Pergamon

Comput., Environ. and Urban Systems, Vol. 21, No. 1, pp. 5–29, 1997

Printed in Great Britain
0198-9715/97 $17.00 + 0.00

PII: S0198-9715(97)00011-2

DEVELOPING USER-FRIENDLY SPATIAL STATISTICAL ANALYSIS MODULES FOR GIS: AN EXAMPLE USING ARCVIEW

Zhiqiang Zhang[1] *and Daniel A. Griffith*
Department of Geography, Syracuse University, Syracuse, NY 13244, U.S.A.

ABSTRACT. *Although it is generally agreed that geographic information systems (GIS) should include more statistical analysis functionalities, the issues of which functionalities should be included and how to integrate statistical analysis with GIS are still widely debated. This paper, based on a brief review of what has been done in this area, points out that it is necessary and worthwhile to develop a user-friendly statistical module in GIS directly, provides an example to illustrate how this can be implemented in ArcView using Avenue, and indicates how GIS and spatial statistical analysis can mutually benefit from such a module.*

INTRODUCTION

Geographic information systems (GIS), as the major handlers of spatial data, are efficient in the input, storage, manipulation and visual output of spatial databases. Some functionalities of spatial operations, such as map overlay, minimum cost path analysis and kriging, are also included in selected advanced versions of GIS, such as ARC/INFO 7.0.4 or IDRISI. But GIS users will find that almost all the current commercial GIS packages are extremely limited in standard statistical, let alone spatial statistical, capabilities; and, spatial statistics is more complex in theory and intensive in numerical computation. The dominant GIS software, ARC/INFO (ESRI, 1994a) and ArcView (ESRI, 1994b), for example, can only give some of the most basic summary statistics about data, including the sum, count, mean, minimum, maximum, range, standard deviation and variance. All other standard statistical techniques, such as OLS regression (the workhorse of conventional statistics) and ANOVA, are not included in ARC/INFO and ArcView, though they are very necessary and helpful for a user who wishes to examine relationships between different variables, and to make statistical decisions with geo-referenced data.

Presumably one of the reasons why GIS vendors have not yet developed their own statistical functionalities is the existence of a large number of reputable, comprehensive commercial statistical packages. Also it is always regarded as nonsensical to reinvent the wheel. So whenever spatial statistical analysis becomes necessary, a user usually has to go

[1]E-mail: zzhang01@maxwell.syr.edu

through the following procedures: export the spatial database from GIS formats to other formats that could be accepted by statistical packages, perform statistical analyses and modeling in these packages, obtain statistical results, and then send them back into a GIS if needed. This implementation process, using the strategy of transferring data back and forth between GIS and statistical packages, obviously has several drawbacks. First, it may be very complicated and inefficient if the analyzed GIS datasets are huge in size and complex in structure. Unfortunately this is usually the case for spatial data, which tend to cover a large study area and have their own uniquely defined structural relationships. Second, the topological complexity of spatial data makes this implementation process almost unable "to preserve spatial structure in such a way that output from standard statistical software packages can be imported back into a GIS" (Griffith, 1993b, p. 108). Finally, and perhaps most importantly, this implementation process is not friendly and transparent to the user. The user would undoubtedly encounter a lot of difficulties if he/she is not familiar with a specific procedure or detail of this process; for instance, how to write programs in statistical packages such as SAS, or how to extract and export data using formats acceptable to SAS. The difficulties of the above implementation process, the lack of spatial statistical functions in GIS, plus the abstract and complex meanings of spatial statistics, prevent many GIS users from exploring data more deeply and gaining more insight into them. Performing statistical analysis, especially spatial statistical analysis on geographic data, therefore, is often seen only as the job of a specialist. This, in our opinion, has greatly hindered the adoption of spatial statistical analysis. In the foreseeable future, as the GIS community grows larger and GIS users become more skilled, the need to perform spatial statistical analysis on GIS data will inevitably become greater. As Openshaw (1991, p. 6) points out, "many users are currently in their data base creation phase but they will soon need the access to relevant spatial analytical methods." Therefore, it is critical to integrate statistical functions into GIS and provide users with transparent and easy-to-use interfaces. This is of special importance to the field of urban and regional analysis (social, economic, or environmental) since much of their analyses are conducted on data aggregated for different geographical areas or zones, such as census tracts, counties, council districts or states. Here we present an example of this endeavor, showing one possible way of achieving this job in ArcView using Avenue, an object-oriented macro programming language.

LINKING SPATIAL STATISTICAL ANALYSIS WITH GIS: AN OVERVIEW

After several world-wide efforts in emphasizing the importance of enhancing the spatial analytical abilities of GIS, especially after the U.K. Economic and Social Research Council (ESRC) sponsored workshop held at the University of Sheffield in 1991 (see Haining & Wise, 1991; Goodchild et al., 1992), it is generally agreed that incorporating at least some statistical analysis into GIS is necessary. But there are still a lot of discussions and disagreements revolving around the specific issues of what functionalities should be included, and how these functionalities should be made accessible.

What Functions Should Be Added to a GIS Toolbox?

Since it would be too costly and generally regarded as foolish to replicate a SAS, MINITAB or SPSS within a GIS software package, the first two questions we might ask

here are: "What necessary functions need to be added to GIS to complement its lack of statistical analysis capabilities?" and, "Are they of value and interest to both the GIS users and vendor community?" To answer these questions, it may be of help to distinguish between two of the most closely related notions in this area: spatial analysis and spatial statistics. Also, recognizing the current interface between ARC/INFO and statistics is enlightening.

The notion of "spatial analysis", or "spatial data analysis" (Bailey & Gatrell, 1995) can include any operations performed on georeferenced data (both locational and attribute). According to Goodchild (1991, p. 4), "spatial analysis is a large but unorganized and uncodified mass of techniques, with no formal structure." Openshaw (1991, p. 7) criticizes what usually is referred to as spatial analysis as being mainly map manipulation, and recommends eight spatial analysis techniques that might be regarded as "GIS appropriate generic." Bailey (1994, p. 17) doubts that what Openshaw has suggested "are not without their own set of problems." His table of "potentially useful statistical spatial analysis techniques" include more than 10 methods dealing with locational, attribute, and interaction data. In fact, due to the "unbounded nature of spatial analysis" (Goodchild, 1991, p. 5), true spatial analysis in GIS is a much longer-term goal, which in many respects may never be realized (Longley & Batty, 1996). From our point of view, though, there is a lack of consensus about what kinds of spatial analysis techniques should be linked with or integrated into GIS, regardless of current commercial GIS vendors' interests in complementing existing spatial analysis capabilities. For example, ARC/INFO 7.0.4 already has included kriging and trend surface analysis in its TIN module. ArcView 3.0 is also able to perform network modeling or raster GRID modeling using CAD files. To fix its statistical insufficiency, ArcView includes a simple "Bivariate Regression" script in its scripts library. But this is apparently not enough for statistical analysis. *So our argument here is: it is the statistical aspect of spatial analysis, rather than the classical spatial analysis, that is still much more ignored by both the GIS users and developers.* We also presently think that there is a serious need to urge both GIS users and developers to pay more attention to statistical analysis, especially spatial statistical analysis.

Spatial statistics is designed to explicitly recognize vocational information contained in geo-referenced and spatial data, i.e., the spatial dependency and correlation between the sample data values. It is mainly composed of two sets of specialized statistical tools: one involves the measuring and testing of spatial autocorrelation (see Griffith, 1987), and the other concerns how to rewrite the classical regression model so that spatial autocorrelation can be taken into account. Therefore, spatial statistics is definitely necessary if a GIS user wants to investigate the geographic distribution of some data, or relationships between geo-referenced variables. Unfortunately, "although the literature is replete with documentation pointing out serious inferential consequences attributable to a disregarding of spatial dependence, overlooking or ignoring this latent dependence more often than not is what is done" (Griffith, 1993b, p. 105). Designing a user-friendly and transparent spatial statistical module in GIS that is really easy for GIS users to use may be the key to changing this troublesome circumstance. According to Griffith (1993b), the spatial statistics toolbox in GIS should include (1) a standard OLS multiple regression procedure (without all of the frills that usually accompany such a procedure nowadays), (2) a test for spatial autocorrelation in regression residuals (e.g., the Moran Coefficient), and (3) a nonlinear regression procedure designed specifically for estimating spatial autoregression parameters. These three aspects, by using regression as the workhorse technique, could cover

all of the basic spatial statistics, supplying the minimum set of tools for a user to investigate his/her GIS database from a statistical point of view.

How Should Statistical Analysis Be Linked with GIS?

Before we delve into the detail of discussing the linkage between GIS and spatial statistical analysis, we may notice that there already exist some good stand-alone spatial statistical software packages, such as the geographical analysis machine (GAM) developed by Openshaw, Charlton, and Wymer (1987), and SpaceStat developed by Anselin (1992). In general, developing stand-alone spatial analysis software packages is "not a good strategy because any spatial analysis package will need facilities for data input, data editing, database management and data display. These are the very areas where GIS is strong, and it seems foolish not to take advantage of this" (Goodchild, Haining, & Wise, 1992). Some raster-based GIS packages such as IDRISI and INFOMAP, and an interactive statistical tool called "Spider" (Haslett, Wills, & Unwin, 1990) have some built-in statistical functions. But the proprietary GIS and mapping packages, namely, ARC/INFO, ArcView and MapInfo, do not have such built-in functions yet. MapInfo is of particular interest at present, now that it is linked with SPSS. Of note also is that ARC/INFO is linked to S+, and SAS–GIS is available now. Here we focus our attention on discussing the possible ways to link statistical analysis with ESRI's GIS software packages.

According to Goodchild (1991) there are generally three strategies of linking spatial analysis and GIS: loose coupling, close coupling, and full integration. Loose coupling and close coupling methods are recognized as more desirable by both Goodchild (1991) and Rowlingson, Flowerdew, and Gatrell (1991), given the nature of the GIS software industry and the unbounded nature of spatial analysis. Anselin, Dodson, and Hudak (1993) further suggest classifying the coupling strategy into three categories (one-directional, two-directional, and dynamic integration) based on the type and number of information flows between GIS and spatial analysis modules. As far as the linkage of spatial statistics (rather than spatial analysis) and GIS is concerned, our opinion, again, is that though it seems inappropriate to provide all possible statistical spatial analysis methods in GIS, it is feasible and meaningful to integrate the minimum set of spatial statistical tools, as defined above, with GIS. This strategy targets "casual" users of GIS and spatial statistical analysis, who, in contrast to "heavy" users, do not have access to the current packages like S+ GIS Link or SAGE (Haining, Ma, & Wise, 1996). This idea is further explained in the following critique of both the "coupling" and the "integration" strategy.

Creating Interfaces Between GIS and Statistical Software

This is possibly the first way one could think of considering the present difference existing between the nature of GIS and standard statistical analysis packages, and the impracticality of duplicating one within the other. It also corresponds to both the loose and close coupling methods (sometimes it is hard to distinguish loose and close couplings). Kehris (1991), by interfacing ARC/INFO with GLIM, devised a set of FORTRAN subroutines that can calculate the Geary and Moran statistics for a given ARC/INFO coverage and an associated variable. SpaceStat, though largely a stand-alone software package, also provides an interface for the importation of files from GIS packages, such as ARC/INFO, IDRISI and OSU-MAP-for-the-PC. The linkage between SpaceStat and

ArcView also is explored by Anselin and Bao (1996). Three of the most significant advances in this area can be seen in: (1) ESRI's recently released ARC/INFO Version 7.0.4 for UNIX systems, which provides a transparent interface that allows ARC/INFO users to achieve S+ analysis and results; (2) the link between ArcView and XGobi (Majure, Symanzik, Cook, et al., 1996), a dynamic graphics program that allows multivariate data to be explored through the manipulation of scatterplots; and (3) SAGE (spatial analysis in a GIS environment), a software package for the interactive analysis of area-based data in a client-server architecture, developed in the Department of Geography at the University of Sheffield. SAGE has two important advantages: first, it supports a set of functions that performs classical and spatial statistics, with statistical plots displayed in a separate graphical window; second, it has a linked window capability, which means that cases highlighted in one window are also highlighted in others.

Though these advances may partly fix the statistical analysis deficiency of a GIS, considerable doubt remains about how much the "coupling" strategy will bring to users, and the extent to which statistical capabilities should be developed in GIS. First, many statistical packages may not be "spatial statistics specific," void of statistical analysis implementation specific to spatial data. Adding spatial statistical functions or modules to them (for example, S+ SpatialStats) requires so significant an extra investment and effort that a lot of GIS users cannot afford to buy them. Second, the internal executing process of this interface requires sending ARC/INFO data back and forth between statistical packages and GIS. This process will more than likely affect the implementation speed of statistical analysis. Finally, and most importantly, S+, XGobi or the SSA module (the spatial statistical analysis module of SAGE) are only a few of the available statistical packages. To the GIS users who do not have these packages in hand, or who use other statistical software such as SAS or SPSS instead of S-PLUS, this interface may be useless. We cannot ask all of the GIS users to buy S-PLUS, and it is unlikely in the short run that interfaces between GIS and all of the remaining statistical software packages will be created. Actually, the strategy of creating interfaces between GIS and statistical software often creates stricter system requirements for spatial statistical analysis since it requires strong facility and software tools for bridging differences between GIS and statistical packages. For example, to run SAGE, users need to have Sun workstations with Solaris 2.4, the Motif 1.2 shared library, the NAG FORTRAN library, and ARC/INFO 7.0.2 with Arctools 7.0.2.

Developing a Built-In Statistical Module Directly in a GIS

This corresponds to the "integration" strategy. A built-in statistical module in GIS can be written either in low-level standard languages such as FORTRAN or C/C++ (largely a job for GIS developers, with its progress depending to a large degree upon their interests and efforts), the macro languages of GIS, or both. In this article we take the approach of using the macro languages of GIS. Compared with the other aforementioned methods, this approach has two major advantages. First, the macro languages of GIS software, such as SML/AML of ARC/INFO, Avenue (ESRI, 1994c) of ArcView, or MapBasic of MapInfo, are designed specifically to access GIS data. Though they are not as complete as low-level languages such as C/C++ or FORTRAN, they allow a user to automate his/her work or analyze GIS data in a much easier way. For example, creating a scatterplot in ArcView only needs a few statements using Avenue language code, but it might be quite complicated

to do this very same job in low-level languages. The other striking advantage is that macro languages are usually much easier to learn and use. Since, presently, GIS is employed mainly for data manipulation and map display, and there is no sign that GIS developers will add exploratory spatial data analysis modules to current commercial GIS in the near future, using powerful and easy-to-use macro languages to customize GIS becomes a natural and practical choice for those users who want to do routine spatial statistical analysis in GIS. Ding and Fotheringham (1992) developed a spatial analysis module (SAM) in ARC/INFO. It consists of several C programs measuring the spatial autocorrelation and association of ARC/INFO data. These programs are accessible through AML commands. An obvious extension to SAM is incorporation of regression analysis. But ARC/INFO itself does not provide powerful tools for visualizing statistical analysis results like the Moran scatterplot (Anselin, 1993) or other statistical plots, and it does not have spreadsheet-like tabular operations either. Another good example is the population density modeling package in ARC/INFO developed by Batty and Xie (1994). This package is written in AML and FORTRAN and has improved the graphical display ability in ARC/INFO, including the statistical plots of urban population counts and population density at different distances from the city center. But the emphasis of this package is more on display rather than statistical analysis. In this paper we explore another new possibility that has not yet been discussed: building a spatial statistical module directly in ArcView using its macro language, Avenue.

INTEGRATING SPATIAL STATISTICAL ANALYSIS IN ARCVIEW

Why ArcView?

Friendly Graphical User Interfaces for Statistical Purposes

The Customize dialog box in ArcView allows a user to design his/her own graphical environment by modifying the existing ArcView controls (menus, menu items, buttons and tools), or linking his/her own scripts with new controls created in any of the windows of Project, View, Table, Chart, Layout or Script. Therefore, users can readily create a friendly graphical user interface (GUI) for statistical analysis purposes by themselves. Besides the powerful customizing tool, a wide selection of message box windows and dialog box windows also can be easily applied using Avenue to help users view, choose, input or report important information and results, or to report error messages. All of these window-based operations make it possible for users to execute abstract statistical techniques simply by pointing and clicking, maximizing the user-friendliness of a statistical module in GIS.

Statistical Graphics

Many statistical concepts cannot be clearly illustrated without graphics. Before ArcView, ARC/INFO users rarely had the chance to use statistical graphics to visually examine spatial data. Rowlingson et al. (1991) developed a module called "Arcgraph" using FORTRAN, and requiring a graphics package called UNIRAS, so that ARC/INFO users may produce some simple statistical graphics in ARC/INFO. With the Chart module in ArcView, users now can directly create area charts, bar and column charts, line charts, pie charts or scatterplots based on an entire table or a selected subset of it. Moreover,

Avenue scripts can be written to develop other statistical graphics, such as histograms (refer to ArcView's script library). In particular, when performing a test of spatial autocorrelation, users may want to create a Moran scatterplot to identify spatial outliers, influential points or spatial regimes.

Multi-Window Explorations of Statistical Data and Output Results

With maps, tables and charts displayed simultaneously in several windows, spatial data can be explored comprehensively in ArcView. Since View, Table and Chart are dynamically linked together, a change or a query performed in one of them also will be reflected in the others. For example, in a multi-window format, the results of running a Moran test on a variable can be displayed separately in a map window, a table window showing the local spatial statistics of each location, and a chart window with the Moran scatterplot. Possible outliers, influential points or spatial regimes may be easily identified from the Moran scatterplot; their details and locations are further highlighted by the map and table.

Flexibility of the Module Written in Avenue

ArcView represents the future trend of Desktop GIS for Windows, and object-oriented Avenue perhaps is the most complete macro language that desktop GIS users can utilize to create user-friendly and transparent graphic interfaces for spatial statistical analysis. An easy-to-use spatial statistical module written in Avenue, as illustrated in the following text, will not only be able to perform some of the most important spatial statistical functions, like those provided by Ding and Fortheringam's SAM module, but also have a similar multi-window and linked graphics capabilities as SAGE, since ArcView is window-based. However, the most significant advantage of this module is that it can be executed on any version of ArcView higher than 2.0,[2] and on both the workstation and PC, since ArcView packages on these two platforms use the same Avenue language. This flexibility will definitely facilitate GIS users in performing routine spatial statistical analysis of data, and thus greatly encourage the dissemination of spatial statistics to GIS users.

Framework for the Module

The module framework described in Figure 1 emphasizes again the three aforementioned functionalities of a spatial statistical toolbox. Spatial data sources for this module can include all the data that can be accepted into ArcView, such as ARC/INFO coverages, ArcView shapefiles, CAD data sources, or a tabular data source containing events such as a file of customers that ArcView can geocode. Once these data are read into ArcView, a typical spatial data manipulation and analysis process can include the following.

(1) Data preparation. This procedure generates spatial relationship configuration files, such as a spatial weights matrix or spatial neighbors list. They are the prerequisite for doing almost any spatial statistical analysis. The attribute data for the variables that are going to be analyzed also need to be extracted from a dataset.

[2]The module developed in this paper has been tested in ArcView version 2.0, 2.1 and 3.0a. Note that in ArcView 3.0a there is an undocumented chartable records limit of 50, which is significantly smaller than previous versions. To create a scatterplot with more than 50 data points, use Avenue request "aChartDisplay.SetMaxDataPoints".

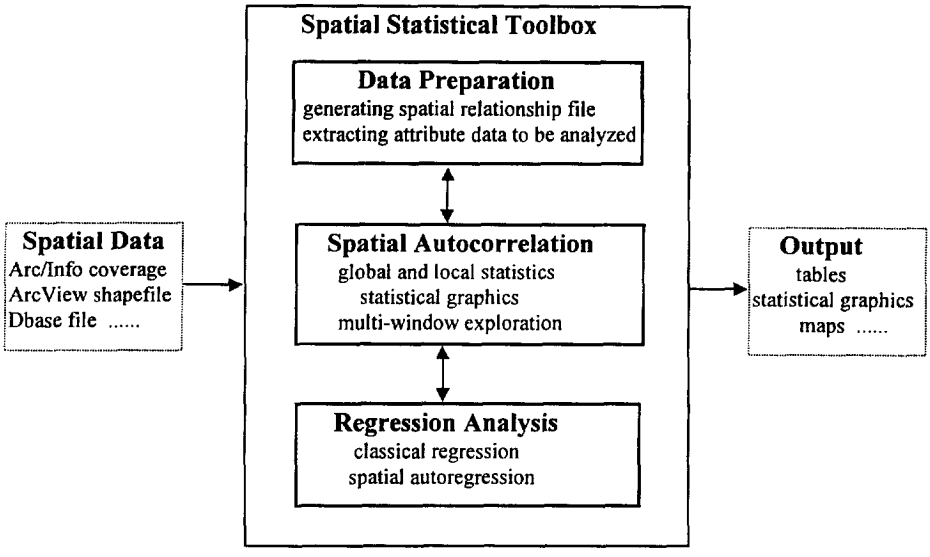


FIGURE 1. Framework for the spatial statistical module in ArcView.

(2) Spatial autocorrelation test. It performs a comprehensive spatial dependence test (both global and local) on the data made ready from the preceding procedure or the residual of a classical regression model produced by the ensuing procedure. By producing necessary statistical graphics and utilizing the linkage between maps, tables and graphics, spatial outliers, influential points or clusters latent in a dataset can be easily identified.

(3) Regression analysis. If spatial autocorrelation is found not to be significant with the preceding procedure, then a user may be able to perform classical regression directly to investigate the relationship between different variables. Otherwise, a spatial autoregression needs to be considered in order to properly account for latent spatial autocorrelation.

IMPLEMENTING STATISTICAL ANALYSIS IN ARCVIEW USING AVENUE

Expressing and Calculating Matrices

In classical regression, the vector $\mathbf{b}$ of OLS coefficients can be calculated with:

$$\mathbf{b} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \tag{1}$$

where $\mathbf{X}$ is an $n \times k$ matrix of the values of the predictor variables, and $\mathbf{Y}$ is an $n \times 1$ vector of the values of the dependent variable. This formula reveals how a matrix should be devised in order to store the data values of variables, and three types of matrix calculation: transpose, multiplication, and inverse. Since matrix transpose and multiplication are relatively straightforward operations, here we focus on how to use Avenue to accomplish matrix inversion.

The Avenue language does not have specific objects that directly store a two-dimensional array of values, but by using the "List" class, a two-dimensional matrix can be

expressed as a one-dimensional array. A List is an ordered collection of heterogeneous objects, and the index numbers of its elements range from 0 (representing the first element) to (list.count − 1) (representing the last element). For example, if we express matrix **X** ($n \times k$) in Equation 1 as a one-dimensional "list" object, the structure and index number of each element of matrix **X** will be:

X1	X2	.....	Xk
0	n		$k-1$*n
1	$n+1$		$k-1$*$n+1$
2	$n+2$		$k-1$*$n+2$
...	...		...
$n-1$	2*$n-1$		k*$n-1$

Alternatively, matrix **X** can be expressed as a nested "list" object. As Figure 2 shows, the "list" object in the first layer has k elements (corresponding to k predictor variables). Each element of this object, instead of storing a single observation value, points to a nested "list" object with n elements, which stores the n observation values of a variable. Once the mechanism of expressing a matrix is known, we need to outline how to use Avenue to accomplish matrix inversion.

There are several linear algebra methods[3] to calculate the inverse of a matrix. Due to the limits of Avenue, the easiest way appears to be the identity matrix method (Forsythe, 1967). Suppose **A** is a $k \times k$ matrix of full rank, and **E** is a $k \times k$ identity matrix. If we juxtapose **A** and **E** together horizontally, and apply only elementary transformations to both **A** and **E** simultaneously, then whenever **A** is transformed into an identity matrix, the original **E** matrix will become the inverse of matrix **A**. This process may be written as follows:

$$\mathbf{AE} <> \mathbf{EA}^{-1}$$

If we use a list to hold the **A** matrix, then **A** can be expressed as (the subscript is the index number of each element in the list):

$$\mathbf{A} = \begin{pmatrix} a_0 & a_1 & ..... & a_{k-1} \\ a_k & a_{k+1} & ..... & a_{2k-1} \\ ... & ... & ... & ... \\ a_{(k-1)k} & a_{(k-1)k+1} & ... & a_{kk-1} \end{pmatrix}$$

The above transformation process can be summarized as the following sequence of calculation steps (see Appendix A for a listing of source code).

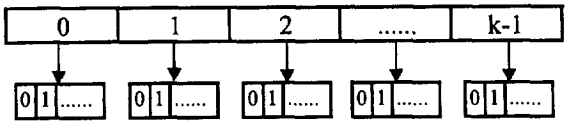


FIGURE 2. Expressing a matrix using a nested "list" object.

[3]Basic or direct matrix algebra methods are employed here. Refer to Searle (1982) for more sophisticated algorithms.

(1) Set the first element of the diagonal vector, i.e., a_0, to 1 by dividing each element of the first row of **AE** by a_0.
(2) Multiply the first row by $-a_k$, and add the results to the second row. This will transform a_k to zero. Similarly, transform all the elements in the same column as and below a_0, i.e., a_k, a_{2k}, ... and $a_{(k-1)k}$, to zero.
(3) Similar to steps (i) and (ii), each element of the diagonal vector, from a_{k+1} to a_{kk-1} respectively, can be set to one, and all the other elements in the same column as a given diagonal element can be transformed to zero. This procedure will make all the elements to the left of the diagonal zero.
(4) Multiply the last row by $-a_{(k-1)k-1}$, and add the results to the next-to-last row. These computations transform $a_{(k-1)k-1}$ to zero. Similarly, transform all the elements in the same column as and above a_{kk-1}, i.e., $a_{(k-1)k-1}$, $a_{(k-2)k-1}$, ... and a_{k-1}, to zero. Finally, transform all the other elements to the right of the diagonal to zero. This entire procedure will transform matrix **A** into an identity matrix.

Test of Spatial Autocorrelation

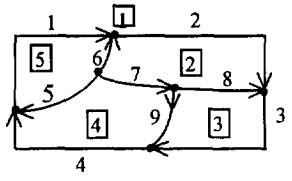
After figuring out how to express and calculate matrices in Avenue, performing classical regression analysis in Avenue becomes straightforward (see Appendix A for a listing of source code). The next question addressed here asks how to implement a test of spatial autocorrelation. Spatial autocorrelation means the self-correlation or spatial dependence among observations of a geo-referenced attribute. It is one of the most conspicuous features of spatial data. There are two different scales for spatial dependence: global indicators (summarizing the autocorrelation in data values in many different locations), and local indicators (identifying the association between a single location and its neighbors). The computation of both of these categories of indicators involves the configuration of areal units depicted by a spatial connectivity matrix **C**, whose entries are one if the corresponding row and column observations are juxtaposed, and zero otherwise.

Constructing a Connectivity Matrix

To construct a connectivity matrix manually is cumbersome, time consuming, and tedious, especially if the spatial dataset is large. This difficulty, however, can be overcome by making full use of the topology relationships defined in GIS data. For an ArcView shapefile, if two polygons (a kind of "shapes" in ArcView) have zero distance in space, then they are spatial neighbors. For an ARC/INFO coverage (one of the most often used GIS data formats), a connectivity matrix can be generated automatically by accessing the polygon-arc topology stored in the Arc Attribute Table (Kehris, 1991; Ding & Fotheringham, 1992). Figure 3 shows how the left–right polygon list in an Arc Attribute Table can be used when constructing the **C** matrix. Coverage named "Map1" has four polygons and nine arcs, with the topology relationships stored in the AAT file and the attribute of interest stored in the PAT file. The "Map1_" field of a PAT file stores the internal number of polygons, while the field of an AAT file stores the internal number of arcs. Each arc has its direction and polygon. By searching through the AAT file, record by record, if two polygons share a common boundary—for instance polygons 2 and 3 share arc 8—then unity is assigned to the corresponding element of the **C** matrix. This procedure yields a **C** matrix like that in Figure 3(d).

If the number of observations is large, say a thousand, the connectivity matrix will be not only large in size ($1{,}000 \times 1{,}000$) but also difficult to read and print. In most cases,

Coverage: Map1



2 Arc number

[2] Polygon number

(a)

Polygon Attribute Table(PAT)

Map1_	Poly_id	attribute_1
1	0	.....
2	1	.....
3	2	.....
4	3	.....
5	4	.....

(b)

Arc Attribute Table(AAT)

Map1_	Lpoly_	Rpoly_
1	1	5
2	1	2
3	1	3
4	1	4
5	4	5
6	5	2
7	2	4
8	2	3
9	3	4

(c)

$$C = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

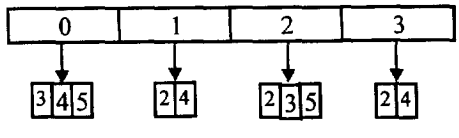
(d)

FIGURE 3. Components for constructing matrix C using ARC/INFO topology.

matrix **C** is a relatively less efficient way of storing the spatial relationship than a spatial neighbors list file. For example, the spatial connectivity relationships contained in Figure 3(d) can be condensed into a spatial neighbors list file as shown in Figure 4(a). The latter has the same row numbers as the connectivity matrix, but its column numbers will be substantially smaller. This efficiency will significantly increase as the number of spatial observations increases. The spatial neighbors list file in Figure 4(a) can also be expressed as a nested "list" object in Avenue as displayed in Figure 4(b). The index number of each element of the "list" object in the top layer plus 2 corresponds to the internal number of each polygon, and each element points to a nested "list" object storing the internal numbers of a polygon's neighbors. (See Appendix A for a listing of the Avenue code for generating a spatial neighbors list object.)

polygon	neighbors
2	3,4,5
3	2,4
4	2,3,5
5	2,4

(a)



(b)

FIGURE 4. Spatial neighbor list file and its Avenue expression.

Computing Global Statistics

There are two well-known global indicators that are used to measure spatial autocorrelation: the Moran coefficient (MC) and the Geary ratio (GR). Their formulae are, respectively,

$$\mathrm{MC} = n\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}(x_i - \bar{x})(x_j - \bar{x}) / [\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} \sum_{i=1}^{n}(x_i - \bar{x})^2] \tag{2}$$

$$\mathrm{GR} = [(n-1)\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}(x_i - x_j)^2] \quad / \quad [2(\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}) \sum_{j=1}^{n}(x_i - \bar{x})^2] \tag{3}$$

where c_{ij} are the elements of the connectivity matrix **C**, and $c_{ij} = 1$ if area units i and j are adjacent; otherwise $c_{ij} = 0$.

These two indices are inversely related, and one index can be expressed in terms of the other. $\mathrm{MC} = -1/(n-1)$ or $\mathrm{GR} = 1$ indicates a random map pattern; $\mathrm{MC} > -1/(n-1)$ or $0 < 1$ indicates that similar values tend to cluster on a map (positive spatial autocorrelation); $\mathrm{MC} < -1/(n-1)$ or $\mathrm{GR} > 1$, on the other hand, indicates that dissimilar values tend to cluster on a map (negative spatial autocorrelation).

Since the computations of MC and GR are similar, here we only discuss how to compute MC and its normalized Z-score with Avenue. Suppose that altogether there are n areal units in the sample data set; then in the case of a single variable, the Moran coefficient can be computed typically using double "For each ... in ... " loops (see Appendix A for a listing of source code). The mathematical expression of the variance of MC under a normality assumption is very lengthy but can be simplified to Equation 4 after using the notations of S_0 and S_1. S_0 and S_1, similarly, can be computed in Avenue using double "For Each ... in ... " loops (see Appendix A for a listing of source code).

$$\sigma^2_{\mathrm{MC}} = \frac{2S_0 n^2 - 4n(S_0 + S_1) + 3S_0^2}{S_0^2(n^2 - 1)} - \frac{1}{(n-1)^2} \tag{4}$$

where

$$S_0 = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} \quad S_1 = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}(\sum_{j=1}^{n} c_{ij} - 1)$$

If the Moran coefficient is used as a diagnostic tool for regression residuals (Cliff & Ord, 1973; Griffith, 1993a), the original formula for calculating its expected value (e.g., $-1/(n-1)$) should be replaced with:

$$\mathrm{Emc} = -n * \mathrm{tr}[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{C}\mathbf{X}] / [(n-p-1)\mathbf{1}^T\mathbf{C}\mathbf{1}] \tag{5}$$

where **X** is the matrix of independent variables (including a vector of ones), and p is the number of predictor variables. This calculation is straightforward following the methods of doing matrix operations in Avenue that have been illustrated above.

Local Statistics and the Visualization of Results

A global indicator summarizes the autocorrelation across a map using only a single value. It contributes little to finding the map patterns that may exist in different local map

areas, especially when the given GIS dataset is large. Getis and Ord (1992) suggest using Gi and Gi* statistics to aid in the identification of "hot spots," local spatial clustering and local instability. Anselin (1995) further summarizes the local statistics notion and defines a class of local indicators of spatial association (LISA), including Local Gamma, Local Moran and Local Geary. Local statistics, when combined with the existing abilities of GIS, greatly facilitate the visualization of spatial statistical results. Figure 5 shows an example of using the combination of Local Moran statistics and a Moran scatterplot (Anselin, 1995) to interactively explore spatial data in ArcView. Note that MC is a standardized slope of the MC scatterplot constructed with matrix **C**.

Local statistics for each observation can be calculated using Avenue (see Appendix A for a listing of the source code for calculating local Moran statistic) and the results can be stored and displayed in a "Table" window. Based on this table, (1) a Moran scatterplot can be drawn in a "chart" window, outliers, leverage points and four types of spatial clustering then can be found and identified both on the map and in the table; and (2) a subset of the table containing observations with significant local Moran values can be extracted by performing a query on the table, and two types of spatial clustering schemes can then be identified both on the map and in the table. A positive local Moran statistic indicates a local spatial clustering of similar values (either high or low), and a negative value a local clustering of dissimilar values.

AN ILLUSTRATION: SPATIAL PATTERNS OF POPULATION IN TENNESSEE STATE

Exploring the spatial patterns of population is one of the most frequently performed spatial analyses. Here we employ the module we describe above to analyze the 1990 county population density data for Tennessee State. A global test for spatial autocorrelation, which is implemented by simply clicking the button labeled "G" on the graphical user interface (Figure 7), yields a global Moran coefficient for the log-transformed population

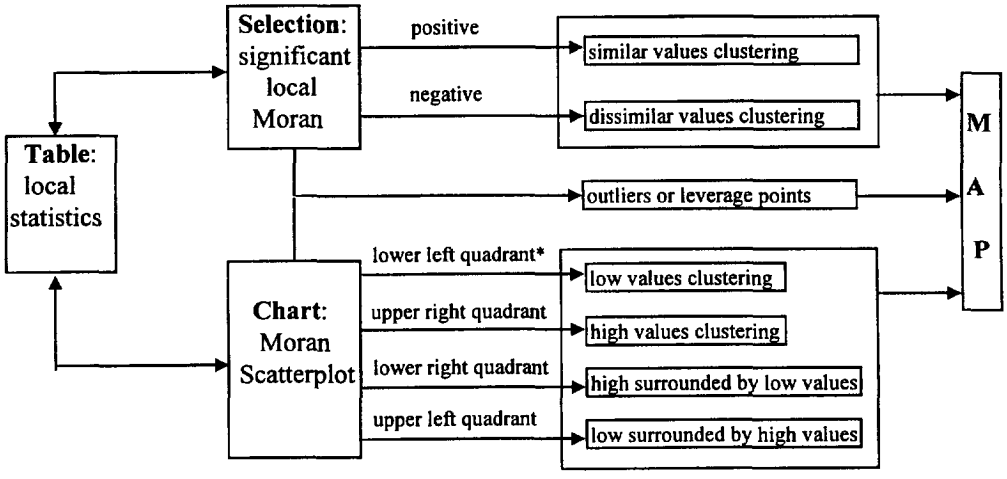


FIGURE 5. Combining the local Moran and the Moran scatterplot to explore the map pattern in ArcView. Notes: *When standardized values are used, the *X* axis and *Y* axis crossing the (0,0) point separate the space into four quadrants. The lower left quadrant indicates the spatial clustering of low values (i.e., less than the mean).

density variable (logpop) of 0.285, and its standardized Z value of 4.71. This result indicates the presence of positive overall spatial autocorrelation in variable "logpop". That is, counties with similar population density tend to cluster together on the map. To get a closer and more detailed look at the spatial pattern, a local test of spatial autocorrelation can be performed by clicking on the button labeled "L" on the graphical user interface (Figure 7). This will give us a table storing the local statistics for each county, including the values of the original variable (logpop), spatially lagged variable (c*logpop), local Moran (Mi), and its standardized Z (Zi) variable. Table 1 is a tabulation of the five counties with the most significant local Moran test statistics.

With an overall significance level of $\alpha = 0.10$, the individual significance level α_i given by a Bonferroni bounds procedure is $\alpha/m = 0.1/95$. or 0.0011.[4] Given this criterion, the local Moran values for three counties (Davidson, Knox, and Perry) are significant. The Moran scatterplot (Figure 6) shows that Davidson and Knox counties fall into the upper right quadrant. This indicates that there is local spatial clustering of high population around Davidson and Knox counties. This is not surprising since Nashville is located in Davidson county and Knoxville in Knox county. The local spatial statistics actually help us detect

Table 1. Individual Statistics for Five Counties with the Most Significant Local Moran Statistics

ID number	County name	Logpop	C*logpop	Local Moran	Z value	P value
31	Davidson	5.97	23.47	10.87	4.6236	0.0000
44	Knox	5.54	26.09	7.72	3.0683	0.0011
64	Perry	1.82	14.46	7.16	3.0526	0.0011
26	Washington	4.69	19.62	4.89	2.2770	0.0114
82	Wayne	1.99	12.65	4.50	2.0964	0.0180

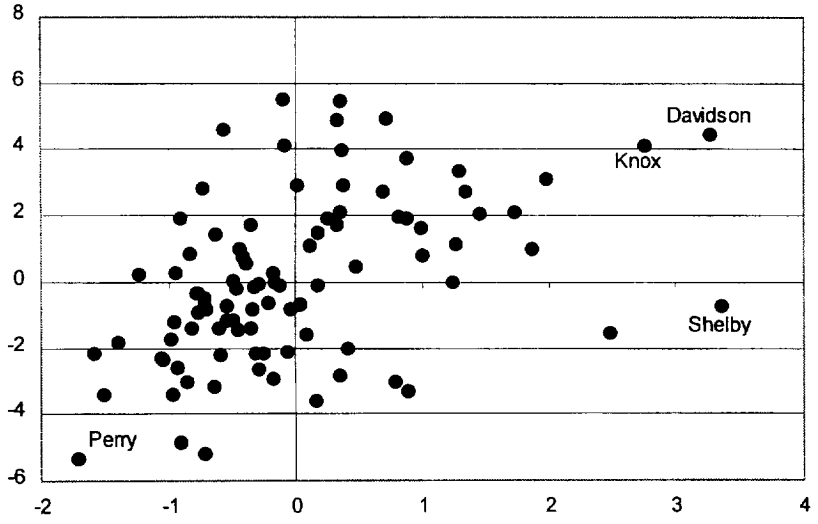


FIGURE 6. Moran scatterplot for the log-transformed population density.

[4]Note that the individual significance levels yielded by a Bonferroni adjustment may be too conservative in the assessment of local spatial statistics. A discussion about this can be found in Anselin (1995), and Ord and Getis (1995).

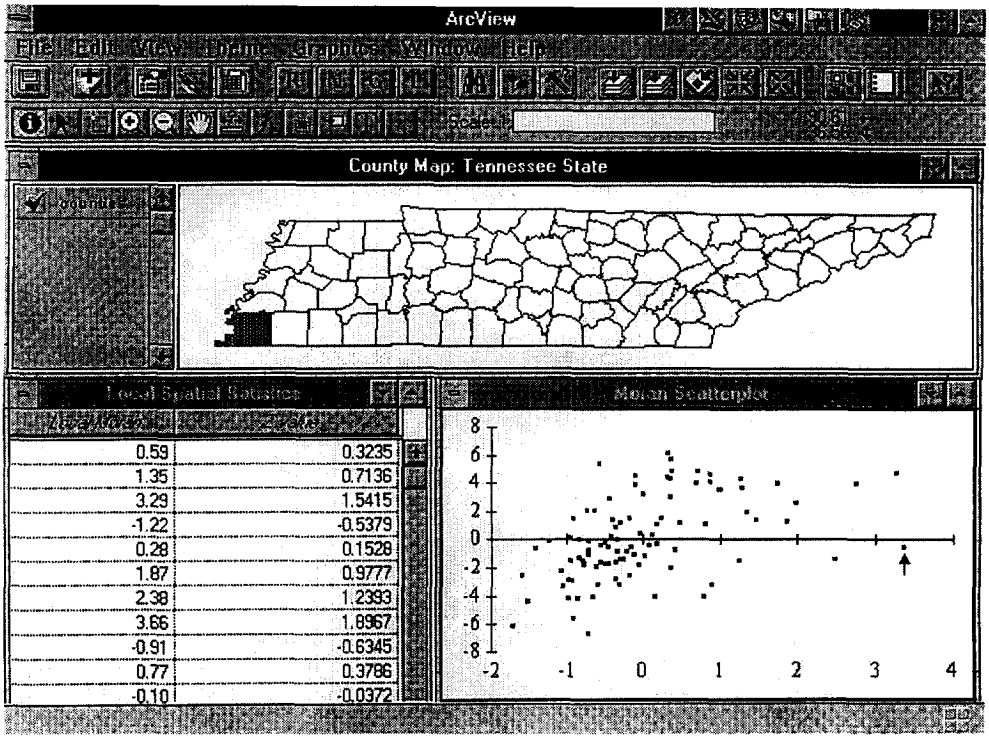


FIGURE 7. A multi-window exploration of the spatial statistical results.

two of the important metropolitan areas in Tennessee. In contrast, Perry county belongs to the lower left quadrant in the Moran scatterplot. Its local Moran value indicates there probably exists a spatial clustering of low population densities around Perry county, which is surrounded by the counties of Wayne, Decatur, Benton, Humphreys, Hickman and Lewis. Consulting the raw data confirms this finding. The population density of Perry county is the lowest of the 95 counties, Wayne the third lowest, Hickman the sixth, Humphreys the ninth, Decatur the 14th, Lewis the 15th, Benton the 22nd, and Hardin the 26th.

Since the map, the chart and the table are linked dynamically, possible outliers or influential points may be easily identified from the Moran scatterplot; their details and locations are simultaneously highlighted by the map and table. For example, as Figure 7 shows, a tabular window of the local statistics, a map window and a chart window of the Moran scatterplot can be displayed in the same ArcView window. As the Moran scatterplot indicates, the extreme point in the lower right-hand corner may be an outlier and deserves closer scrutiny. By using the identify tool to highlight it, one can immediately check out the detailed data for this observation in the table, which county this point represents (Shelby), and where the relative location of this county is (shaded part of the map) in the View window. Shelby county includes Memphis, the biggest metropolitan area in Tennessee. It is in a very unique location on the map, it shares its left boundary with Arkansas and its south boundary with the state of Mississippi. The lack of neighbor counties caused by the state boundary (edge effect), and the significant population density

contrast between Shelby and its neighboring counties (Fayette and Tipton) may be the reason why Shelby stands out as a spatial outlier.

CONCLUSIONS

The major aims of this paper were to explore the conceptual importance and technical possibilities of improving statistical analysis capabilities in GIS, and to illustrate advantages of fully integrating statistical analysis with GIS. ArcView together with Avenue may be a good platform for a user-friendly statistical analysis module, though ArcView and Avenue have their own limitations. To name just a few, Avenue can only use List objects to store matrices in one dimension, the implementation speed of Avenue may not be as fast as the programs written in low-level languages, such as C/C++ or FORTRAN, and the topology of an ArcView shapefile seems not to be as explicit as that of an ARC/INFO coverage. Additionally, there is a need to do more research on how to develop graphics in ArcView that are more suitable for statistical purposes, such as boxplots and stern-and-leaf plots, and how to enhance the dynamic linking between a map view, chart and table in ArcView. Currently the linkage between the View and Chart windows has to be launched from their intermediate Table windows.

The effort to develop a statistical module in GIS directly, however, should not be hampered by these minor limitations. Our example, by focusing on how to integrate standard OLS regression analysis and the test of spatial autocorrelation into ArcView, exemplifies how GIS and statistical analysis can benefit from each other. A statistical module in GIS can take full advantage of the topology relationships and the visualization tools provided by GIS. GIS, on the other hand, may use this statistical module to extend its analytical capabilities from simple descriptive statistics to more exploratory and inferential data analysis. This example implementation, however, is far removed from covering all the statistical methods that potentially can be integrated into GIS. We have not yet discussed how to implement other local spatial statistics, such as Gi and Gi*, which is based on the distance connectivity matrix, nor the spatial autoregressive models. In fact, though there is general consensus that GIS should increase its statistical analysis abilities and develop its own statistical module, which techniques should be included in this module is still open to debate. As Bailey (1994) points out, these techniques should be particularly amenable to visual display in the form of a map, or useful in conjunction with a map. Obviously, more interesting questions and challenges will emerge from the arena of integrating GIS with statistical analysis. The outcomes reported in this paper may establish a foundation for us to develop further spatial statistical functions in ArcView with Avenue.

REFERENCES

Anselin, L. (1992). *SpaceStat tutorial: A workbook for using spaceStat in the analysis of spatial data.* UC–Santa Barbara: Department of Geography.

Anselin, L. (1993). *The Moran scatterplot as a means to visualize instability in spatial autocorrelation.* Position paper prepared for the Workshop on Exploratory Spatial Data Analysis and GIS, NCGIA, Santa Barbara, CA.

Anselin, L., Dodson, R., & Hudak, S. (1993). Linking GIS and spatial data analysis in practice. *Geographical Systems, 1,* 3–23.

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

Anselin, L. & Bao, S. (1996). *Exploratory spatial data analysis linking SpaceStat and ArcView*. Working Paper. Morgantown, WV: Regional Research Institute.

Bailey, T. C. (1994). A review of statistical spatial analysis in geographical information systems. In S. Fotheringham & P. Rogerson (Eds.), *Spatial analysis and GIS* (pp. 13–44). Bristol, PA: Taylor & Francis.

Bailey, T. C. & Gatrell, A. C. (1995). *Interactive spatial data analysis*. Essex, UK: Longman Scientific and Technical.

Batty, M. & Xie, Y. (1994). Urban analysis in a GIS environment: population density modelling using ARC/INFO. In S. Fotheringham and P. Rogerson (Eds.), *Spatial analysis and GIS* (pp. 189–220). Bristol, PA: Taylor & Francis.

Cliff, A. D. & Ord, J. K. (1973). *Spatial autocorrelation*. Monographs in Spatial and Environmental Systems Analysis. New York, USA: Methuen, Inc.

Ding, Y. & Fotheringham, A. S. (1992). The integration of spatial analysis and GIS. *Computers, Environment and Urban Systems, 16*(1), 3–19.

Environmental Systems Research Institute, Inc. (1994a). *Understanding GIS: The ARC/INFO Method*. Redlands, CA: ESRI.

Environmental Systems Research Institute, Inc. (1994b). *Introducing ArcView*. Redlands, CA: ESRI.

Environmental Systems Research Institute, Inc. (1994c). *Introducing Avenue*. Redlands, CA: ESRI.

Forsythe, G. E. (1967). *Computer solution of linear algebraic systems*. Englewood Cliffs, NJ: Prentice-Hall.

Goodchild, M. (1991). The case for linking GIS and spatial analysis. In R. P. Haining & S. M. Wise (Eds.), *GIS and spatial data analysis: Report on the Sheffield workshop* (pp. 4–6). Regional Research Laboratory Initiative Discussion Paper Number 11.

Goodchild, M., Haining, R. P., & Wise, S. M. (1992). Integrating GIS and spatial data analysis: Problems and possibilities. *International Journal of Geographical Information Systems, 6*(5), 407–423.

Getis, A., & Ord, K. (1992). The analysis of spatial association by use of distance statistics. *Geographical Analysis, 24*, 189–206.

Griffith, D. A. (1987). *Spatial autocorrelation: A primer*. AAG Resource Publications in Geography.

Griffith, D. A. (1993a). *Spatial regression analysis on the PC: Spatial statistics using SAS*. AAG Publication.

Griffith, D. A. (1993b). Which spatial statistics techniques should be converted to GIS functions? In M. M. Fischer & P. Nijkamp (Eds.), *Geographic information systems, spatial modelling, and policy evaluation* (pp. 103–114). New York: Springer-Verlag.

Haining, R. P., & Wise, S. M. (Eds.) (1991). *GIS and spatial data analysis: Report on the Sheffield workshop*. Regional Research Laboratory Initiative Discussion Paper Number 11.

Haining, R. P., Ma, J., & Wise, S. M. (1996). Design of a software system for interactive spatial statistical analysis linked to a GIS. *Computational Statistics, 11*, 449–466.

Haslett, J., Wills, G., & Unwin, A. (1990). SPIDER — An interactive statistical tool for the analysis of spatially distributed data. *International Journal of Geographical Information Systems., 4*(3), 285–296.

Longley, P., & Batty, M. (1996). Analysis, modelling, forecasting, and GIS technology. In P. Longley & M. Batty (Eds.), *Spatial analysis: Modelling in a GIS environment* (pp. 1–15).

Kehris, E. (1991). *Spatial autocorrelation statistics in ARC/INFO*. Research Report No. 16. North West Regional Research Laboratory, Lancaster University.

Majure, J., Symanzik, J., Cook, D., et al. (1996). *ArcView2.1–XGobi Link Version 1.3*. Available: http://www.gis.iastate.edu/XGobi-AV2/XGobi-AV2.html

Openshaw, S., Charlton, M., & Wymer, C. (1987). A Mark 1 geographical analysis machine for the automated analysis of point data sets. *International Journal of Geographical Information Systems, 1*, 335–358.

Openshaw, S. (1991). What types of spatial analysis techniques are needed in GIS? In R. P. Haining & S. M. Wise (Eds.), *GIS and spatial data analysis: Report on the Sheffield workshop* (pp. 6–7). Regional Research Laboratory Initiative Discussion Paper Number 11.

Ord, J. K., & Getis, A. (1995). Local spatial autocorrelation statistics: Distributional issues and an application. *Geographical Analysis, 27*, 286–306.

Rowlingson, B. S., Flowerdew, R. & Gatrell, A. (1991). *Statistical spatial analysis in a geographical information systems framework* (Research Report No. 23). Lancaster University: North West Regional Research Laboratory.

Searle, S. R. (1982). *Matrix Algebra Useful for Statistics*. New York: John Wiley & Sons.

APPENDIX A: AVENUE SOURCE CODE

Multiple Regression: Parameter Estimates, Inferential Statistics, and Residual Plots

```
'GET THE THEME, CHECK FOR SELECTION, GET INDEPENDENT AND DEPENDENT
    'VARIABLE FIELDS

theView = av.GetActiveDoc
theFTheme = theView.GetActiveThemes.Get(0)
theFtab = theFtheme.GetFtab
flist = list.make
for each f in theFtab.GetFields
   if (f.isTypeNumber = true) then
      flist.add(f)
   end
end
if (flist.count < 2) then
   exit
end
if (theFtab.GetSelection.Count 0) then
   theFtab. GetSelection. SetAll
   theFtab. GetSelection.clear(0)
end
DV = Msgbox.Choice(flist,"Choose a field for your dependent variable (Y)", "Multivariate Regression")
If (DV = Nil) then
   exit
end
IVn = 1
IVlist = list.make
while(true)
   aField = Msgbox.Choice(flist,"Choose fields for your independent variables (X)" + NL + "(Cancel to
      stop)". "Multivariate Regression")
   if (aField = Nil) then
   break
end
   IVlist.add(aField)
   IVn = Ivn + 1
end
If (IVlist.get(0) = nil) then
   exit
end

'GET THE MATRIX FOR THE INDEPENDENT VARIABLE FIELDS

xlist = list.make
xsumlist = list.make
nn = 0
For each i in 1..(ivn - 1)
   xsum = 0
   nn = 0
   For each f in theFtab.GetSelection
      xret = theFtab.ReturnValueNumber(IVlist.get(i - 1),f)
      if(xret.Isnull = true) then
         continue
      end
```

```
      xlist.add(xret
      xsum = xsum + xret
      nn = nn + 1
   end
   xsumlist.add(xsum)
   XBar = xsum/nn
end

'Insert 1 vector to x matrix
For each r in 1.nn
   xlist.insert(1)
end

'GET THE MATRIX FOR THE DEPENDENT VARIABLE FIELDS

ylist = list.make
Ysumlist = list.make
ysum = 0
nn = 0
For each f in theFtab.GetSelection
   yret = theFtab.ReturnValueNumber(DV,f)

   'Check for null values
   if (yret.IsNull true) then
      Continue
   end
   ylist.add(yret)
   ysum = ysum + yret
   nn = nn + 1
end
YBar = ysum/nn

'GET THE PRODUCT OF X MATRIX AND ITS TRANSPOSE

n = ylist.count
txx = list.make
For each j in 1..IVn
   For each k in 1.IVn
      txxsum = 0
      tn1 = j - 1)*n
      tn2 = (k - 1)*n
      For each i in 1..n
         txxret = xlist.get(tn1)*xlist.get(tn2)
         txxsum = txxsum + txxret
         tn1 = tn1 + 1
         tn2 = tn2 + 1
      end
      txx.add(txxsum)
   end
end

'GET THE PRODUCT OF Y MATRIX AND THE TRANSPOSE OF X MATRIX

txy = list.make
For each k in 1..IVn
   txy,sum = 0
   tn1 = (k - 1)*n
   For each i in 1..n
      txyret = xlist.get(tn1) * ylist.get(i - 1)
      txysum = txysum + txyret
```

```
      tn1 = tn1 + 1
   end
   txy.add(txysum)
end

'GET THE INVERSE OF THE PRODUCT OF X AND X TRANSPOSE

   'DEFINE AN IDENTITY MATRIX
ident = list.make
For each i in 1..IVn
   For each j in 1..IVn
      if (i = j) then
         ident.add(1)
      else
         ident.add(0)
      end
   end
end

   'transform the lower left half of product matrix to zero
IVn1 = IVn - 1
For each p in 0..IVn1
   diag = p*IVn + p
   c = txx.get(diag)
   For each i in 0..IVn1 'set the diagonal to one
      ni = p*IVn + i
      oldvx = txx.get(ni)
      txx.set(ni,oldvx/c)

      oldvi = ident.get(ni)
      ident.set(ni, oldvi/c)
   end
   If (p = Ivn1) then
      break
   end
   For each j in (p + 1)..IVn1 'set the 1st element below the diag to zero
      nj = j*IVn + p
      c1 = 0 - txx.get(nj)
      For each k in 0.. (Ivn - 1)

         oldvx = txx.get(j*IVn + k)
         c2 = txx.get(p*IVn + k)* cl
         txx.set(j*IVn + k,oldvx + c2)
         oldvi = ident.get(j*IVn + k)
         ci2 = ident.get(p*IVn + k)* cl
         ident.set(j*IVn + k,oldvi + ci2)
      end
   end

end

   'transform the upper right half of product matrix to zero
p = Ivn - 1
while(p > 0)
   For each j in 0..(p - 1) by 1
      nj = j*IVn + p
      c1 = 0 - txx.get(nj)
```

```
      For each k in 0.. (IVn-1) by 1
         oldvx=txx.get(j*IVn+k)
         c2=txx.get(p*IVn+k) * cl
         txx.set(j*IVn+k, oldvx+c2)

         oldvi=ident.get(j*IVn+k)
         ci2=ident.get(p*IVn+k) * cl
         ident.set(j*IVn+k, oldvi+ci2)
      end
   end
   p=p-1
end

'GET THE b VALUES FOR THE REGRESSION EQUATION

b=list.make
For each i in 0..(IVn-1)
   sum=0
   For each k in 0.. (IVn-1)
      sum=sum+(ident.get(i*IVn+k) * txy.get(k))
   end
   b.add(sum)
end

'GET THE INFERENTIAL STATISTICS

xb=list.make
sse=0
sst=0
e=list.make
   'get the product of x and b, and residual
For each i in 0..(n-1)
   xbsum=0
   For each k in 0..(b.count-1)
      xbsum=xbsum+(xlist.get(k*n+i) b.get(k))
   end
   xb.add(xbsum)
   ehat=ylist.get(i)-xbsum
   sse=ehat*ehat+sse
   e.add(ehat)
end

   'get the MSE, R Square and adjusted R Square
For each i in 1..(ylist.count)
   sst=(ylist.get(i-1)-ybar)* (ylist.get(i-1)-ybar)+sst
end
ssr=sst-sse
RSQ=ssr/sst
MSE=sse/(n-IVn)
MST=sst/(n-1)
MSR=ssr/(IVn-1)
ADJR=1-(MSE/MST)
RMS=mse.sqrt

   'get the standard error and t value
selist=list.make
tlist=list.make
For each i in 0..(IVn-1)
   se=RMS * (ident.get(i*IVn+i).sqrt)
```

```
    selist.add(se)
    tlist.add(b.get(i)/se)
end

'MAKE A RESIDUAL PLOT

    'add residual into the feature table as a field
spVtab = vtab.makenew("residual,dbf".asfilename, DBASE)
f = field.make(dv.getname,dv.gettype, 12,0)
rf = field.make("residual",dv.gettype,12,0)
spvtab.addfields({frf})

For each rn in 1..ylist.count
    rec = spvtab.addrecord
    spvtab.setvalue(f, rec, ylist.get(rn - 1)
    spvtab.setvalue(rf, rec, e.get(rn - 1))
end

    'define a chart
sp = Chart.make(spvtab,{frf})
sp.GetChartDisplay.SetType(#CHARTDISPLAY_XYSCATTER)
sp.GetChartDisplay.SetStyle(#CHARTDISPLAY_VIEW_SIDEBYSIDE)
sp.GetChartDisplay.SetMark(#CHARTDISPLAY_MARK_DOT)
sp.GetChartDisplay. SetSeriesColor(0,color.getred)
sp.Getchartdisplay.setseriescolor(1,color - getred)

sp.GetTitle.Setname("Residual Plot for" + + dv.getname)
sp.GetXAxis, SetLabelVisible(true)
sp.GetYAxis. SetLabelVisible(false)
sp. GetChartLegend.Setvisible(False)

    'get the chart's window and open it
spwin = sp.GetWin
spwin.open
spwin.Activate
```

Generating a Polygon Neighbors List (PNL) Table

```
theTable = av.GetActiveDoc
theVtab = theTable.GetVtab
Lpoly = theVtab.FindField("Lpoly_")
If (lpoly = Nil) then
Msgbox.error("Cannot find Left Polygon Field","Creating Polygon Neighbor")
exit
end
Rpoly = theVtab.FindField("Rpoly_")
If (Rpoly = Nil) then
    Msgbox.error("Cannot find Right Polygon Field","Creating Polygon Neighbor")
    exit
end
pnl = list.make
For each rec in theVtab
    lpv = theVtab.retumvaluenumber(Lpoly,rec)
    rpv = theVtab.retumvaluenumber(Rpoly,rec)
    if((lpv = 1)or(rpv = 1))then
        continue
    end
```

```
    m = lpv max rpv
    if(m > (pnl.count + 1)) then
        diff = m - pnl. count - 1
        For each i in 1..diff
            pnl.add(list.make)
        end
    end
    pnl.get(lpv - 2).add(rpy)
    pnl.get(rpv - 2).add(ipv)
end

'create the pnl table
PNL_Vtab = vtab.makenew("pnl.dbf".asfilename, DBASE)
pf = field.make("poly#",lpoly.gettype,lpoly.getwidth,lpoly.getprecision)
nbr = field.make("neighbor list",#FIELD_CHAR,70,0)
PNL_vtab.addfields({pf, nbr}
For each i in 1..pnl.count
    pnl.get(i - 1).removeduplicates
    pnl.get(i - 1).sort(true)
    rec = PNL_vtab.addrecord
    nblist = ""
    size = pnl.get(i - 1).count
    For each j in 1..(size - 1)
        nblist = nblist + pnl.get(i - 1).get(j - 1).asstring + ","
    end
    nblist = nblist + pnl.get(i - 1).get(size - 1).asstring
    PNL_vtab.setvaluenumber(pf, rec, i + 1)
    PNL_vtab.setvaluestring(nbr,rec, nblist)
end
PNL_table = table.make(PNL_vtab
PNL_table.setname("Polygon neighbors list")
PNL_table.getwin.open
```

Calculating the Global Moran Coefficient

Suppose "ylist" is a list holding the observation values of a variable, ysum is the sum of observation values, and "pnl" is the polygon neighbors list.

```
'GET THE NUMBER OF OBSERVATIONS AND THE MEAN OF ATTRIBUTE VALUES
size = pnl.count
ybar = ysum/size
'GET THE STANDARD DEVIATION OF ATTRIBUTE VALUES
sum = 0
For each i in 0..(size - 1)
    sum = (ylist.get(i) - ybar)*(ylist.get(i) - ybar) + sum
end
sd = (sum/(size - 1)). sqrt

'CENTRALIZE AND STANDARDIZE VARIABLE VALUE
cenylist = list.make
sdylist = list.make
For each i in 0..(size - 1)
    oldv = ylist.get(i)
    newv = oldv - ybar
    cenylist.add(newv)
    sdylist.add(newv/sd)
end
```

```
'CALCULATE MORAN COEFFICIENT
S0 = 0
s1 = 0
Sumij = 0
Sumii = 0
For each i in 1..size
   pnl.get(i – 1).removeduplicates
   pnl.get(i – 1).sort(true)
   size2 = pnl.get(i – 1).count
   For each j in 1..size2
   k = pnl.get(i – 1).get(j – 1)
   Sumij = cenylist.get(i – 1)* cenylist.get(k – 2) + sumij
end
sumii = cenylist.get(i – 1) * cenylist – get(i – 1) + sumii
s0 = size2 + s0
s1 = size2 * (size2 – 1) + s1
end

MC = (size/s0) * (sumij/sumii)
EMC = 0 – (1/(size – 1))
VAR_MC = ((2*s0*size*size) – (4*size*(s0 + s1)) + (3*s0*s0))/(s0*s0*(size*size – 1)) – (EMC*EMC)
STD_MC = VAR_MC.sqrt
ZScore = (MC – EMC)/STD_MC

'REPORT THE RESULT
result = "Test Of Spatial Autocorrelation" + NL + "– – – – – – – – – – – – – – – – – – – – – – – –" + NL
result = result + " Moran Coefficient " + MC.asstring + NL
result = result + " Standard Deviation " + STD_MC.asstring + NL
result = result + " Mean " + EMC.asstring + NL
result = result + " Z-Score " + Zscore.asstring
msgbox.report(result,"Moran Coefficient")
theVtab.getselection.clearall
theFtab.getselection.clearall
```

Calculating the Local Moran Statistics and Generating Moran Scatterplot

Suppose "ylist" is a list holding the observation values of a variable, "cenylist" and "stdylist" are respectively the list holding the centralized values and the list holding the standardized values, and "pnl" is the polygon neighbors list.

```
'compute c*y (original scale) and local Moran
cylist = list.make
estdylist = list.make
Iilist = list.make
Zilist = list.make
y2bar = y2sum/size
y4bar = y4sum/size
b2 = y4bar/(y2bar*y2bar)

For each i in 1..size
   size2 = pnl.get(i – 1).count
   cysum = 0
   cenysum = 0
   stdysum = 0
   For each j in 1..size2
      k = pnl.get(i – 1),get(j – l)
      cysum = ylist.get(k – 2) + cysum
      cenysum = cenylist.get(k – 2) + cenysum
      stdysum = stdylist.get(k – 2) + stdysum
```

```
    end
    cylist.add(cysum)
    cstdylist.add(stdysum)
    Ii = cenylist.get(i-1)*cenysum
    Iilist.add(Ii)
    Ei = -size2/(size-1)
    vi = size2*(size-b2)1(size-1)+(size2*size2*(2*b2-size)/((size-1)*(size-2)))-(size2*size2/((size-1)
        *(size)))
    Zilist.add((Ii-Ei)/(vi.sqrt)
end

'add cy and Ii into the feature table as a field
spVtab = vtab.makenew("MC.dbf".asfilename, DBASE)

pf = polyfield.clone
f = vf.clone

cvf = field.make("Neighbor" + + vf.getname, vf.gettype, (vf.getwidth + 1), vf.getprecision)
f1 = field.make("Std." + + vf.getname, vf.gettype, vf.getwidth, (vf.getprecision + 2))
cvf1 = field.make("Neighbor" + + f1.getname, f1.gettype, (f1.getwidth + 1), f1.getprecision)
mif = field. make("Local Moran", vf.gettype, (vf.getwidth + 1), vf.getprecision)
zif = field.make("Z value", vf.gettype, vf.getwidth, (vf.getprecision + 2))
spvtab.addfields({pf, f cvf, fl, cvfl, mif zif})

For each rn in 1..ylist.count
    rec = spvtab.addrecord
    spvtab.setvalue(pf, rec, polylist.get(rn-1)
    spvtab.setvalue(f, rec, ylist.get rn-1))
    spvtab.setvalue(cvf, rec, cylist.get(rn-1))
    spvtab.setvalue(f1, rec, stdylist.get(rn-1)
    spvtab.setvalue(cvf1, rec, cstdylist.get(rn-1))
    spvtab.setvalue(mif, rec, Iilist.get(rn-1))
    spvtab.setvalue(zif, rec, Zilist.get(rn-1))
end

'test result
sptable = table.make(spvtab)
sptable.setname("Local Spatial Satistics")
sptable.getwin.open

'Link two vtabs together
spvtab.link(pf, theFtab, polyfield)

'create a Moran scatterplot
sp = Chart.make(spvtab,{f1,cvf1})
sp.GetChartDisplay.SetType(#CHARTDISPLAY_XYSCATTER)
sp.GetChartDisplay.SetStyle(#CHARTDISPLAY_VIEW_SIDEBYSIDE)
sp.GetChartDisplay.SetMark(#CHARTDISPLAY_MARK_DOT)
sp.GetChartDisplay.SetSeriesColor(0,color.getred)
sp.Getchartdisplay.setseriescolor(1,color.getred)

sp.GetTitle.Setname("Scatter Plot for" + fl.getname)
sp.GetXAxis.SetLabelVisible(false)
sp.GetYAxis.SetLabelVisible(false)
sp.GetChartLegend.Setvisible(False)

spwin = sp.GetWin
spwin.open
spwin. Activate
```