Collection of Full Papers

การประชุมวิชาการทางคณิตศาสตร์ประจำปี 2552 (ครั้งที่ 14)

The 14th Annual Meeting in Mathematics 2009

5-6 มีนาคม 2552

Collection of Full Papers: 14th Annual Meeting in Mathematics 2009, 5-6 March 2009

GIS Functions in R for Handling Regions

Mayuening Eso, Metta Kuning and Nittaya McNeil

Abstract : There are many methods for presenting statistical information, such as tables, graphs, pictures and maps. Nowadays geographical data is used by many organizations in fields that include epidemiology, biology, agriculture and military, etc. There are many types of software that support analysis of geographical data, such as ArcView, MapInfo and Arc/Info, but the most GIS software are expensive. The use of R program, which is open source (free) would be attractive. This study aimed to create functions, using R program, for management of region boundary data contained in the GIS package. In writing this package, UTM coordinates were used to create maps, using geometrical and database concepts to handle complex regions. The use of these functions in the GIS package makes presentation of geographical data interesting and clear.

Keywords : Complex regions, GIS, GIS package

1 Introduction

A Geographic Information System (GIS) is a tool for displaying and analyzing spatial data. It uses relational a database management system for managing spatial data and attribute data (Demers, 1999). A GIS can be used in a variety of fields such as epidemiology, biology, agriculture as well as in the military. It can show patterns and trends of disease outbreaks, land changing patterns, traffic bottlenecks and natural disasters. There are many types of software that support the analysis and display of spatial data, and many of these are commercially available. R is a software environment for statistical computing and graphics, the design being influenced largely by S, and more recently S-Plus, a widely used, commercially available statistical package. R is available under the terms of the GNU General Public License as published by the Free Software Foundation, in source code format (Bivand, 2002).

In R, there are packages to display maps and analyze spatial data. Levin-Koh and Bivand (2008) created the *maptools* package to manage and display maps from ESRI (Environmental Systems Research Institute) shapefiles. Brownrigg (2008) created the *maps* package which has an archive of various geographical maps. Stabler (2006) created the *shapefiles* package to read and write ESRI shapefiles and *spdep* package was created by Bivand (2008) to create spatial weights matrix objects from polygon contiguities. Most of these packages use ESRI shapefiles to create maps and analyze spatial data. ESRI shapefiles have their own specific format, and users can not modify them. In the analysis of spatial data, users may want to manage boundaries of regions to present their information, for example combining regions with the same geographic characteristic, such as mountains, lakes, rural areas and urban area, or population characteristics, such as religion and! language.

In this study, we aim to create and manage spatial data by using our own functions written in R that read in text files containing Cartesian coordinates of the Universal Transverse Mercator (UTM) system.

2 General Description

The functions must have at least two files to create the map and display the information, namely a spatial data file and an attribute data file. The spatial data contains the coordinates in UTM system. The attribute data file contains the statistical data. Each file must contain a primary key, which can be composite, to uniquely identify each record. Figure 1 shows an example of a spatial data file. It contains two columns *plotID* and *pointID*, representing the primary key, and *coorx* and *coory*, representing the X- and Y-coordinates, respectively. The column called *plotID* represents the region code while the *pointID* represents a sequential index of coordinates in each region. Figure 2 shows another example of spatial data containing islands. Null values (NA) in the columns *coorx* and *coory* are used to separate main regions from islands, lakes or rivers, also known as *complex regions*. Fun! ctions to handle complex regions will be written up elsewhere. Figure 3 shows and example of an attribute data file. In this file *plotID* is the primary key, representing the region code. The column called *name* is the name of the region. The *numEvn* and *numEvngrp* columns are the variables to display on the map. In this example, *numEvn* is the number of terrorist events in each region, while *numEvngrp* is the same data categorized into groups.

3 Functions for Handling Regions

In this study we created 5 functions, namely *create.map*(), *combine.map*(), *area.map*(), *perim.map*() and *colorshade*(). The first function, *create.map*(), is the main function to create the map. Users can specify the color for each region. Figure 4 shows the results of calling this function.

```
plotID   pointID   coorx      coory
940101      1      748125.6   758534.1
940101      2      748030.2   758504.1
940101      3      747937.3   758467.1
   .        .          .          .
   .        .          .          .
   .        .          .          .
940101     81      748125.6   758534.1
---------------------------------------
940102      1      748553.7   759425.3
940102      2      748612.2   759756.3
940102      3      748748.5   760087.4
   .        .          .          .
   .        .          .          .
   .        .          .          .
940102     36      748553.7   759425.3
---------------------------------------
940103      1      749278.6   759369.6
940103      2      749243.5   759276.1
940103      3      749210.8   759181.6
   .        .          .          .
   .        .          .          .
   .        .          .          .
940103     32      749278.6   759369.6
---------------------------------------
```

Figure 1: An example of a spatial data file

The *combine.map*() function allows users to combine different regions into one region. Users can choose the variable as the condition for combining regions. For example, as shown in Figure 5, the map has five regions, which we will call A, B, C, D and E. Suppose the user wants to combine regions A, B and D and use a dark goldenrod color for the final combined region. The panel on the left shows the three regions highlighted with boundaries shown as solid lines. The middle panel shows the common boundaries of the three regions as dotted lines. The panel on the right shows the final region with the dotted line removed.

The *area.map*() function computes the area of the map using the following formula:

$$A = \frac{1}{2}\sum_{i=0}^{N-1}(x_i y_{i+1} - x_{i+1} y_i) \tag{3.1}$$

where A is the area, i is a index for every Cartesian coordinate, and N is the total number of coordinates (Bourke, 1988). The *perim.map*() function computes the perimeter of each region by aggregating the distances between every pair of points, using the following formula:

$$d = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \tag{3.2}$$

where d is the distance, and i is the index for every Cartesian coordinate (LongLey, 1988). Users can compute the area and perimeter of any single region or combination of regions. The *colorshade*() function displays the statistical data on the map. Users can supply a continuous or categorical variable for showing the contrasting region colors. If the user does not identify the color for each

plotID	pointID	coorx	coory
208184	1	688485.8	778128.7
208184	2	688487.8	778128.8
208184	3	688496.6	778126.1
.	.	.	.
.	.	.	.
.	.	.	.
208184	1124	688485.8	778128.7
208184	1125	NA	NA
208184	1126	689098.7	776879.8
208184	1127	689076.9	776866.9
208184	1128	689068.4	776824.8
.	.	.	.
.	.	.	.
.	.	.	.
208184	1377	689098.7	776879.8

Figure 2: Example of a spatial data file with an island

plotID	name	numEvn	numEvngrp
940101	Sabarang	59	3:40+
940102	ArNuhRu	12	1:0-20
940103	ChabangTiKo	4	1:0-20
940104	Bana	50	3:40+
940105	Tanyonglulo	14	1:0-20
940106	KhlongManing	11	1:0-20
940107	Kamiyo	5	1:0-20
940108	Barahom	11	1:0-20
940109	Pakaharang	31	2:21-40
940110	Rusamilae	47	3:40+

Figure 3: Example of an attribute data file

group of categorical variable, this function generates the color automatically. For continuous variables, users can specify the number of distinct categories, and the function will automatically divide the data into groups based on equal ranges. If the number of categories is not specified by the user, the function will determine the most appropriate number based on the total number of regions.

4 Application of Functions

This example uses social unrest data from the four southern-most provinces of Thailand, namely Pattani, Yala, Narathiwas and Songkhla. The regions include all districts of Pattani, Yala and Narathiwas province plus four districts of Songkhla. These districts have been subjected to continuing social unrest over recent years. Figure 6 shows the distribution of violent events in Pattani province. Three contrasting colors were used to display the level of unrest in the area, based on the number of events in each district.

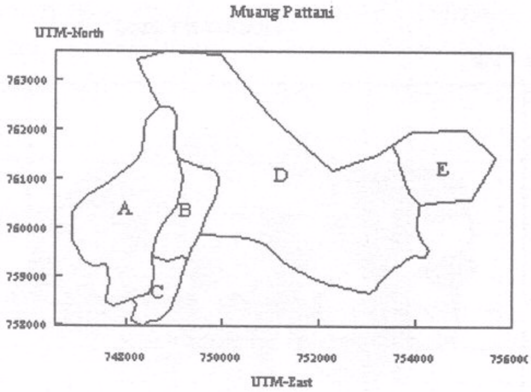

Figure 4: Results of function *create.map*()

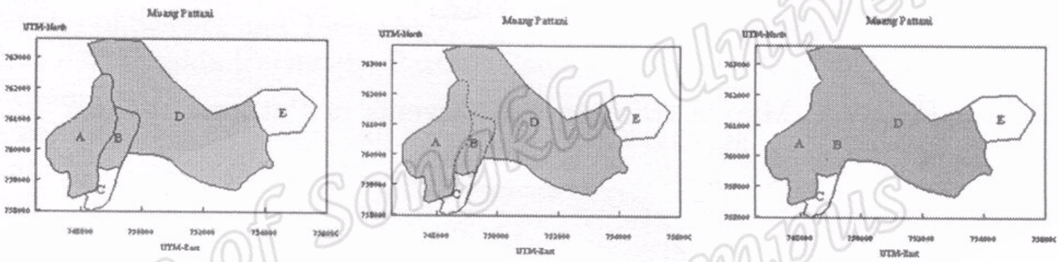

Figure 5: Results of *combine.map*() function

5 Ongoing work

We will continue to develop these functions to be able to manage boundary data of complex regions, such as holes or areas interspersed with smaller regions. For example, as shown in Figure 7, the number of regions is composed of an island (C), two lakes (B and D) and land (A).

Acknowledgement : We would like to thank Professor Don McNeil for his invaluable encouragement and guidance.

References

[1] B. Stabler, Read and Write ESRI Shapefiles, 2006. Available:http://cran.r-project.org/web/packages/shapefiles/shapefiles.pdf [22/7/2551].

[2] J. Verzani, Using R in Introductory Statistics Courses with the pmg Graphical User Interface, *Journal of Statistics Education* bf 16(2008).

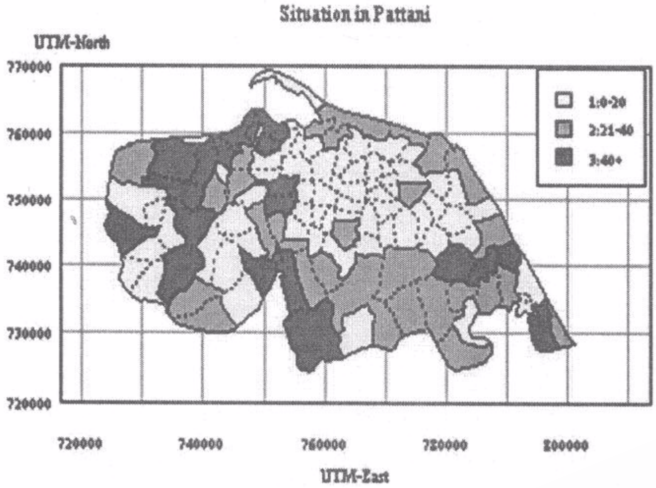

Figure 6: Map showing terrorist events in Pattani province

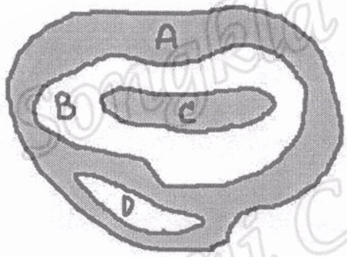

Figure 7: Complex regions

[3] M.N. Demers, *Fundamentals of Geographic Information Systems*, John Wiley & Sons, New York, 1999.

[4] N.J. Lewin-Koh and R. Bivand, Tools for reading and handling spatial objects, 2008. Available:http://cran.r-project.org/web/packages/maptools/maptools.pdf[5/12/2008].

[5] P. LongLey, M. Goodchild, D. Mahuire and D. Rhind, *Geographic Information Systems and Science*, John Wiley & Sons, England, 1988.

[6] P. Bourke, Calculating The Area and Centroid of a Polygon, 1988. Available:http://local.wasp.uwa.edu.au/~pbourke/geometry/polyarea/ [22/7/2551].

[7] R. Bivand, *Implementing spatial data analysis software tools in R*, Economic Geography Section, Department of Economics, Norwegian School of economics and business administration, Bergen, Norway, 2002.

[8] R. Bivand, Spatial dependence: weighting schemes, statistics and models, 2008.
Available:http://cran.r-project.org/web/packages/spdep/spdep.pdf [5/12/2008].

[9] R. Brownrigg, Draw Geographical Maps, 2008.
Available:http://cran.r-project.org/web/packages/maps/maps.pdf [5/12/2008].

Mayuening Eso, Metta Kuning and Nittaya McNeil
Department of Mathematics and Computer Science,
Faculty of Science and Technology,
Prince of Songkla University, Pattani Campus
Pattani 94000.
e-mail : `g5020320016@pn.psu.ac.th`, `Kmetta@bunga.pn.psu.ac.th` and `Nittaya@bunga.pn.psu.ac.th`