Spatial Descriptive Statistics

Getting Started

Before we begin this week's practical, we need to carry out some data preparation.

1. **On Moodle there is a file containing extra categorical, ratio and geographical data that we will need to add to our existing London data file. Download and unzip the file into a new wk7 directory on your Z: drive.**
2. **Save a copy of your LondonData.xls file from week 1 into your wk7 folder**
3. **Append the new data to your LondonData file making sure that your data are ordered correctly, and delete the bottom rows so that you only have 625 rows of data.**
4. **At this point, you should check that your data matches correctly. One way of guaranteeing this is to use the `VLOOKUP()` function to match your data, rather than just using copy and paste. If you are not confident that your data match, follow the steps below, otherwise, move to the main tasks and step 15.**
5. **Add a new to LondonData.xls and copy your newly downloaded data into it sheet (in this example I'll assume it's your new sheet is Sheet2, if it's Sheet3 or something else, when change the formulas accordingly).**
6. **In your ward data file, scroll across to the household tenure columns (probably AM, AN and AO). If this data file contains all of the relevant information about housing tenure (i.e. if everyone either owns or rents privately or socially), these three columns should sum to 100%, but this is not the case**
7. **There is a problem– we are missing some data. These data come from the 2011 Census and visiting http://www.nomisweb.co.uk/ and interrogating Table KS402EW, we can discover that data for the percentage of shared owners and those living in accommodation rent free are missing.**
8. **Add two new empty columns (probably AP and AQ) next to column AP (or whichever column contains the last entry for housing tenure data). In cell AP1, type the following formula:**

```
=VLOOKUP(C2,Sheet2!$C$2:$K$626,2,FALSE)
```

9. **Press return and the function should return a value of 0.296465.**
10. **The `VLOOKUP()` function looks up the value in cell C2, and then searches for it in the left-most column of the table defined as Sheet2, starting in cell C2 and ending in cell K626. The function then returns the value from column 2 of that table.**
11. **Highlight Cell AP2. In the bottom right-hand corner of the cell there will be a small square box. Double click this and your formula should be copied down the whole column. In all cells below you should now have some values. We now want to get rid of the formulae and just keep the**

values. Highlight the whole of column AP, copy the column and then paste special > values so that only the codes are retained. Save your file.

12. **Repeat this vlookup formula in cell AQ2, but this time you want the value from the third column of the table, e.g.**

```
=VLOOKUP(C2,Sheet2!$C$2:$K$626,3,FALSE)
```

13. **Repeat the process again, but this time add the rest of the missing data to the end of the file**
14. **Remove (using the replace facility and replacing them with an empty sting) all non-numeric ('n/a' and '#VALUE!') values from the numeric data in the file and save it as a .csv file called LondonData.csv**

Spatial Descriptive Statistics – what will you achieve by the end of this practical.

1. You will learn how to make descriptive plots (histograms and boxplots) to help understand the frequency distributions of your data
2. You will understand how you can use R to write custom functions to process your data
3. You produce a location quotient map to highlight interesting (above and below average) patterns in your data
4. You will see an example of how you can write a function in R to produce a range of different maps based on user inputs

Main Tasks

- Read your newly updated LondonData.csv file into R

Task 1 - Descriptive Statistics

Using the lecture notes for guidance, you should generate the following graphs and descriptive statistics using standard functions and ggplot2 in R. Each element should be copied and saved to a word document or something similar:

```
#First read your data in to a new data frame called LondonWards

LondonWards <- read.csv("LondonData.csv")

#In case you haven't done this already, subset LondonWards so that
#just the wards are in the dataset

LondonWards<-LondonWards[1:625,]
```

Generate the following from your LondonWards data frame (*hint, use the code in the lecture notes to help you if you are unsure how to do this*):

15. **A simple histogram for a scale/ratio variable of your choice**
16. **A simple histogram for a scale/ratio variable of your with a different frequency bin-width**
17. **The same histogram with vertical lines for the mean, median and mode (the mode will be the mid value for the bin with the largest count) and the inter-quartile range. *hint – use `summary(table$variable)` to find the values if you are not sure***
18. **The same histogram with three different kernel density smoothed frequency gradients**
19. **A boxplot of the same variable**
20. **A faceted grid of histograms with for every variable in your London Wards data file. In order to do this, you will need to remove Factor (non-numeric) variables from your dataset and re-shape your data using the `melt()` function in the `reshape2` package (hint – check the help file for `melt.data.frame()` to understand what the code below is doing):**

```
library(reshape2)

library(gglot2)

LondonMelt <- melt(LondonWards, id=1:3)

attach(LondonMelt)

hist <- ggplot(LondonMelt, aes(x=value)) + geom_histogram(aes(y =
..density..)) + geom_density(colour="red", size=1, adjust=1)

hist + facet_wrap(~ variable, scales="free")
```

21. **Make a note of which variables appear normally distributed and which appear to be skewed. What do the histograms for nominal and ordinal data look like?**
22. **Try performing a log10() transformation on the x variables and plotting a similar facet grid of histograms – what does this do to some of the skewed variables?**

```
hist <- ggplot(LondonMelt, aes(x=log10(value))) +
geom_histogram(aes(y = ..density..)) + stat_function(fun=dnorm,
colour="red", size=1)
```

23. **Create a 2D histogram and 2D kernel density estimate of ward centroids in London using the Eastings and Northings data in the x and y columns of your dataset.**

Task 2 - Introduction to functions in R

- One of the great strengths of R is that is lets users define their own functions. Here we will practice writing a couple of basic functions to process some of the data we have been working with.
- One of the benefits of a function is that it generalises some set of operations that can then be repeated over and again on different data.
- In the lecture, it was mentioned that sometimes we should recode variables to reduce the amount of information contained in order that different tests can be carried out on the data. Here we will recode some of our scale/ratio data into some nominal/weak-ordinal data to carry out some basic analysis on.
- There are various online guides which will help you a little more in writing functions, but the structure of a function in R is given below:

```
myfunction <- function(arg1, arg2, ... ){

  statements

  return(object)

}
```

- A function to recode data in our dataset might look like the one below:

```
newvar<-0
```

```
recode<-function(variable,high,medium,low){
  newvar[variable<=high]<-"High"
  newvar[variable<=medium]<-"Medium"
  newvar[variable<=low]<-"Low"
  return(newvar)
}
```

- First we initialise a new variable called `newvar` and set it to 0. The function called 'recode' then takes in 4 pieces of information. A variable (called variable) and three values called high, medium and low. It outputs a value to the new string variable `newvar` based on the values of high, medium and low that are given to the function.
- To create the function in R, **highlight the all of the code in the function** and then run the whole block (ctrl-Return in R-Studio). You will see that the function is stored in the workspace.
- We can now use this function to recode any of our continuous variables into high, medium and low values based on the values we enter into the function.
- We are going to recode the Average GCSE Score variable into High, Medium and Low values – High will be anything above the 3rd Quartile, Low will be anything below the 1st Quartile and Medium – anything in between.

```
#Note, your data will have different figures to these figures below
summary(AvgGCSE2013)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  245.0   332.3   343.7   345.8   358.3   409.1
```

24. **Create a new column in your data frame and fill it with recoded data for the Average GCSE Score in 2013. To do this, pass the AvgGCSE2013 variable to the `recode()` function, along with and the three values for high, medium and low. You should create a new variable called `gcse_recode` and use the function to fill it with values**
25. **If you wanted to be really fancy, you could try altering the function to calculate these "High", "Medium" and "Low"**

```
LondonWards$GCSE_recode<-recode(AvgGCSE2013,409.1,358.3,332.3)
```

26. **You should also create a second re-coded variable from the unauthorised absence variable using the same function – call this variable `unauth_recode` and again, used the 3rd and 1st quartiles to define your high, medium and low values.**
27. **Make sure these are saved to your data frame as we will use these in next week's practical.**

- On to another function. This time, we will calculate some location quotients for housing tenure in London. If you remember, a location quotient is simply the ratio of a local distribution to the ratio of a global distribution. In our case, our global distribution will be London.

```
LQ<-function(pctVariable){
  pctVariable /mean(pctVariable)
}
LQ1<-function(variable,rowtotal){
  localprop<-variable/rowtotal
  globalprop<-sum(variable)/sum(rowtotal)
  return(localprop/globalprop)
}
```

- The two functions above calculate the same Location Quotient, but the first one works on variables which have **already been converted into row percentages**, the second will work on raw variables where an additional column for the row totals is stored in a separate column – e.g. "age 0-15", "age 16-64" and "age 65 plus" all sum to the "Pop2013" column in our data London Wards data set:

WardName	OldCode	Wardcode	Pop2013	Aged0_15	Aged16_64	Aged65plus
City of London	00AA	E09000001	8000	600	6200	1200
Barking and Dagenham - Abbey	00ABFX	E05000026	13650	3450	9550	700
Barking and Dagenham - Alibon	00ABFY	E05000027	10400	2700	6600	1100
Barking and Dagenham - Becontree	00ABFZ	E05000028	12050	3000	8000	1100
Barking and Dagenham - Chadwell Heath	00ABGA	E05000029	10150	2450	6150	1550

28. Calculate Location Quotients for the 5 Housing tenure variables (Owner Occupied, Private Rent, Social Rent, Shared Ownership, Rent Free) in your data set using either of the functions above. Save these as 5 new variables in your dataset. *Hint – use the function to create the variable directly, for example:

```
dataframe$newLQVariable <- LQ(originalPercentageVariable)
#or
dataframe$newLQVariable <- LQ1(originalVariable,rowTotalVariable)
```

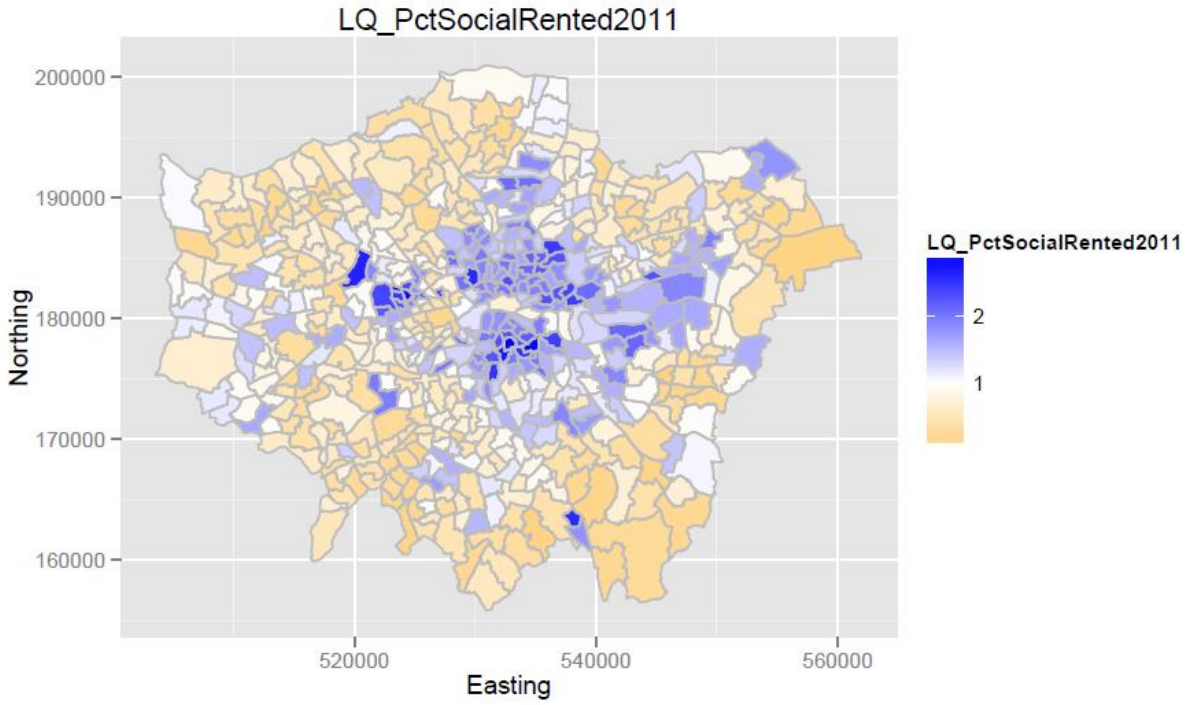
Task 3 – Mapping Location Quotients

29. **You should now try and create a map or series of maps of your housing tenure location quotients using ggplot() – you have two options to try and accomplish this:**
30. **(Easy option) Create a map by referring back to the Week 3 practical and following the steps from there (or your memory)**
31. **(Double hard option) If you want to blow your mind, try the function below **Warning**, I wrote this, so the code is pretty untidy and relies on the input data to already be in the form of row percentages. Creating the maps using it is the easy bit, but see if you can figure out what's going on! Here there are functions inside a function. You should, however, be able to Copy the code into R, run the whole function and then test it out:**

```
############################################################
##A Function for creating various location quotient maps
##
##By Adam Dennett October 2014
##
##Please note, this function requires input data to already be in
##the form of row percentages. To create the function, highlight the
##whole block of code and run it. To run the function, simply use
##LQMapper(your_dataframe)


library(rgeos)
library(ggplot2)


LQMapper<-function(dataframe){
  print(colnames(dataframe))
  vars<-readline("From the list above, select the variables
                 you want to calculate location quotients for
                 separated by spaces...")


  # split the string at the spaces
  vars<-unlist(strsplit(vars, split = "\\s"))
  # now save vars as a list
  vars<-as.list(vars)


  shapefile<-readline("Now enter the name of your shapefile,
```

```
                    e.g. foo.shp")
LondonShp<-readShapePoly(shapefile)

print(colnames(LondonShp@data))

reg<-readline("Now, from the list above, choose the column
              header (variable) you are going to use to split
              your shapefile regions in fortify, e.g. WD11CD.
              *note, this will the the column you will match
              your data on later")

print("fortifying")
London_geom<-fortify(LondonShp, region=reg)

print("looping to create new location quotient variables...")
attach(dataframe)
for(i in 1:length(vars)){
  pctVariable<-vars[[i]]
  colvect<-which(colnames(dataframe)==vars[[i]])

  #this is a little function to calculate location quotients
  LQ<-function(pctVariable){
    pctVariable/mean(pctVariable)
  }
  #use LQ function here to create new variable in dataframe
  #and save it
  v<-dataframe[,colvect]
  dataframe[,paste("LQ_",pctVariable, sep="")]<-LQ(v)
}

#reset i as we're going to use it again in a minute
i=0

matchvar<-readline("Now enter the name of the column header in
```

```
                        your dataframe that you will match your
                        shapefile region to, e.g. Wardcode")


  print("merging new data to fortified spatial dataframe")
  London_geom<-merge(London_geom,dataframe,by.x="id", by.y=matchvar)


  print("now entering the plotting loop")
  for(i in 1:length(vars)){
    print("I'm plotting")
    pctVariable<-paste("LQ_",vars[[i]],sep="")
    colvect<-
which(colnames(dataframe)==paste("LQ_",vars[[i]],sep=""))


    #now make some plot layers - note the use of aes_string which
    #allows us to pass variable names into the aesthetic mappings
    #of ggplot
    layer1<-geom_polygon(aes_string(x="long", y="lat",
fill=pctVariable, group="group"), data=London_geom)
    palette1<-scale_fill_gradient2(low="orange",mid="white",
high="blue", midpoint =1)
    labels<-labs(list(title=pctVariable,x="Easting", y="Northing"))
    layer2<-geom_path(aes(x=long, y=lat,
group=group),data=London_geom, colour="#bdbdbd")


    #create the plot
    LQMapperPlot<-
ggplot()+layer1+palette1+layer2+labels+coord_equal()
    LQMapperPlot
    #save the plot to a pdf and give it a name based on its variable
    ggsave(LQMapperPlot, filename=paste(pctVariable,".pdf",sep=""))


  }
  return(dataframe)
}


#####################################################################
```



Task 4 – Creating a Basic Geodemographic Classification

As we saw in the lecture, geodemographic classifications are widely used to classify areas according to the characteristics of the population that inhabits them. All geodemographic classifications are created using cluster analysis algorithms. Many of these algorithms exist, but one of the most commonly used is k-means. One of the pitfalls of these algorithms is that they will always find a solution, whether the variables have been selected appropriately or standardised correctly. This means that it's very easy to create a classification which is misleading.

All of that said, it is useful to see how straightforward it is to create a classification yourself to describe some spatial data you have.

32. In a cluster analysis, you should select variables that are:

- **a. Ranged on the same scale**
- **b. Normally distributed**
- **c. Not highly correlated**

33. To make this task easier, we will just select two variables to make our classification from. In a real geodemographic classification, hundreds of variables are often used.

```
#clustering

# Create a new data frame just containing the two variables we are
#interested in

mydata<-as.data.frame(LondonWards[,c("PctOwned11","PctNoEng11")])

attach(mydata)

#- check variable distributions first

histplot <- ggplot(data=mydata, aes(x=PctOwned11))

histplot+geom_histogram()

histplot <- ggplot(data=mydata, aes(x= PctNoEng11))

histplot+geom_histogram()


# run a k-means to find 3 clusters - use 25 iterations

fit <- kmeans(mydata, 3, nstart=25) # 3 cluster solution

# get cluster means

centroid<-aggregate(mydata,by=list(fit$cluster),FUN=mean)

#print the results of the cluster groupings

Centroid


# as we only have variable two dimensions we can plot the clusters
on a graph

p<-ggplot(mydata,aes(PctOwned11, PctNoEng11))

p+geom_point(aes(colour=factor(fit.cluster)))+geom_point(data=centro
id[,2:3],aes(PctOwned11, PctNoEng11), size=7, shape=18)+
theme(legend.position="none")


#add the cluster groups to the LondonWards data frame

LondonWards$cluster<-mydata$fit.cluster
```

Extension 1 – Other Summary Spatial Indices

The measurement of spatial patterns using summary indexes: measures of evenness (Diversity, Dissimilarity, Entropy), exposure (Isolation), concentration (Delta), centralization (Distance) & clustering.

Thanks to Professor Phil Rees of the University of Leeds for producing the original summary spatial indices exercises from which much of this has been based.

This section will explain how you could calculate a series of different spatial indices for Wards in London using R

Table 1 – An example of a population data matrix

WardCode	Wardcode	Aged0_15	Aged16_64	Aged65plus	PopCensus2011
AA	E09000001	620	5720	1035	7375
00ABFX	E05000026	3125	9021	640	12786
00ABFY	E05000027	2742	6565	1078	10385
00ABFZ	E05000028	2937	7514	1094	11545
00ABGA	E05000029	2456	6001	1564	10021
00ABGB	E05000030	2232	6731	1543	10506
00ABGC	E05000031	3129	7387	1108	11624
00ABGD	E05000032	3922	7774	756	12452
00ABGE	E05000033	2993	7144	1130	11267
⋮	⋮	⋮	⋮	⋮	⋮
00BKGU	E05000648	2784	8768	1207	11478
00BKGW	E05000649	837	8584	1154	10342
London	London	1624768	5644424	904749	8173941

Notation

Let us try to formalise our description of the population data matrix by using some very simple algebra. If you do not understand the algebra to begin with, this is not vital. In many cases it is possible to follow a verbal description of how to compute an indicator or to follow a recipe for using formulae in a spreadsheet. However, algebraic formulae are frequently used in population analysis, and you need to be able to translate what they mean in terms of how to compute the new variable(s) they generate.

We can define such a population matrix as follows:

Let P represent a count of people or a population.

Let i represent an index for an area such as a London ward. Let e represent an index for a population group, e.g. an ethnic group such as Chinese, or an age group such as 0-15 years old, or a housing tenure group such as owner occupied.

Let f represent an index for another population group (needed when we compare two groups).

Let $+$ represent the summation of a variable over the index it replaces.

We can now construct some general algebraic variables from these building blocks. We use the indexes (of areas or groups or both) as subscripts attached to the main count variables. Subscripts are represented by smaller characters placed a half line below and to the right of the main variables.

For example, let P_{ie} be the people living in residential area i and who are members of population group, (age or ethnic group or housing tenure group, etc) e.

We will want to define the ranges of values of the indexes.

Let n be the number of areas that we are interested in (e.g. $n = 625$ for London wards + the city of London).

Let m be the number of groups in the populations (e.g. $m = 3$, number of broad age groups in the London Wards Data file).

Some Summary Variables

We can now derive some summary variables.

Let P_{i+} be the sum over all ethnic indexes from $e = 1$ to $e = m$ of the population living in area i. Let P_{+e} be the sum over all area indexes from $i = 1$ to n of the population belonging to ethnic group e.

To represent this summation, we can use the Greek symbol $\sum$ (the Greek capital letter sigma):

$$P_{i+} = \sum_{e=1,m} P_{ie}$$

$$P_{+e} = \sum_{e=1,n} P_{ie}$$

where $e = 1, m$ and $i = 1, n$ indicate the start and end values of the index being summed, with all the intermediate values implicitly included.

The Data Matrix

We can arrange these populations in the form of Table 2 with the rows as areas and the population groups as columns. The symbol $\cdots$ (an ellipsis) means that there are intermediate terms in the tables, which are not explicitly shown.

Table 2: A table of area and population group populations

	Population Groups						
Areas	**Group1**	**Group2**	$\cdots$	**Group e**	$\cdots$	**Group m**	**Totals**
Ward 1	P_{11}	P_{12}	$\cdots$	P_{1e}	$\cdots$	P_{1m}	P_{1+}
Ward 2	P_{21}	P_{22}	$\cdots$	P_{2e}	$\cdots$	P_{2m}	P_{2+}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
Ward i	P_{i1}	P_{i2}	$\cdots$	P_{ie}	$\cdots$	P_{im}	P_{i+}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
Ward n	P_{n1}	P_{n2}	$\cdots$	P_{ne}	$\cdots$	P_{nm}	P_{n+}
Totals	P_{+1}	P_{+2}	$\cdots$	P_{+e}	$\cdots$	P_{+m}	P_{++}

The interior of the table is the data matrix; with addition of row totals in the rightmost column and column totals in the bottom row, we obtain a table. The row totals are the total populations of the areas of all ethnicities. The column totals are the total populations of the ethnic groups in all areas being studied. The small areas add up to a bigger area and these totals are for the bigger area. An example of such a data matrix is shown in Table 1: the populations of London, UK wards by 3 age groups in the 2011 Census.

We will use this algebraic framework to derive indicators from the population counts. We will derive the following indicators:

R = row percent (a measure of the share that an population group has of an area's total population)

C = column percent (a measure of the share a population group has of the population of the study region)

LQ = location quotient (a measure of the concentration of an population group in an area relative to its concentration in the whole study region)

Div = a measure of the diversity of an area in terms of its population mix

E = entropy index

The Composition of Areas in Terms of Groups: Row Percentages

The first indicator to derive is the percentage of a ward's population that belongs to a population group. Using the notation outlined earlier, this can be defined as:

$$R_{ie} = 100 * (\frac{P_{ie}}{P_{i+}})$$

where R_{ie} is the row percentage for ward i and population group e. This indicator is computed for all m groups in each of the n areas.

While we won't use these quite yet, column percentages and proportions can also be calculated:

$$C_{ie} = 100 * (\frac{P_{ie}}{P_{+e}})$$

The in R code for calculating row percentages for the age group values in Table 1 would be as follows:

```
LondonWards$pctAged0_15 <-
100*(LondonWards$Aged0_15/(LondonWards$Aged0_15+LondonWards$Aged16_6
4+LondonWards$Aged65plus))
```

Or if the age group data are in columns 5 to 7 of our dataframe we can refer to the column index using square brackets, where the syntax is [row,column] and a range can be referred to using a colon, e.g. [,1:20] refers to columns 1 to 20 and [4:8,] refers to rows 4 to 8:

```
LondonWards$pctAged0_15  <-
100*(LondonWards[,5]/rowSums(LondonWards[,5:7]))
```

To calculate column sums, we can use something like:

```
AgeSum1<-colSums(LondonWards[5:7])
```

Row Proportions, Index of Diversity and Entropy Index for Areas

Row proportions are defined as follows:

$$r_{ie} = (\frac{P_{ie}}{P_{i+}})$$

These can be computed for Table 1 in R using very similar formulae to those described above: column proportions - c_{ie} – would follow the same convention.

Index of Diversity

The **Index of Diversity** measures how diverse is the mix of population groups within an area.

The following description is taken from an extremely interesting Atlas produced using the racial and ethnicity information from the 2000 US Census of Population.

"The diversity index reports the percentage of times two randomly selected people would differ by race/ethnicity. Working with percents expressed as ratios (e.g. 63 per cent = 0.63), the index is calculated in three steps. A. Square the percent for each group. B. Sum the squares, and C. Subtract the sum from 1.00." Brewer and Suchan (2001), p.22.

We can convert this verbal recipe for calculation into the following:

$$Div_i = 1 - \sum_{e=1,m} ({r_{ie}}^2)$$

Some simple R code which would calculate a Diversity Index for Table 1 (assuming we are going to store the results of the squared proportions for columns 5 to 7 in columns 78 to 80 of our data frame) would be:

```
LondonWards[78] <- (LondonWards[,5]/rowSums(LondonWards[,5:7]))^2
LondonWards[79] <- (LondonWards[,6]/rowSums(LondonWards[,5:7]))^2
LondonWards[80] <- (LondonWards[,7]/rowSums(LondonWards[,5:7]))^2
LondonWards$Div<-1-(rowSums(LondonWards[,78:80]))
```

This code could, however, be generalised further and turned into a generic function for calculating an index of diversity – see the location quotient function above for ideas…

Entropy Index

Another indicator of interest is the Entropy Index which is an alternative measure of diversity and is sometimes called the Shannon index (White, 1986).

The equation for Entropy (H) is:

$$H = \left(- \sum_{e=1,m} r_{ie} \ln(r_{ie}) \right) / \ln(m)$$

When there is only one group in the population then the index will be zero. The maximum is attained when all the ethnic groups are equally present and is calculated as 1.

If you look down the values that you have created in this Entropy column you will notice some error entries. This is caused by values of zero in your row proportion cells and the natural logarithm of zero cannot be calculated. To avoid this problem all zeros must be altered to 0.000001. This can either be carried out directly in R, or in some pre-processing using Excel.

Other Spatial Indices:

For the next set of summary indices, we can also define the following variables:

Let X_i = the easting coordinate of the centre of zone i.

Let Y_i = the northing coordinate of the centre of zone i.

Let A_i = the area of zone i.

Let d_{ij}= the distance between the centre of zone i and the centre of zone j.

The centres are defined as the centres of gravity of the areas of the zones. In the UK case population weighted centres are also often with the census data. These are computed from information about the distribution of population within a zone (e.g. OA populations within a ward; postcode populations within an OA; household populations within an OA). In practice, the choice of geographically weighted or population weighted centre does not make a great deal of difference in summary measure computations. In other situations, it might be more sensible to choose one or the other.

We can compute the distances between zone centres by using the Euclidean distance formula, based on Pythagoras' theorem that the square of the hypotenuse of a right angled triangle is equal to the sum of the squares of the other two sides.

That is:

$$d_{ij} = \sqrt{\left\{\left[X_i - X_j\right]^2 + \left[Y_i - Y_j\right]^2\right\}}$$

[If you don't understand the derivation of this distance formula, then consult a simple geometry text.]

The Index of Dissimilarity formula

The general definition of an index of dissimilarity, D, between two population groups, e and f, is as follows:

$$D_{ef} = {}^{1}/_{2} \sum_{j} \left| 100(P_{ie}/P_{+e}) - 100\left(P_{if}/P_{+f}\right) \right|$$

where the vertical lines around the right hand expressions represent the absolute value function, which returns a positive value for both positive and negative numbers. What the formula does is to subtract from the percentage of group e living in area i the percentage of group f that resides in area i. The differences may be either negative or positive and the absolute value function converts them all into positive numbers. The summation sign means that all the area i values are summed. Finally, the sum is multiplied by ½ in order to fix the range of the index between 0 and 100. The range would otherwise be from 0 to 200. A value of 0 indicates that there is no dissimilarity between the distributions in their relative shares across the areas, while 100 indicates that the two distributions are completely dissimilar. Note that if we swap e and f in the above formula, we get the same index value. The index is symmetric with respect to the groups. The index of dissimilarity measures the percentage of one group that would need to move to produce exactly similar geographic distributions for the two groups.

Note that the expressions $100(P_{ie}/P_{+e})$ and $100\left(P_{if}/P_{+f}\right)$ are the "column percentages" shown earlier.

Index of Dissimilarity examples:

A couple of examples will help show you how the index works. Study these so you understand how the calculations are done. In the calculations that you do, there is no need to put into the table the counts and column percentages, which have been previously computed. You will just have columns showing the absolute values of the differences in column percents between the ethnic groups. Care must be taken, of course, in referencing in the cell formulae back to the right cells containing the previous information.

Table 3 – An example of two groups with dissimilar distributions

Area	Group e: counts	Group f: counts	Group e: %	Group f: %	Diff. in %	Absolute difference
Area 1	7	2	37.	12.	25.	**25.**
Area 2	7	2	37.	12.	25.	**25.**
Area 3	2	7	12.	37.	-	**25.**
Area 4	2	7	12.	37.	-	**25.**
Total	20	20	100.	100.	0.	**100.**
Index of Dissimilarity						**50.**

These two groups have the same totals but different distributions. The index of dissimilarity is 50. A second example shows two groups with very similar distributions.

Table 4 – An example of two groups with similar distributions

Area	Group	Group	%	%	%e -%f	abs(diff
Area 1	20	60	20	20	0	**0**
Area 2	20	60	20	20	0	**0**
Area 3	20	60	20	20	0	**0**
Area 4	40	12	40	40	0	**0**
Total	10	30	10	10	0	**0**
Index of Dissimilarity						**0**

Although there are different numbers in group e from group f, their relative distribution across the residential areas is the same. Hence the index value of 0.

THE INDEX OF EXPOSURE

Another aspect of the comparison between two population groups is the degree of exposure that one group has to members of the other group. What is the average percentage of group e in an area that members of group f are exposed to? What is the average percentage of group f in an area that group e are exposed to? In the case of exposure indices, you usually get a different result depending on which group is being exposed to which. The exposure indices are asymmetric ($E_{ef} \neq E_{fe}$) whereas the indices of dissimilarity are symmetric ($D_{ef} = D_{fe}$). The exposure of group e to group f across a set of residential areas can be defined as follows:

$$E_{ef} = \sum_j \left[(P_{ie}/P_{+e}) \times 100 \left(P_{if}/P_{i+} \right) \right]$$

For each area the percentage of the population made up of members of group f is computed (row percentage) and multiplied by the proportion of group e's population resident in that area (column proportion). The values are summed for all areas to give an average percentage of group f to which group e is exposed. Note that if we swap e and f in the formula we get a different result. Table 5 shows examples of computation of exposure indices.

Table 5: Examples of exposure index computation

Group e's exposure to group f						
Area	Group e	Group f	Total	Prob(e)	% f	Prob(e) × % f
Area 1	150	25	175	.375	14	5.25
Area 2	150	25	175	.375	14	5.25
Area 3	50	75	125	.125	60	7.50
Area 4	50	75	125	.125	60	7.50
Total	400	200	600	1.000	index	25.50
Group f's exposure to group e						
Area	Group e	Group f	Total	Prob(f)	% e	Prob(f) × % e
Area 1	150	25	175	.125	86	10.75
Area 2	150	25	175	.125	86	10.75
Area 3	50	75	125	.375	40	15.00
Area 4	50	75	125	.375	40	15.00
Total	400	200	600	1.000	index	51.50

Group e is exposed on average to area populations with 26% group f members while group f is exposed on average to area populations with 52% group e members.

THE DELTA INDEX

One aspect of the distribution of population groups which deserves measurement is the degree to which they are spread over the territorial space of the study region. One group may occupy only a small part of the total area of the study region while another may occupy a large proportion of the region. A simple measure of this spatial concentration, called "delta", was proposed originally by Hoover (1941) (cited by Massey and Denton 1988, p.289). This is just the Index of Dissimilarity between the population group and land area:

$$DELTA = {}^{1}/_{2} \sum_{j} \left| 100(P_{ie}/P_{+e}) - 100(A_j/A_+) \right|$$

Where A_i is the land area of residential zone i, while $A_{+\ *}$ is the total land area in the study region.

To compute this DELTA Index, use the area information contained in your London Wards data file

CENTRALIZATION

The last dimension of spatial patterning which will be discussed in detail is the degree of centralization of a group. In cities in the UK and US newcomers have traditionally located in the inner city, where cheap housing was available for those arriving with limited resources. Of course, that is not true of all groups (e.g. Brits in the US, Yanks in the UK) and it is has been changing as city centre living has revived. In London, for example, there have been increases in numbers of young whites in some gentrifying inner boroughs.

A simple index to compute is the average distance from a central point recorded by a population group.

Let d_{ik} be the distance from residential zone i to central location k. The average distance for population group e is:

$$DAVE_e = \sum_i d_{ik} \times \left(\frac{P_{ie}}{P_{+e}} \right)$$

To compute the distance to the centre, set up a spreadsheet with the Eastings and Northings for each ward in the London UK Study Region. Assume that the city centre is located at The City of London.

Then compute the distance from city centre to each ward by using the formula described earlier.

References:

Brewer, C. and Suchan, T. (2001) *Mapping Census 2000: The Geography of U.S. Diversity*. U.S. Census Bureau, Census Special Reports, Series CENSR/01-1. U.S. Government Printing Office, Washington, DC.

Coleman, D. and Salt, J. (1996) *Ethnicity in the 1991 Census. Volume 1: Demographic Characteristics of the Ethnic Minority Populations*. HMSO, London.

Johnston, R., Poulsen, M. and Forrest, J. (2003) Ethnic residential concentration and a 'New Spatial Order': exploratory analyses of four United States Metropolitan Areas, 1980-2000. *International Journal of Population Geography*, 9(1), 39-56.

Karn, V. (ed.) (1996) *Ethnicity in the 1991 Census. Volume 4: Employment, Education and Housing among Ethnic Minority Populations of Britain.* HMSO, London.

Peach, C. (ed.) (1996) *Ethnicity in the 1991 Census. Volume 2: The Ethnic Minority Populations of Great Britain*. HMSO, London.

Ratcliffe, P. (ed.) (1996) *Ethnicity in the 1991 Census. Volume 3: Social Geography and Ethnicity in Britain: Geographical Spread, Spatial Concentration and Internal Migration*. HMSO, London.

Rees, P. and Butt, F. (2004, in press) Ethnic change and diversity in England, 1981-2001.
Are
a.

Rees, P., Phillips, D. and Medway, D. (1995) The socioeconomic geography of ethnic groups in two northern cities. *Environment and Planning A*, 27, 4, 557-591.
White, M.J. (1986) Segregation and Diversity Measures in Population Distribution. *Population Index* 52(2): 198-221.

Hoover, E.M. (1941) Interstate redistribution of population, 1850-1940. *Journal of Economic History*, 1, 199-205.

Duncan, O.D. and Duncan, B. (1955) A methodological analysis of segregation indices. *American Sociological Review*, 20, 210-217.

White, M.J. (1986) Segregation and diversity measures in population distribution. *Population Index*, 52(2), 198-221.

Massey, D.S. and Denton, N.A. (1988) The dimensions of residential segregation. *Social Forces,* 67(1), 281-315.

Massey, D.S., White, M.J. and Phua, V-C. (1996) The dimensions of segregation revisited. *Sociological Methods and Research*, 25(2), 172-206.

On residential profiles

Johnson, Poulsen and Forest have produced a large number of papers in which they use cumulative concentration profiles to compare the spatial concentration patterns of different racial or ethnic groups.

Johnston, R., Poulsen, M. and Forrest, J. (2003) Ethnic residential concentration and a 'New Spatial Order': exploratory analyses of four United States Metropolitan Areas, 1980-2000. *International Journal of Population Geography*, 9(1), 39-56.
[Note: This journal paper is accessible to you on-line because the University has a subscription to the electronic version.]

Summary measures used to monitor the evolution of spatial polarization over time

Dorling, D. and Rees, P. (2003) A nation still dividing: the British census and social polarisation 1971-2001. *Environment and Planning A*, 35(7), 1287-1313.