Spatial Analysis and Data Exploration in History and Archaeology, Spring 2021
LDA-H313

WEEK 2 EXERCISE: Points Pattern Analysis

1 Welcome

This week we will look at the various ways to examine points patters. First, please place the **week2** folder, which contains the data we will examine, in the **sade2021** folder and set your workspace. Make sure that the path is exactly as indicated, for example with out a space in the folder name:

1.1 Setting up

```
Windows:
      setwd("C:\\sade2021\\week2")
Mac:
      setwd("~/Documents/sade2021/week2")
```

1.2 Warming up

As a warm-up exercise, we'll first create an example data frame, i.e. a table. You can do this by creating some vectors, and then combining them into a single table with the **`data.frame`** function. Note how c(6:10) creates a vector of values from 6 to 10.

```
column1 <- c("a","b","c","d","e")
column2 <- c(1,2,3,4,5)
column3 <- c(6:10)

example <- data.frame(column1, column2, column3)
```

Print the dataframe by writing the object's name and pressing enter:

```
example
```

You can now look at specific values in the dataframe by using square brackets. These are used to index the data frame. Try the following. The first number indicates the row, and the second the column.

```
example[2,3]
```

You should get 7. You can also look at entire rows or columns by leaving the other value empty:

```
example[4,]
```

```
example [,2]
```

And if you want to, you can create in the workspace a new object with the selected values:

```
new_object <- example[3:5,]
new_object
```

The new object is a new data frame, which contains rows 3 to 5 of the original data frame.

2. Point Densities

Now lets load up the libraries we will need for the following exercises. You haven't installed them already, please first install the spatstat and maptools packages with **`install.packages()`**.

```
library(rgdal)
library(raster)
library(spatstat)
library(maptools)
```

2.1 Loading and looking at the data

First we will load a polygon shapefile representing the area (England and Wales) that we will be working with. The point events (historical market settlements) are all located within this space. The complex geometry of the polygon (that's the coastline of England and Wales, and the land border with Scotland) can make the exercises computational quite intensive, so I have simplified the geometry. It is assumed that you are using the shapefile **engwales_simple.shp** but if you are finding that some calculations still take very long (hours rather than minutes at most) you can use the further simplified polygon **engwales_verysimple.shp** in the same folder. Just substitute that name in the below command.

```
polyg <- readOGR(dsn="englandwales", layer="engwales_simple")
```

As you can see in the above command, the file we loaded is called **engwales_simple**, and is located in the folder **englandwales**. You can use **?** to examine the **`readOGR`** function - this is always good for troubleshooting.

```
?readOGR
```

Now we'll load a Digital Elevation Model (DEM) taken from NASA's Shuttle Radar Topography Mission data. We won't use it for calculations, but is a creates map for contextualising the historical sites. We'll load it, plot the polygon and add the DEM to the graph (note the add=T argument). The DEM is a big file, so be patient, drawing it may take a few moments.

```
dem <- raster("dem/engwales_dem.tif")
plot(polyg)
plot(dem, add=T)
```

Note that the scale bar gives relative values, not absolute height values in this instance. The colour scheme you see is what R automatically draws, and the visualisation can be a bit different depending on what kind of a computer you are working. Usually, it gives green colours for higher ground and lighter colours for lower ground, which for an elevation model is somewhat backwards. Fortunately R comes with several option for colour ramps, and we'll use **terrain.color** (note: no "u" in "color") redraw the map.

```
plot(dem, add=T, col=terrain.colors(10))
```

Now you have a rather nicer map. The value (10) sets the value ranges of the hues. Try a lower value, such a 3, and replot the map. **Note!** You can use the up and down keys to scroll through previous commands, which speeds things up.

Next we'll load and map the locations of medieval market sites (many of them also towns) that have been recorded in documents by AD 1334. This data is taken from Samantha Letters's (2002) *Gazetteer of Markets and Fairs in England and Wales to 1516*.

```
markets <- readOGR(dsn="markets1334", layer="markets1334")
points(markets, pch=19, cex=0.2)
```

Note how the topography (a **first order effect**) visibly influences the locations of markets in many parts of England and Wales! Most obviously there are not many

markets in regions of higher ground, which were historically sparsely populated. Similarly, there is a lacuna of sites in east-central England, north of Cambridge. This was the Fenlands, a low-lying marshy region not drained until the Early Modern period.

If you enter **`class(markets)`** you will see this object is a Spatial Points Data Frame, or a table with spatial point locations. You can examine the column headers and, say, the first ten rows with **`head(markets, n=10)`** to get an idea of what it contains. In order to carry out the analysis, we'll need to convert it into a different class of an object, a spastat object. Before we do this, we need to make sure that all points fall with the polygon boundaries.

2.2. Manipulating the data

First, lets make sure that the two have the same coordinate reference system (CRS); we'll take the polygon's CRS and apply it to the markets. It is not actually necessary here (they both use the British Ordnance Survey projection) but you might need to do this in the future.

```
markets <- spTransform(markets, CRS(proj4string(polyg)))
```

Then we'll subset the markets with the polygon area, deleting all points that fall outside the latter. This is especially important, since we are using simplified coastlines and some coastal sites may now fall outside the somewhat abstracted bounds. We might not want to do this for a research project, perhaps instead adjusting the polygon so that it covers all sites, but for this exercise it will do.

```
markets <- markets[polyg, ]
```

We know that the recorded markets are, in fact, ***events*** taking place on a specific week day, and there may be one than more market event in the same place over the course of the week. We can remove these spatial duplicates easily, if we want to examine the distribution of market ***sites*** rather than market ***events***. First we'll create a back-up object of the market event dataset, though, since we'll want to look at it again towards the end of the exercise.

```
market_events <- markets
markets <- remove.duplicates(markets)
```

Finally, create spastat objects. We can ignore the warning message about duplicate points.

```
sp_markets <- as.ppp(coordinates(markets),as.owin(polyg))
```

2.3. Kernel Density Estimate (KDE)

Now we can create a kernel density estimate of the distribution of markets, visualised as a kind of a heatmap that helps to pick out structures and patterns in the distribution. We have a lot of sites, and the calculation depends on the complexity of the enclosing polygon, so this may take a few minutes.

```
dens <- density(sp_markets, sigma=10000, edge=TRUE, eps=500)
plot(dens)
```

The bandwidth (or "search radius") of the kernel density is set by the argument **sigma** (here 10000 meters), and the cell size of the resulting surface (i.e. its granularity) by **eps**. Toggling the bandwidth has a big impact on the results, you can try with a few different values to what the results look like (e.g. 5000 or 30000). There are computation ways to determine an "optimal" bandwidth, such as:

```
bw.diggle(sp_markets)
```

This gives about 20 km. But in practise it may depend on what patterns you want to draw out. It can be judgement call, but also one what you need to be able to argue for.

I find the default colour scheme to be a bit psychedelic, so lets you use yellow-red "heat map" colour ramp and replot the map, and add the markets as points on top of it. Can you see how the distribution expresses itself in the kernel density surface?

```
plot(dens, col=heat.colors(10))
points(markets, pch=19, cex=0.2)
```

2.4 Weighing the KDE

We can further interrogate the data by taking into consideration the properties or **attributes** of individual sites. Lets first take another peek at the data frame:

```
head(markets, n=20)
```

The different columns (or "fields") assign different attribute values to the point events, such as their modern names (MODNAMES), and the value of movable goods assessed in the 1334 tax assessment (VAL_1334). This latter creates as rare national-

scale record of assessed wealth during the Central Middle Ages. Lets take a close look at it:

```
summary(markets$VAL_1334)
```

We can see that among these over two thousand market settlements, the mean recorded value was about £73, with the maximum being £11,000 - a big spread! The density function allows us to weigh the resulting estimate with an attribute value. In this case, if we do this for the financial valuation of market settlements, it allows us to better examine regional distribution of wealth as each settlement will be weighted by the tax return.

```
dens_weighted <- density(sp_markets, sigma=10000,
weights=markets$VAL_1334, edge=TRUE, eps=500)
plot(dens_weighted, col=heat.colors(10))
points(markets, pch=19, cex=0.2)
```

There is a problem, though: the great wealth of London, already a dominant metropolis in the fourteenth century, creates a scale that hides patterns in almost all other regions! What can we do about this - lets take another look at the assessment data using a boxplot.

```
boxplot(markets$VAL_1334)
```

This does not look a lot like the boxplot you created last week. The round dots are statistical outliers, and their range is such that they squash the vast majority of the data into a narrow band at the bottom. You'd get a similarly a not very granular result if you used a histogram with **`hist(markets$VAL_1334)`**

Lets extract some stats from the boxplot we created. The following command gives, in the top row, a vector of five values that determine the boxplot's extent.

```
boxplot.stats(markets$VAL_1334)
```

In order: the lower whisker or minimum, the interquartile range of the middle (the second and fourth values), the median (third value) and the upper whisker or maximum. Values beyond the first and the last are "outliers". Because we are here interested in broad distributional structures across the country, we can for now get rid of the high outliers by **subsetting** the data. **`subset`** is a useful function for editing and manipulating data frames. We'll also have to recreate the spatstat object. First we'll save these all the market locations into another back-up object, though, and also the major markets into a separate object.

```
markets_backup <- markets
markets_major <- subset(markets, VAL_1334 > 205)

markets <- subset(markets, VAL_1334 < 205)
sp_markets <- as.ppp(coordinates(markets),as.owin(polyg))
```

Now lets recalculate the density surface and replot it side by the side with the base distribution KDE. We'll use a third colour ramp, the **`topo.color`**, which has a wider colour range, helping to pick out finer features.

Finally, we'll plot as red squares the major markets, which we saved as a separate object above, so that we can still consider their distributions.

```
dens_weighted <- density(sp_markets, sigma=10000,
weights=markets$VAL_1334, edge=TRUE, eps=500)

dev.new(device=pdf, height=6, width=12)
par(mfrow=c(1,2), mai=c(0.5, 0.5, 0.5, 0.5))
plot(dens, col=topo.colors(50))
points(markets, pch=19, cex=0.1)
plot(dens_weighted, col=topo.colors(50))
points(markets, pch=19, cex=0.1)
points(markets_major, pch=15, cex=0.7, col="red")
```

While the broad division between the economically poorer north-western and the richer south-eastern halves England persists (a pattern that is still present today), in this analysis the multiplicity of wealthier individual settlements concentrates to Eastern Midlands and coastal East Anglia, possibly echoing maritime trade. It is also interesting that despite the significant concentration of markets around London many do not at large appear to be assessed at a very high rate. This raises questions about the economic impact of the metropolis on its immediate hinterland.

3 Relative risk surface

Next we will look at creating a relative risk surface. This is a density surface that looks at the relative density of a subset of events, compared to the overall parent population. It has many applications in archaeological analysis, when merely looking at the absolute distribution of the events of a subset might create a false impression of distributional patterns.

New market sites were established through the twelfth and thirteenth centuries, but not at a similar pace in all parts of England and Wales. We will look only at markets established by 1250, and we want to learn in which regions networks of local markets developed early (by 1200) and which developed it later (1201-1250). We will use a workflow originally written by Andy Bevan for his analysis in 2012 article 'Spatial Methods for Analysing Large Scape Artefact Inventories', *Antiquity* 86. You will find the article in the *Further reading* folder for Week 2.

3.1 Setting up

In this exercise we are not interested in markets recorded from the late thirteenth century onwards for a number of reasons, the main being that many were either particularly short lived, or even just speculative franchise grants by the king without a real functional market being established at all. Lets go back to the back-up market object we created earlier, and from it now remove all markets that were established after 1250. We will look at the markets data frame's DEFYR column, which gives the earliest known date a given market sites was established. We'll then recreate the spastat object.

```
markets <- subset(markets_backup, DEFYR<1251)

sp_markets <- as.ppp(coordinates(markets),as.owin(polyg))
```

There is a column ('by1200') which indicates whether a given market's date falls by 1200 (1) or after (0). We will next sub-set the data by these attribute values, and then turn into a multitype point pattern that marks the data by these attribute values.

```
earlier <- as.ppp(coordinates(markets[markets$by1200=="1",
]),as.owin(polyg))
later <- as.ppp(coordinates(markets[markets$by1200=="0",
]),as.owin(polyg))

multit <- sp_markets

marks(multit) <- as.factor(markets$by1200)
```

So if you now plot the multitype object, you should get a map that differentiates the earlier and later markets with different symbols.

```
plot(multit)
```

Next we will create a density estimate of all the markets by 1250 and plot it. We'll use a larger bandwidth at 15 km.

```
dens1250 <- density(sp_markets, sigma=25000, edge=TRUE,
eps=500)
plot(dens1250, col=topo.colors(50))
points(markets, pch=19, cex=0.2)
```

As before, you can see that the south-eastern part of England has, by far, the highest concentration of market sites.

Next we will create a relative risk surface from the multitype object, contrasting markets founded earlier with those founded later.

Regions with either no data, or very sparse data, can create statistical misrepresentation, so it is often a good idea to restrict the surface to exclude these. The second line of the command subsets the rrs object by removing very low values. In this example we have just removed a portion of Wales and northern England that had no nearby market sites. You can experiment with the results by slightly changing the cut-off value 0.000000001 up or down.

```
rrs <- relrisk(multit, sigma=25000, edge=TRUE, eps=500)

rrs[as.matrix(dens1250)<(0.000000001)] <- NA

dev.new(device=pdf, height=6, width=6)
plot(polyg)
plot(rrs, col=rev(topo.colors(50)), add=T)
plot(multit, cex=0.3, add=T)
```

We've marked the plot using the two different symbols. Different regions are now highlighted, with it being clear that especially northern England saw more market foundations in proportion to existing franchises in the thirteenth century. This is because commercialisation and monetisation developed in the local economies slower, in many places taking off almost a century later than in the south (see Oksanen and Lewis, 'Medieval Commercial Sites: as seen through Portable Antiquities Scheme data', in *Antiquaries Journal* 100, 1-32). The significant growth of new market networks in coastal East Anglia (the eastern "bulge") also took place during this period, making it the arguably the wealthiest region of medieval England.

Now, can you plot the two maps side by side, like we did above?

4 Nearest Neighbour

In the final exercise we will look at calculating some useful spatial statistics. The kernel density estimates map the distribution of sites in a very useful manner, but we may want to know more about how they are organised in relation to each other.

A widely used method in many disciplines is P. Clark and F. Evans nearest neighbour index, originally developed in the field of biology: 'Distance to Nearest Neighbour as a Measure of Spatial Relationships in Populations', in *Ecology* 35.4 (1954). First, a study area must be set. Then the method identifies for each event its nearest neighbour (i.e. nearest other market) and calculates the distance to it. An average of these values is then taken, and compared to another value that is calculated based on an expectation of what distances should be (based on a formula) given the shape and the size of the study area if the pattern was completely random. The comparison between these two values gives us information about the structure and organisation of the point events.

4.1. Looking at the data

Lets try this out in practice by looking at the markets data. First, lets calculate the distances between market sites and their closest neighbours. We are still working wit the 1250 data.

```
nndist(sp_markets)
```

This is quite a long list, so lets examine it closer with a couple functions: creating a histogram and taking the mean.

```
hist(nndist(sp_markets), breaks=seq(0, 50000, 1000))
mean(nndist(sp_markets))
```

You'll see that most markets are within 4-8 km of another. The results are roughly in accordance with medieval legal sources, which restrict founding of new markets too close to existing ones - this is a **second order effect**. We must also consider the fact that not all of the listed markets would have been around at the same time, with old market events declined and disappeared, and new ones were founded to replace them. So the actual mean distance would probably be even a bit higher. But the main takeaway, perhaps, is that by the thirteenth century there existed a robust and dense network of local markets in towns and villages that served the economic needs of the population and facilitated commercial growth.

4.2 Clark and Evans test

But how are these sites organised with regard to each other? In this test there are three main patterns that a point distribution can fall into: the points can be **clustered** into groups, they can **dispersed** or the pattern can be **random**. We can use the Clark and Evans test to generate a value that indicates which one these markets fall into. Lets try it out first the distribution of market **events** and then secondly with those just the market **sites**

```
clarkevans.test(sp_markets, corrections="none")
```

The test gives two key values, make note of them. The first is the R value. When R is lower than 1, it indicates that the pattern in clustered. When it is higher than 1, the pattern is dispersed. When it is approximately 1, the pattern in random. The p-value indicates the whether the result is statistically significant; for this a p-value must be 0.05 or lower.

Our results are statistically significant (p = <0.05) and indicate that markets are slightly slightly dispersed (note: 3.206e-06 = 0.000003206). This makes sense; after all, there is pressure not to establish new sites too close to each other.

4.3 Market events

To test our data, we can also look at the dataset of **market events** that we set aside back in section 2.2. Lets subset it to market events recorded by 1250, create a spastat object and run the same analysis.

```
market_events1250 <- subset(market_events, DEFYR<1251)
sp_market_events <-
as.ppp(coordinates(market_events1250),as.owin(polyg))

hist(nndist(sp_market_events), breaks=seq(0, 50000, 1000))
mean(nndist(sp_market_events))
clarkevans.test(sp_market_events, corrections="none")
```

Note the tall first bar in the histogram, denoting market events that share the same market site. The Clark and Evans test returns R = 0.96332, now indicating a slight clustering, which is indeed expected given the different character of this dataset.

These approaches are useful when considering, for example, the historical processes that produce the data we examine. They are not without their problems, however. The first is known as the “edge effect”. The size and shape of the study region impacts on the analysis. Imagine that we have selected just one historical county and

the markets within it for our analysis. What if there are markets very close to the county's border but just outside it - these would not be taking into account, skewing the results. There are various mathematic solutions to this problem, and you can read what R can do if you type **`?clarkevans.test`** and study the help sheet.

The second issue is that the Clark and Evans test gives just a single value for all events. Clustering can look different at different spatial scales, however, so this too can be potentially misleading. Next week we will continue with point pattern analysis, and look at various solutions to the issue.

Finally, save your workspace.

```
save.image("week2_sade.RData")
```

You are now finished with this week's exercise.