local statistics and, as you will discover Chris Brunsdon has been one of the pioneer developers of this class of statistical measures.

Global statistics have severe limitations for work in spatial data analysis. First, the assumption is usually that assumed generating process is stationary over all the studied space. As GIS systems have enabled researchers to use either larger study regions or, equivalently, data sets at much finer spatial resolution, so this assumption seems more and more unrealistic. Basic geographical theory shows that such spatial homogeneity over large areas of the earth's surface or at fine resolution is extremely unlikely. What may well happen is that large areas of uninteresting spatial variation swamps those of real interest, so we are back to the problem that we addressed when dealing with point events, deciding *where* there is a cluster/concentration of 'interesting' observations in some lattice data. For this part of the lesson, Anselin (1996) is probably the best source of additional information.

As the readings show, a number of local indicators of spatial association (LISA) statistics have been defined and used, including the Getis/Ord (1995) *G* and *G** statistics, and Anselin's local version of Moran's *I* (Anselin, 1995, 1996). It cannot be stressed too highly that, just as the global versions of these measure depend on how we chose to define and weight the neighborhood of some element of a spatial lattice, so all these measures exhibit exactly the same dependence. In fact, the issue is related to the scale at which we chose to observe and is very general, appearing in slightly different guises in almost all spatial statistical analysis. In Lessons 1 and 2 we saw how the bandwidth affects the view we get using kernel density estimation and in this lesson we have already seen how the definition of a spatial structure matrix **W** affects our autocorrelation measure. In the next lesson we'll see that essentially the same issues of range and weighting arise when we interpolate geostatistical data.

Given that this has been a long and hard lesson, we'll illustrate LISA using local Moran's *I* and a related graphical device called the *Moran Scatterplot* using the lip cancer data and the `moran.plot` command. To find values of the local version of Moran's *I* at each location the following quantity is calculated:

$$I_i = z_i \sum_j w_{ij} z_j$$

where the *z* values are standardized *z*-scores determined from the values of the attribute of interest from the whole dataset. Positive values of I_i result where either low or high values of the attribute are near one another, and negative values result where low and high values are found in the same area of the map. Thus local Moran's *I* gives an indication of data homogeneity and diversity. In their example, Bivand *et al*. (2008, pages 268-269) use the raw data in computing the local Moran's *I,* but in common with most authors, I prefer to use standardized values, so first we transform the variable rate into *z*-scores:

```
>mean_rate<-mean(rate)
> mean_rate
[1] 0.05977358)
> sd_rate <- sd(rate)
> sd_rate
[1] 0.05846674
> zrate <- (rate-mean_rate)/sd_rate
> zrate
 [1]  4.41239128  1.86108611  1.23923187  … etc … -1.02235176
```

At this point we can compute and plot the Moran Scatter plot with these normalized values using moran.plot:

```
> moran.plot( zrate, listw = delaun_listw)
```

It is sometimes useful to access the computed values for each zone using localmoran:

```
> # compute zone values of local I,  its expected values, z-score and hence p value
> local_delaun <- localmoran(zrate,delaun_listw)
> local_delaun
             Ii        E.Ii   Var.Ii        Z.Ii    Pr(z > 0)
 [1,] 13.95576149 -0.07272727 3.521745  7.47535691 3.849730e-14
 [2,] 10.22969493 -0.10909091 5.285365  4.49709470 3.444416e-06
 [3,]  7.79903923 -0.09090909 4.403886  3.75972548 8.504995e-05
etc
 [56,]  1.50973674 -0.10909091 5.285365  0.70414663 2.406707e-01
attr(,"call")
localmoran(x = zrate, listw = delaun_listw)
attr(,"class")
[1] "localmoran" "matrix"
```

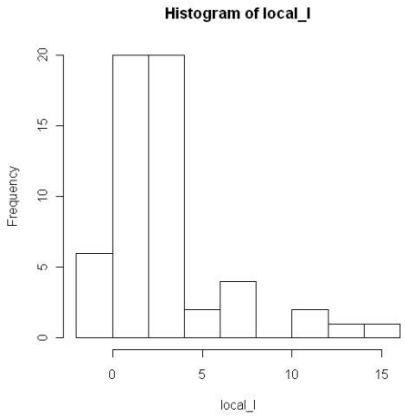
Note that the object created by localmoran is a matrix, so we can extract specific columns and use them in standard operations such as forming an histogram of the values:

```
> local_I <- local_delaun[,1]
> local_I
 [1] 13.95576149 10.22969493  7.79903923   … etc … 1.50973674
> #draw the histogram
> hist(local_I)
```

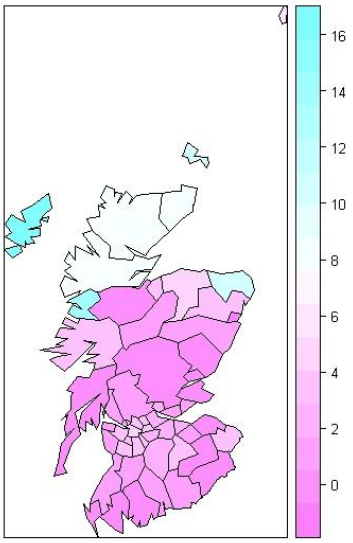
More usefully we can bind them back to a spatialPolygonDataFrame and draw maps:

```
> lips_extra <- spCbind(lips_extra, local_I)
> names (lips_extra)
 [1] "CODENO"    "AREA"      "PERIMETER" "RECORD_ID" "DISTRICT"  "NAME"
 [7] "CODE"      "CANCER"    "POP"       "CEXP"      "AFF"       "pmap"
[13] "type"      "local_I"
>#draw the map
> spplot(lips_extra,"local_I")
```

The three graphics created by the above sequence are presented below:

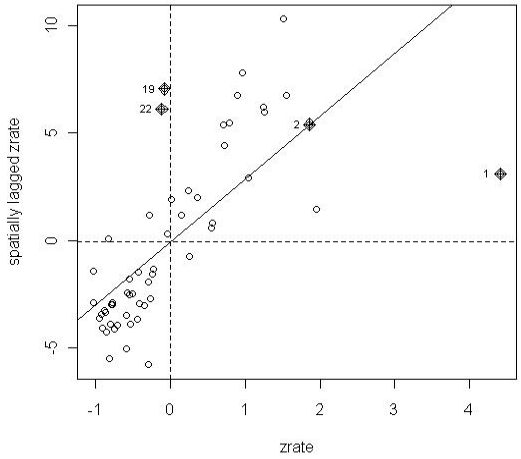


Histogram of local Moran's I, lip cancer rate with Delauney neighborhood and binary weights



A choropleth map of the values of local Moran's I shows areas unusually high or low relative to their neighbors

It has to be remembered that these are local values, relative to the defined neighborhood of each zone and their interpretation can thus be tricky. I have to say that I am not convinced that the map of local Moran's *I* in this instance greatly helps further our understanding of the geography of this disease, but you might have a different view.



Moran Scatter Plot of these same data, for explanation see lesson text

The Moran Scatter Plot provides a visual exploration of spatial autocorrelation in which we have the spatial lag of the variable on the vertical axis and the original variable on the horizontal axis (spatial lag being the mean value of its defined neighbors). Using standardized variables means that the value at the cross-

hairs of the graph for both variables is a z-score of 0 and that the units on both axes are in terms of standard deviations. The horizontal *x*-axis gives the standard scores for the observed values in each of the *n* areas, against which on the vertical *y*-axis is plotted the mean of the standardized neighbors as defined in the **W** matrix used. What is perhaps less obvious is that the slope of the best-fit line through the points is proportional to the global Moran's *I* for the dataset. Take a second or two to think through why this should be so. The four quadrants in the plot provide a classification of four types of spatial autocorrelation. The table below provides a guide to the interpretation that can be given to each of the plotted area objects*:*

z-scores	***Scatter plot quadrant***	***Autocorrelation***	***Interpretation***
high-high	upper right	positive	Cluster - "I'm high and my neighbors are high."
high-low	lower right	negative	Outlier - "I'm a high outlier among low neighbors."
low-low	lower left	positive	Cluster - "I'm low and my neighbors are low."
low-high	upper left	negative	Outlier - "I'm a low outlier among high neighbors."

In Anselin's own *GeoDa*™ system the Moran Scatter Plot can be object linked back to the map, and this seems a useful exploratory device. Quite apart from the distributional and inferential problems O'Sullivan and I mention (Section 8.4, pages 223-226), my academic background in geography and cartography makes me mildly suspicious of these local approaches and for the good reason that often the results, usually another choropleth map of the chosen statistic, seem to me all too frequently to have transformed one mapped pattern we find hard to understand (the original data) into another choropleth map (of *G*, *G**, or Local Moran's *I*) that is equally, if not more, perplexing. You may think otherwise and I'd be interested in your comments.

Task 3.6: Moran Scatter Plot

Now complete the next part of your assignment for this lesson, which asks you to generate and comment on a Moran Scatter plot for your own data.

3. 6 Geographically Weighted Statistical Models

The basic idea behind a LISA statistic, that most geographical data will be non-stationary over space can be taken further into a whole class of statistical methods that are *geographically weighted* (GW) versions of standard models.

There are usually three elements in the derivation of all GW models. First, we need to define for each and every location of interest a neighbourhood over which the model is to be applied using exactly the approaches that we saw in Section 3.3 to define a structure matrix **W.** In effect, the point events or area objects for which we have data that lie within this neighbourhood give a sub-sample of the entire data. Second, we compute the required statistic or calibrate the required model using just this subsample, but with the individual observations weighted 'geographically'. Although a binary in/out weight (0/1) can be used it is usual to weight observations by some function of their distance from the location for which the calculation is being performed such that distant observation carry less weight than those that are close. We have of course met this approach before when we discussed kernel density estimation and we will meet it again in Lesson 4 when we look at the interpolation of continuous surfaces. Just as in KDE, the third and final step is almost always to map the results.

Task 3.7: GWR

Finally, read Section 8.5 of O'Sullivan and and Unwin (2010, pages 226-233 on the method of Geographically Weighted Regression (GWR) developed by Dr Chris Brunsdon and his colleagues from 1996 onwards. The example in the text uses a geostatistical variable (rainfall) that is spatially continuous and is developed more fully in Brunsdon, McClatchey and Unwin (2005). The method itself is developed in Fotheringham, Brunsdon and Charlton (2002) and the work is summarized in Brunsdon and Comber (2015) Section 8.7, pages 278-295. The package GWmodel implements the basic approach.

Although GWR was the first GW model to be implemented using standard ordinary least squares for the calibration, the team that developed the approach in the late 1990s have gone on to implement other forms of regression including generalised linear models, GW summary statistics, GW distribution analysis, GW principal components analysis, GW discriminant analysis, and so on. The GW modelling framework continues to evolve and these models have been usefully

applied to data from a wide range of disciplines in the natural and social sciences. In fact, there is enough in this approach to justify an entire *statistics.com* course, but if this introduction has fired your imagination, can I end by drawing your attention to the necessary resources in the R environment?

First, the GWmodel package available from your CRAN mirror provides functions to a conduct many of these GW analyses. Second, there is a useful tutorial guide to the package written by its originators at

//arxiv.org/pdf/1306.0413v2.pdf (checked 24-10-14)

which contains (almost) all you need to get started.

3.7 Conclusion

When we analyze any lattice data, for example the aggregated population numbers from a census, it is fairly obvious that the areal units could be changed and so possible change our ideas about these data. This is the modifiable areal unit problem (MAUP) that is sometimes turned to politician's advantage by the process of *gerrymandering*, modifying the boundaries of an electoral constituency to help achieve some desired result. Sometimes this might not matter, but very often it does. However, when we come to map these same lattice data, the modifiability of both the zone boundaries and the numbers to be mapped becomes an issue. When we apply almost any standard statistical technique, the problem of spatial autocorrelation rears its head but it would be a funny old world if lots of things weren't spatially autocorrelated. Sometimes I hear quite senior geography professor who don't engage in any spatial analysis say that they 'don't understand spatial autocorrelation'. My response is almost always to say 'then you can't be a geographer' and I walk away leaving them mystified. Think about it. What would our world be like without spatial autocorrelation? Would 'geographers' and even 'spatial analysts' have jobs?

3.8 Data Files Used

a) scotlip.zip a compressed ESRI shapefile from the website at http://geodacenter.asu.edu.
b) Your own choice of shapefile from either the same or some other website.

3.9 References

Anselin, L (1995) Local indicators of spatial association – LISA. *Geographical Analysis,* 27(2): 93-115.

Anselin, L. (1996) The Moran scatterplot as an ESDA too to assess local instability in spatial association. IN: Fischer, M., H.J. Scholten and D.J. Unwin

Spatial Analytical Perspectives on GIS (London: Taylor and Francis, GISDATA 4), Chapter 8, pages 111-125

Bailey T, and A. C. Gatrell A (1995) *Interactive Spatial Data Analysis*, (Harlow: Longman), especially pp. 303–306 and the exercise, pp. 328–330. This is a particularly clear exposition and part of a very fine book.

Banerjee, S, Carlin, B.P. and A.E. Gelfand (2004) *Hierarchical Modeling and Analysis for Spatial Data* (London: Chapman and Hall)

Binbin Lu, Harris, P., Charlton, M and C.Brunsdon (2013) *The GWmodel R package: Further Topics for Exploring Spatial Heterogeneity using Geographically Weighted Models* available at http//arxiv.org/pdf/1306.0413v2.pdf

Brunsdon, C., McClatchey, J. and D.J. Unwin (2001) Spatial variations in the average rainfall/height relationshipin Great Britain: an approach using geographically weighted regression. *International Journal of Climatology*, 21: 455-466.

Census Research Unit (1980) *People in Britain: a census Atlas*. (London: HMSO)

Clayton, D. and J. Kaldor (1987) Empirical Bayes estimates of age-standardized relative risks for use in disease mapping. *Biometrics*, 43; 671-681.

Choynowki, M. (1959) Maps based on probabilities. *Journal of the American Statistical Association*, 54: 385-388

Cressie, N. A. C. (1993). *Statistics for Spatial Data.* Hoboken, NY: Wiley)

Dykes, J. and D.J. Unwin (1998) *Maps of the census: a rough guide*. (UK Advisory Group on Computer Graphics available at http://www.agocg.ac.uk/reports/visual/casestud/dykes/issue1_1.htm

Fotheringham, A.S., Brunsdon, C and M. Charlton (2002) *Geographically Weighted Regression: The Analysis of Spatially Varying Relationships* (Chichester: Wiley)Ord, J.K. and A. Getis (1995) Local spatial autocorrelation statistics: distributional issues and an application. *Geographical Analysis*, 27(4): 286-306.

Lawson *et al.* (1999). *Disease Mapping and Risk Assessment for Public Health.* New York: Wiley, especially pages 68-69.

Marshall R M (1991) Mapping disease and mortality rates using Empirical Bayes Estimators, *Applied Statistics*, 40, 283–294

Martuzzi M, and P. Elliott P (1996) Empirical Bayes estimation of small area prevalence of non-rare conditions, *Statistics in Medicine* 15, 1867–1873.

Visvalingam, M (2000) *Data Rich - Information Poor: Signed chi-squares* , see: http://www2.dcs.hull.ac.uk/CISRG/p.rojects/IAP2K/images/posters/chi-squares.htm

Waller, L.A. and C.A. Gotway (2004) *Applied Spatial Statistics for Public Health Data* (Hoboken, NY: Wiley)

Revised 26-11-13, rev to include GWmodel23-10-14, rev 7-12-15