Mapping and Measuring Country Shapes

The cshapes Package

by Nils B. Weidmann and Kristian Skrede Gleditsch

Abstract The article introduces the **cshapes** R package, which includes our CShapes dataset of contemporary and historical country boundaries, as well as computational tools for computing geographical measures from these maps. We provide an overview of the need for considering spatial dependence in comparative research, how this requires appropriate historical maps, and detail how the **cshapes** associated R package **cshapes** can contribute to these ends. We illustrate the use of the package for drawing maps, computing spatial variables for countries, and generating weights matrices for spatial statistics.

Introduction

Our CShapes project (Weidmann et al., 2010) provides electronic maps of the international system for the period 1946–2008 in a standard Geographic Information System (GIS) format. Most existing GIS maps consider only contemporary country configurations and do not include information on changes in countries and borders over time. Beyond the value of more accurate historical maps in their own right for visualization, such maps can also help improve statistical analyses. Comparative research in the social sciences often considers country-level characteristics such as total population, economic performance, or the degree to which a country's political institutions can be characterized as democratic. Many theories relate such outcomes to other characteristics of individual countries, and typically ignore how possible interdependence between countries may also determine outcomes. However, over recent years there has been more attention to the spatial relationship between observations. For example, the level of democracy in a country can be strongly influenced by how democratic its neighbors are (Gleditsch, 2002). Similarly, conflict diffusion across national boundaries implies that the risk of civil war in a country will be partly determined by conflict in its immediate proximity (Gleditsch, 2007). Tobler's (1970, 236) *First Law of Geography* states that "everything is related to everything else, but near things are more related than distant things". To the extent that this applies to the international system, we need to–at a theoretical level–specify the transnational mechanisms that cause these interdependencies, and–at an analytical level–our empirical techniques must properly incorporate spatial interdependence among observations.

The CShapes dataset and the R package **cshapes** presented in this article help address the second task. Assessing spatial dependence in the international system requires measures of the connections and distances between our units of analysis, i.e., states. There exist various data collections that provide different indicators of spatial interdependence: a contiguity dataset (Gochman, 1991), the minimum-distance database by Gleditsch and Ward (2001), a dataset of shared boundary lengths (Furlong and Gleditsch, 2003), and distances between capitals (Gleditsch, 2008). However, although all of these collections entail indicators from a map of the international system at a given point in time, these databases provide the final indicators in fixed form rather than directly code the information from the underlying maps.

Our CShapes project takes a different approach and provides electronic maps of the international system for the period 1946–2008 from which such geographical indicators can be coded directly. The R package **cshapes** allows computing important distance measures directly from these maps. This approach has several advantages over the existing spatial indicators mentioned above. First, it ensures that the computed indicators are comparable, since they all refer to the same geopolitical entities. Second, maintaining and updating the dataset becomes much more convenient: If a change occurs in the international system – such as for example the independence of Montenegro in 2006 – we only need to update the electronic maps and re-run the computations in order to get up-to-date distance measures as of 2006. Third, the availability of electronic maps opens up new possibilities for visualization of country-level variables.

In the remainder of this article we provide a brief overview of the underlying spatial dataset of country boundaries, describing our coding approach and the data format. We then turn to the **cshapes** package for R, and describe three main applications the package can be used for: (i) visualizing country data as maps, (ii) computing new spatial indicators for countries, and (iii) creating weights matrices for spatial regression models.

The CShapes Dataset

Coding Criteria

Creating the CShapes dataset requires us to define coding rules with respect to three questions: (i) what constitutes an independent state, (ii) what is its spatial extent, and (iii) what constitutes a territorial change to a state. The following paragraphs summarize our approach. For a more detailed discussion,

we refer the reader to Weidmann et al. (2010).

What is a state? There are various criteria for defining states. Research in Political Science often relies on the list of independent states provided by the *Correlates of War, 2008* (COW) project. This list relies on several criteria, including recognition by the UK and France, membership in the League of Nations and the United Nations, as well as other ad-hoc criteria. An alternative *Gleditsch and Ward* (GW) (1999) list of independent states addresses some limitations of the COW list, and has also gained a wide acceptance in the research community. As a result, we have made CShapes compatible with both the COW and GW state lists.

What is the spatial extent of a state? In many cases, the territory of a state comprises several (and possibly disjoint) regions with different status. For colonial states, an obvious distinction can be made between the "homeland" of a state and its dependent territories or colonies. Territory can also be disputed between states, such as for example the Golan Heights in the border region between Syria and Israel, or territory claimed may not be directly controlled or accepted by other states, as in the case of claims in Antarctica. As there are many such cases it is often impossible to allocate particular territories to states. To avoid this difficulty, we only code the core territories of state, without dependent territories or disputed regions.

What constitutes a territorial change? Given the spatial extent of a state, we need to define the territorial changes that should be reflected in the dataset. Our data encompass three types of changes are given in the dataset. First, new states can become independent, as in the case of the the secession of Eritrea from Ethiopia in 1993. Second, states can merge, as in the case of the dissolution of the (Eastern) German Democratic Republic in 1990 and its ascension to the (Western) German Federal Republic. Third, we may have territorial changes to a state that do not involve the emergence or disappearance of states.

CShapes includes all changes of the first two types. To keep the effort manageable, we only code changes of the third type that (i) affect the core territory of a state, and (ii) affect an area of at least 10,000 square kilometers. Our coding here is based on an existing dataset of territorial changes by Tir et al. (1998).

Implementation

Geographic data are typically stored in one of two formats: raster and vector data. Raster data represent continuous spatial quantities, such as territorial elevation, while vector data are used to capture discrete spatial entities ("features"), such as cities, rivers or lakes. There are three basic types of vector data: points, lines and polygons. The latter are the most appropriate for representing state boundaries. Apart from the spatial features themselves, vector data formats store non-spatial information related to them in an *attribute table*. For example, in a dataset of point features representing cities, we could store the name and population of each city in the attribute table.

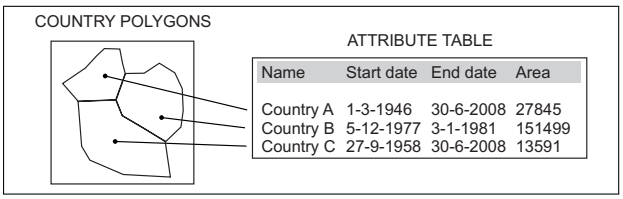

Figure 1: CShapes vector data representation as polygons with lifetimes (from Weidmann et al., 2010).

In CShapes, we use polygons to represent state boundaries at a given time, and supplement the temporal information for these polygons in the attribute table. This is done by assigning a lifetime (i.e., a start day and an end day) to each polygon. By filtering the "active" polygons for a given date, it is possible to get a snapshot of the international system for this date. Figure 1 illustrates the CShapes data representation with each polygon's lifetime in the attribute table. The attribute table also contains the country identifier according to the COW and GW lists, as well as the location of the state capital in latitude/longitude coordinates. This information is required for the calculation of capital distances between states (see below). The CShapes vector dataset is provided in the shapefile format (ESRI, 1998), which can be used directly in standard GIS applications. However, in order to provide a more accessible interface to the data and to enable additional computations, we have developed the **cshapes** R package introduced in the next section.

The cshapes R Package and Applications

The **cshapes** package for R has two main purposes: first, to make the CShapes data available for use within R, and second, to perform additional operations on the data, in particular weights matrix computation for spatial regression modeling. In the following, we introduce the basic functionality of the package and later demonstrate its use through three illustrative applications.

Accessing the Data

In R, basic support for spatial data is provided by the **sp** package, which includes a set of classes both for vector and raster data. Other packages add functionality to work with these classes, as for example the **maptools** package for reading/writing shapefiles, drawing maps, or the **spdep** package for spatial regression. For an introduction to R's spatial features, we refer to the overview in Bivand (2006), and to Bivand et al. (2008) for a detailed introduction. **cshapes** draws on these packages in making the historical country boundaries available for use within R. The `cshp()` function is a convenience method to load the spatial dataset and returns a `SpatialPolygonsDataFrame`, which can be used to access and modify both the polygons and the attached attribute table. As an example to illustrate this, we first demonstrate how to extract country boundaries for a given point in time, and later use these data to visualize data, in this case the regime type of countries.

```
# load country border as of 2002
> cmap.2002 <- cshp(date=as.Date("2002-1-1"))
```

We can now join the extracted polygons for this date with supplementary information about countries. The Polity IV database (Marshall and Jaggers, 2008) provides a summary indicator about regime type. The indicator ranges from -10 to 10, with low values corresponding to autocratic regimes, and high values corresponding to democratic systems. For the example below, the Polity dataset is downloaded and filtered to retain only the 2002 values. Furthermore, the `polity.2002` data frame contains only two columns, the COW country identifier (ccode) and the regime type indicator (polity2). In order to match this information, we first need to match country identifiers from cshapes with those in the Polity dataset. As the Polity data is not available for all countries in 2002, we do not get a regime type value for each country in `cmap.2002`.

```
# download dataset, filter scores for 2002
> polity.file <- paste(tempdir(),
+ "p4v2008.sav",  sep="/")
> download.file("http://www.systemicpeace.org/
+ inscr/p4v2008.sav", polity.file)
> polity <- read.spss(polity.file, to.data.frame=T)
> polity.2002 <- polity[polity$year==2002,]
> polity.2002 <- subset(polity.2002,
+ !is.na(polity2), select=c(ccode, polity2))

# match country identifiers from both datasets
> o <- match(cmap.2002$COWCODE, polity.2002$ccode)

# order polity dataset accordingly
polity.2002 <- polity.2002[o,]

# set row names, required for spCbind function
row.names(polity.2002) <- cmap.2002$FEATUREID

# append using spCbind
cmap.2002.m <- spCbind(cmap.2002, polity.2002)
```

Drawing Maps

The above created dataset with appended regime type scores for 2002 can now be used for visualizing regime type across space. As described in Bivand et al. (2008, ch. 3), the **classInt** package is useful for creating maps with color shadings. The literature commonly distinguishes between three regime types: autocracies (Polity scores -10 through -7), anocracies (scores -6 through 6) and democracies (scores 7 through 10), a typology we employ for our example. Using the `classIntervals()` and `findColors()` functions, we divide regime type scores into three different classes and assign colors of different intensity, where more autocratic countries are displayed using darker colors. The following example shows the code necessary to generate the map in Figure 2.

```
# generate a color palette
> pal <- grey.colors(3, 0.25, 0.95)

# find the class intervals and colors
> breaks <- classIntervals(cmap.2002.m$polity2,
+ n=3, style="fixed", fixedBreaks=c(-10, -6, 7, 10))
> colors <- findColours(breaks, pal)

# create plot and add legend
> plot(cmap.2002.m, bty="n", col=colors)
> legend(x="bottomleft",
+ legend=c("Autocracy", "Anocracy", "Democracy"),
+ fill = attr(colors, "palette"),
+ bty = "n", title="Regime type")
```

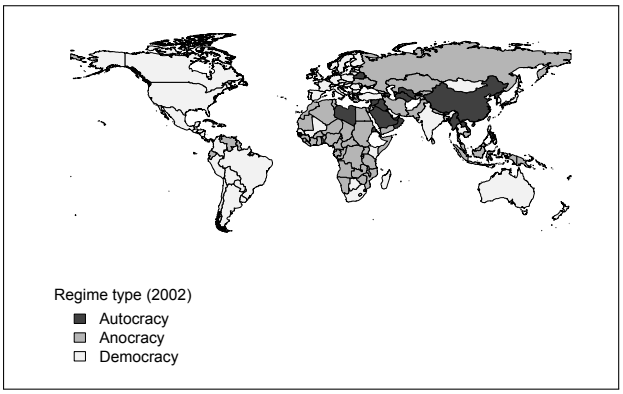

Figure 2: Mapping regime type scores according to the Polity dataset for 2002.

Spatial Variables for Countries

As an example of spatial characteristics that can be calculated using **cshapes** we consider the distance between a country's capital and its geographical center. The extent to which a country contains extensive peripheries or hinterlands is often held to be an

important influence on state consolidation and the risk of conflict (Herbst, 2000; Buhaug et al., 2008). A simple measure of unfavorable geography can be derived from comparing the position of a state capital relative to the geographic center of a country. Figure 3 shows the borders of two African countries, the Democratic Republic of the Congo and Nigeria, as well as their capitals and their polygon centroids. In the Democratic Republic of the Congo (left), the capital Kinshasa is located in the far West of the country, which creates large hinterlands in the East. Nigeria (right) has a more balanced configuration, with the capital Abuja located close to the centroid.

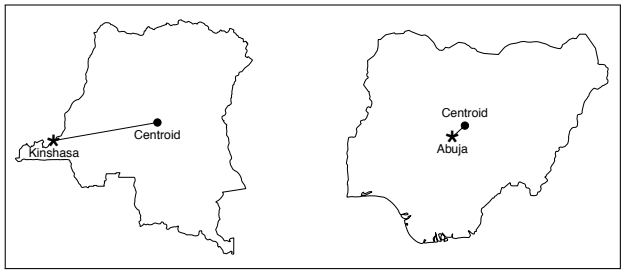

Figure 3: Illustration of the measure of deviation between the geographic centroid of a polygon and the capital. Left: Democratic Republic of Congo, right: Nigeria (from Weidmann et al., 2010).

For a systematic statistical test, we compute the capital–centroid deviation from the capital to the polygon centroid for each country/year. We do not show the complete regression analysis here, but focus on the computation of this indicator. We first obtain the centroids for all polygons using the `coordinates()` function in the **sp** package. We then apply a newly created function called `distance()` to each polygon. This function is essentially a wrapper around the `spDistsN1` function from the **sp** package and computes the distance between each polygons's centroid and its capital. The capital longitude and latitude is obtained directly from the respective polygon's attributes (the CAPLONG and CAPLAT fields). Note that since the CShapes dataset is unprojected, we need to set the `longlat=TRUE` parameter in the `spDistsN1` function in order to obtain great circle distances.

```
> cmap <- cshp()
> centroids <- coordinates(cmap)
> distance <- function(from.x, from.y, to.x, to.y)
+ from <- matrix(c(from.x, from.y), ncol=2, nrow=1)
+ spDistsN1(from, c(to.x, to.y), longlat = TRUE)
+
> cmap$capcentdist <- mapply(distance, centroids[,1],
+ centroids[,2], cmap$CAPLONG, cmap$CAPLAT)
```

Computing Country Distances

Spatial statistical applications typically require that we first specify spatial structure. One way to represent this is by means of a matrix indicating adjacency or distance between any two observations i, j. A matrix of this kind can then be transformed into a weights matrix that assigns a higher weight to spatially close observations. As discussed above, there are many advantages to computing these distances directly from a spatial dataset such as CShapes, which is why we provide a dynamic way of distance computation in our package, rather than offering pre-computed distances. This also makes it possible to compute these matrices for arbitrary points in time.

With countries as the unit of analysis, distance matrices are typically based on one of three types of distance (Ward and Gleditsch, 2008). First, we can consider the distance between the capital cities of two countries. Second, rather than capitals, we can use the geographical center point of a country for distance computation. This center point or centroid is the country's geometric center of gravity. Third, we can measure the minimum distance between two countries, i.e. the distance between two points, one from each country, that are closest to each other. Gleditsch and Ward (2001) discuss the relative advantages of different measures in greater detail. Computing distances between predetermined points as in the first two alternatives is straightforward; we only need to take into account that the CShapes dataset is unprojected, which is why we compute great circle distances. Finding the minimum distance, however, is more complicated as we need to consider the shape and position for two countries to find the closest points.

The `distmatrix()` function. **cshapes** computes distance matrices of all three types with its `distmatrix()` function. Capital and centroid distances are easy to compute, as there is only one point per polygon used for the computation. Obtaining the minimum distance is more difficult. Given two countries, we need to compute the distance of each point on one of the polygons to all lines in the other polygon. This procedure must be repeated for each pair of countries. Because the computational costs of this procedure, the **cshapes** package implements three ways for speeding up this computation. First, we implement the distance computation in Java, using the **rJava** package for interfacing and the Java Topology Suite[1] (JTS) for handling spatial data in Java. Second, we resort to computing distances between pairs of points in the two polygons, rather than between points and lines. Third, we reduce the number of points in a polygon, which results in fewer comparisons to be made. The polygon sim-

[1] http://www.vividsolutions.com/jts/jtshome.htm

plification is readily implemented in JTS according to the algorithm proposed by Douglas and Peucker (1973). The accuracy of the simplification can be controlled by a threshold parameter that indicates the maximum allowed deviation of the simplified shape from the original one. Figure 4 shows an example of the shapes generated by the polygon simplification, using our default threshold (0.1).

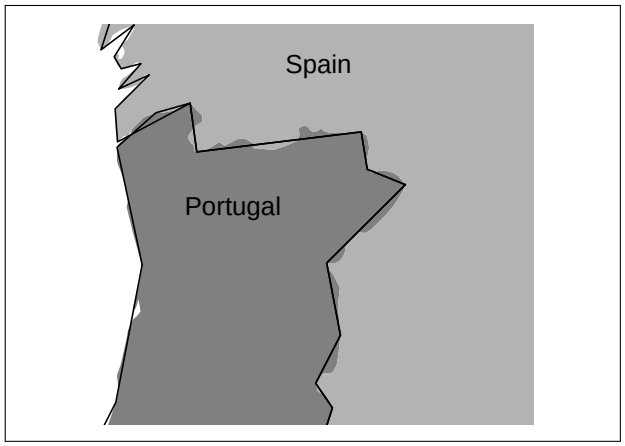

Figure 4: Result of the polygon simplification procedure using the Douglas-Peucker algorithm. The shaded areas show the borders between Portugal and Spain according to the original, detailed dataset. The solid line shows these boundaries after the simplification process, where twisted segments have been replaced by straight ones, while preserving the basic shape of a country.

`distmatrix()` is the **cshapes** function to generate distance matrices. It returns a symmetric matrix $D_{i,j}$, where the entries $d_{i,j}$ correspond to the distance between countries i and j in kilometers, and all $d_{i,j} = 0$. The rows and columns are labeled by the respective country identifier, according to the COW or GW coding system. Since the layout of the international system changes over time, the first parameter of the function specifies the date at which the matrix should be computed. The country coding system to be used is set by the `useGW` parameter, which defaults to the Gleditsch and Ward system. The distance type is specified by the `type` parameter that has three possible values: `capdist`, `centdist` and `mindist`. Furthermore, the `tolerance` parameter sets the threshold for the Douglas-Peucker polygon simplification described above. This parameter only affects minimum distance computation. The default value of this parameter is 0.1, and a value of 0 disables polygon simplification. Below, we compare the execution time obtained with default polygon simplification to no simplification.[2]

```
> system.time(dmat.simple <- distmatrix(
+ as.Date("2002-1-1"), type="mindist"))
   user  system elapsed
223.286   0.842 224.523

> system.time(dmat.full <- distmatrix(
+ as.Date("2002-1-1"), type="mindist",
+ tolerance=0))
    user   system  elapsed
12452.40    48.67 12686.44

> cor(as.vector(dmat.simple),
+ as.vector(dmat.full))
[1] 0.9999999
```

Because of the quadratic scaling of the minimum distance computation, the reduction in computational costs achieved by the simplification procedure is extremely high. Whereas default polygon simplification allows us to compute the complete matrix in less than four minutes, the same procedure on the full dataset takes about four hours. Still, the inaccuracies introduces by the simplification are very small; the computed distance matrices correlate at 0.999.

Computing spatial weights. Once a distance matrix has been computed, we can transform it to different weights matrices, depending on the respective application. Weights matrices based on spatial adjacency assign a weight of 1 to all entries with $d_{i,j} \leq t$ for a distance threshold t. Empirical research suggests that it is helpful for many applications to have a relatively inclusive threshold t (for example, 900 km) to determine relevant neighboring states, in order to prevent excluding states that are not directly contiguous yet in practice often have high levels of interaction (Gleditsch, 2002). Alternatively, we can transform a distance matrix into a continuous weights matrix by inverting the $d_{i,j}$. The following code illustrates how to generate weights from **cshapes** distance matrices.

```
> dmat <- distmatrix(as.Date("2002-1-1"),
+ type="mindist")

# adjacency matrix, 900 km threshold
> adjmat <- ifelse(dmat > 900, 0, 1)
> diag(adjmat) <- 0

# inverse distance
> invdmat <- 1/dmat
> diag(invdmat) <- 0

# save matrix for later use
> write.table(adjmat, "adjmat2002.txt",
+ row.names=T, col.names=T)

# load matrix
> adjmat <- read.table("adjmat2002.txt",
+ header=T, check.names=F)
```

Integration with spdep. **spdep** is one of R's most advanced packages for spatial statistical analysis.

[2] Hardware: Macbook 2.4 GHz Intel Core 2 Duo, 4 GB RAM.

It relies on two basic representations for spatial structure: neighbor lists for discrete neighborhood relationships, and weights lists for storing spatial weights between observations. Using functions from the **spdep** package, it is possible to convert the matrices created by **cshapes**. The following code demonstrates how to do this.

```
# neighbor list from adjacency matrix:
# create a weights list first...
> adjmat.lw <- mat2listw(adjmat,
+ row.names=row.names(dmat))

# ... and retrieve neighbor list
> adjmat.nb <- adjmat.lw$neighbours

# weights list from weights matrix
> invdmat.lw <- mat2listw(invdmat,
+ row.names=row.names(invdmat))
```

The neighbor lists obtained from **cshapes** can then be used to fit spatial regression models using **spdep**.

Conclusions

Accurate historical maps are essential for comparative cross-national research, and for incorporating spatial dependence into applied empirical work. Our CShapes dataset provides such information in a standard GIS format for the period 1946–2008. In this paper, we have introduced our **cshapes** package that provides a series of tools for useful operations on these data within R. Our package supplements the existing valuable packages for spatial analysis in R, and we hope that the package may facilitate greater attention to spatial dependence in cross-national research.

Bibliography

R. S. Bivand. Implementing spatial data analysis software tools in R. *Geographical Analysis*, 38(2006):23–40, 2006.

R. S. Bivand, E. J. Pebesma, and V. Gómez-Rubio. *Applied Spatial Data Analysis with R*. Springer, New York, 2008.

H. Buhaug, L.-E. Cederman, and J. K. Rød. Disaggregating ethnic conflict: A dyadic model of exclusion theory. *International Organization*, 62(3):531–551, 2008.

Correlates of War Project. State system membership list, v2008.1, 2008. Available online at http://correlatesofwar.org.

D. Douglas and T. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10(2):112–122, 1973.

ESRI. ESRI shapefile technical description, 1998. Available at http://esri.com/library/whitepapers/pdfs/shapefile.pdf.

K. Furlong and N. P. Gleditsch. The boundary dataset. *Conflict Management and Peace Science*, 29 (1):93–117, 2003.

K. S. Gleditsch. *All Politics is Local: The Diffusion of Conflict, Integration, and Democratization*. University of Michigan Press, Ann Arbor, MI, 2002.

K. S. Gleditsch. Transnational dimensions of civil war. *Journal of Peace Research*, 44(3):293–309, 2007.

K. S. Gleditsch. Distances between capital cities, 2008. Available at http://privatewww.essex.ac.uk/~ksg/data-5.html.

K. S. Gleditsch and M. D. Ward. Measuring space: A minimum distance database and applications to international studies. *Journal of Peace Research*, 38 (6):749–768, December 2001.

K. S. Gleditsch and M. D. Ward. Interstate system membership: A revised list of the independent states since 1816. *International Interactions*, 25:393–413, 1999.

C. S. Gochman. Interstate metrics: Conceptualizing, operationalizing, and measuring the geographic proximity of states since the congress of Vienna. *International Interactions*, 17(1):93–112, 1991.

J. Herbst. *States and Power in Africa: Comparative Lessons in Authority and Control*. Princeton University Press, Princeton, NJ, 2000.

M. G. Marshall and K. Jaggers. Polity IV project: Political regime characteristics and transitions, 1800-2007, 2008. Available at http://www.systemicpeace.org/polity/polity4.htm.

J. Tir, P. Schafer, P. F. Diehl, and G. Goertz. Territorial changes, 1816-1996. *Conflict Management and Peace Science*, 16:89–97, 1998.

W. R. Tobler. A computer movie simulating urban growth in the Detroit region. *Economic Geography*, 46:234–240, 1970.

M. D. Ward and K. S. Gleditsch. *Spatial Regression Models*. Sage, Thousand Oaks, CA, 2008.

N. B. Weidmann, D. Kuse, and K. S. Gleditsch. The geography of the international system: The CShapes dataset. *International Interactions*, 36(1), 2010.

Nils B. Weidmann
Woodrow Wilson School
Princeton University, USA
nils.weidmann@gmail.com

Kristian Skrede Gleditsch
Department of Government
University of Essex, UK
ksg@essex.ac.uk