Spatial Data Analysis in the Dynamically Linked ArcView/XGobi/XploRe Environment

J. Symanzik

Department of Statistics
Iowa State University
Ames, IA 50011

symanzik@iastate.edu

T. Kötter

Humboldt–Universität zu Berlin
Institut für Statistik und Ökonometrie
Spandauer Str. 1
10178 Berlin, Germany

thomas@wiwi.hu-berlin.de

S. Schmelzer

Humboldt–Universität zu Berlin
Institut für Statistik und Ökonometrie
Spandauer Str. 1
10178 Berlin, Germany

swetlana@wiwi.hu-berlin.de

S. Klinke

Humboldt–Universität zu Berlin
Institut für Statistik und Ökonometrie
Spandauer Str. 1
10178 Berlin, Germany

sigbert@wiwi.hu-berlin.de

D. Cook

Department of Statistics
Iowa State University
Ames, IA 50011

dicook@iastate.edu

D. F. Swayne

Statistics Research Department
AT&T Labs
Florham Park, NJ 07932

dfs@research.att.com

Abstract

In order to fully analyze spatial data, from satellite images to precision agricultural data, the user needs to do these three things: (1) display spatial locations and concomitant geographic variables on maps, (2) visualize these concomitant variables using more general interactive multivariate visualization methods, and (3) execute statistical analyses. At present, in order to accomplish those goals, three different kinds of software must be used: a Geographic Information System (GIS), a Dynamic Statistical Graphics (DSG) program, and a Statistical Computing Environment (SCE).

We have developed a method to dynamically link three pieces of software, one in each category, so that they can exchange commands and data. The different data analytic and graphical methods of these packages complement each other. This linking increases ease and efficiency at all stages of spatial data analysis.

The three programs are the GIS ArcView 3.0$^{\mathrm{TM}}$, the DSG program XGobi, and the SCE XploRe 4. The ArcView/XGobi/XploRe environment will allow us to directly issue and use XploRe commands from within any of the other two packages and it will dynamically link different types of views uniquely available in each program.

Two of the links have been successfully implemented, and the other should be completed soon.

1 Introduction

The following citation from Haining, Ma, and Wise (1996) provides a clear statement of the goal of the work described in this paper: *"In order to carry out a program of SSA [Spatial Statistical Analysis], the user needs to access software that can store and manipulate spatially referenced information, can execute appropriate statistical analysis and finally allow good interactive visualization of the raw data or analysis outcomes. In each case there already exist packages which provide these facilities — in the order listed above GIS, statistical packages (although the large commercial statistical packages are not designed to facilitate SSA) and visualization packages. However, no current package provides all three."*

We describe a dynamic link among three software packages that goes a long way toward providing an integrated environment for spatial statistical analysis. The three packages are the Geographic Information System (GIS) ArcView 3.0, the interactive Dynamic Statistical Graphics (DSG) program XGobi (Swayne, Cook, and Buja, 1991b), and the Statistical Computing Environ-

ArcView 3.0 is a trademark of Environmental Systems Research Institute, Inc.

ment (SCE) XploRe 4 (Härdle, Klinke, and Turlach, 1995). The main focus of this paper is on spatial data analysis in the ArcView/XGobi/XploRe environment but other aspects will be discussed as well.

GISs, in particular ArcView, are widely used for the management of multivariate spatial data bases. Unfortunately, they have very limited capabilities for statistical analysis and for interactive exploratory data analysis. If we need to conduct statistical analyses, use sophisticated statistical graphics, study longitudinal data sets, and use 3D geographical data, we must use software packages in combination. Linking these packages so that they can communicate dynamically makes their combined use much easier and more efficient.

XGobi (Swayne et al., 1991b) is an interactive DSG program for data visualization in the X Window System^{TM} that allows the exploration of multivariate data, mostly by manipulating and displaying scatterplots in arbitrary dimensions. Examples of its use range from customer panel data (Koschat and Swayne, 1996) to experimental measurements of laser performance and high–dimensional geometric objects (Buja, Cook, and Swayne, 1996).

XploRe (Härdle et al., 1995) is a statistical computing environment (SCE) with a built–in set of statistical methods; users can extend XploRe both by using a high–level programming language and by loading functions from external libraries. It supports highly interactive graphics in arbitrarily many displays (i. e., windows). XploRe is portable: it has its own editors for data and programs, so it has a consistent interface no matter what underlying hardware platform is used. XploRe is networked: it can be accessed remotely using a Java–based interface (Kötter, 1997), and its help system can be used from within a standard web browser. (Using help files locally is also supported.)

For more information about XploRe, see Härdle et al. (1995); it is based on XploRe 3 but most of it is still valid for XploRe 4. For an update on XploRe 4 refer to Klinke and Kötter (1996).

The new link connecting the three programs ArcView, XGobi, and XploRe is an extension of the existing bidirectional link between ArcView and XGobi (see Symanzik, Majure, Cook, and Megretskaia, 1997b, for technical details; see Cook, Majure, Symanzik, and Cressie, 1996, and Cook, Symanzik, Majure, and Cressie, 1997, for examples of its use). Symanzik et al. (1997b) also provides an overview on similar projects where GIS and statistical packages have been linked.

The addition of XploRe to the ArcView/XGobi link now gives us the following functionality: We can issue XploRe commands from within XGobi (and, in the near future, from within ArcView); we can interactively exchange data among these three programs without manually manipulating external files, and we can link different types of views uniquely available in each program. For example, we can link a view with geographic information and sampling locations in ArcView, a pairwise scatterplot matrix of the attribute data in XploRe, and a grand tour of the same attribute data in XGobi.

The ArcView/XGobi/XploRe link is based on the use of Remote Procedure Calls (RPCs) as the underlying Interprocess Communication (IPC) mechanism. (See Stevens, 1990, and Corbin, 1991, for general information on RPCs.) Only minor modifications were required in ArcView and XGobi, because RPC communication had already been established between those two, but the complete RPC functionality had to be added to XploRe. Once all the software extensions are complete, not only will ArcView, XGobi, and XploRe be able to communicate with each other, but extensions to support dynamic communication with other software can then be added easily. Further implementation details will be described in a future paper.

In Section 2 of this paper we describe current features of the ArcView/XGobi/XploRe environment, in particular the XGobi/XploRe direction of the link and aspects of spatial data analysis in XploRe. In Section 3 we describe features that we plan to include in the near future. The ArcView/XploRe link still remains to be built: we foresee no technical problems in completing this work, and it will be completed as time permits. In Section 4 we discuss more ambitious extensions of this work.

The software has been developed and tested on Sun^{TM}/Sparc^{TM} workstations and DEC^{TM} alphastations. The ArcView/XGobi part of the link can be downloaded from the WWW at the URL `http://www.gis.iastate.edu/XGobi-AV2/XGobi-AV2.html`. XploRe for UNIX® systems is available for free from `http://wotan.wiwi.hu-berlin.de/xplore/Index.html`.

2 Current Features and Applications

The link between ArcView and XGobi has already been described in Cook et al. (1996, 1997). Its main features include the multivariate link, the SCDF link, the variogram–cloud link, the spatially lagged scatterplot link, and the multivariate variogram–cloud link. In this

X Window System is a trademark of MIT.

Sun is a trademark of Sun Microsystems, Inc.

Sparc is a trademark of Sun Microsystems, Inc.

DEC is a trademark of Digital Equipment Corporation.

UNIX is a registered trademark of UNIX Systems Laboratories.

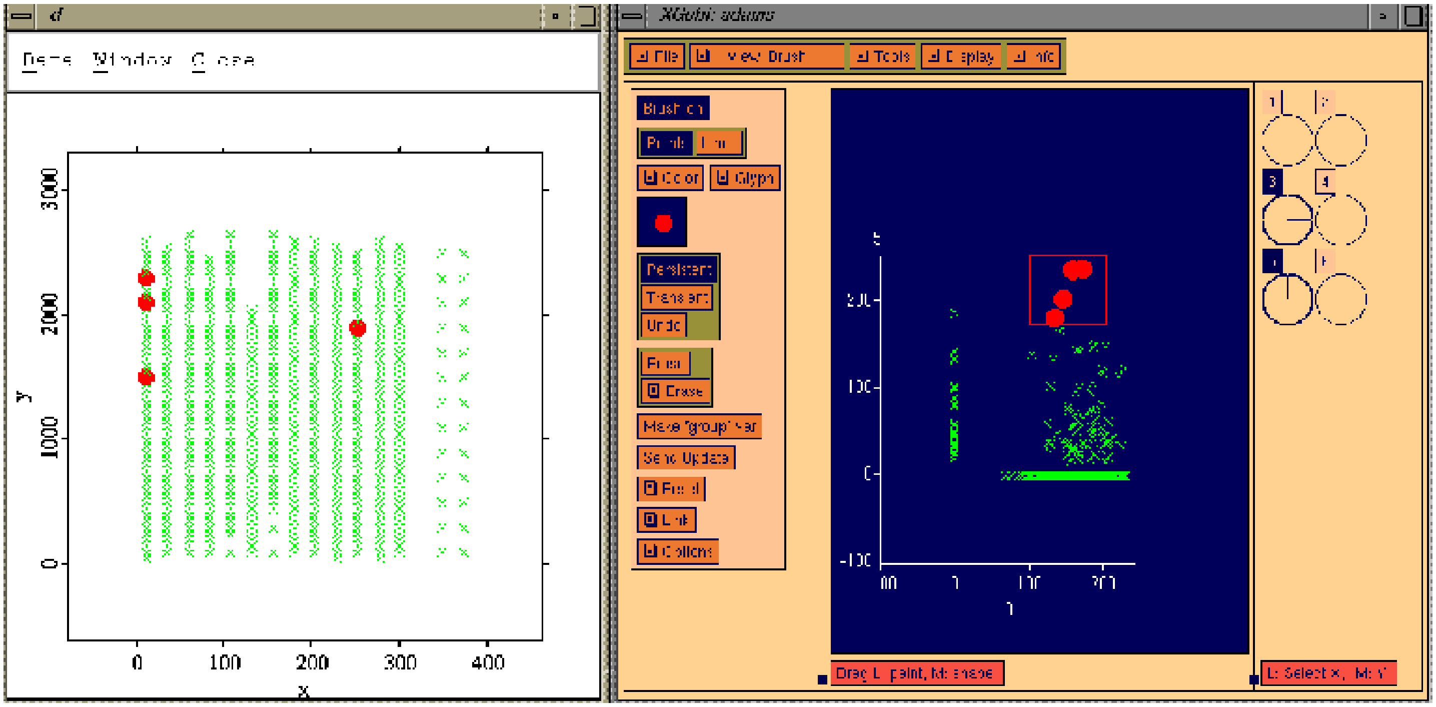

Figure 1: Linked Brushing between XploRe (left Window) and XGobi (right Window).

section, we describe new features and applications that have become available with the addition of the link between XGobi and XploRe.

The data set used throughout this section has been provided by Carol Gotway and has been previously analyzed in Gotway and Hartford (1996). The data has been collected at a field on the Milton Ruhter farm in Adams County, Nebraska, in 1992–1993. It is an example of a precision agricultural data set where a Global Positioning System (GPS) has been used to obtain corn yield measurements. These measurements have been averaged over an area of 30 feet and result in 638 corn yield observations (in bushels per acre). After harvest, 104 samples of soil cores were taken and their cumulative nitrate (NO_3) concentration to a depth of four feet were determined (in pounds per acre). All distances in this example are in feet.

2.1 Linked Brushing Between XGobi and XploRe

The communication between XGobi and XploRe permits bidirectional linked brushing between these two packages. Figure 1 shows the spatial locations where either yield or nitrate (or both) measurements were taken (left XploRe view) while the related measurements are displayed in a scatterplot (right XGobi view). Nitrate is plotted vertically and yield is plotted horizontally. Missing values where either the Nitrate or the yield measurement is not available are represented as 0. We brushed four high nitrate values in XGobi. Through linked brushing, the related spatial locations immediately are brushed in XploRe, using the same color, glyph, and size. Linked brushing also works when points are selected within XploRe.

2.2 Additional Smoothers in XGobi

Currently, XGobi provides four built–in smoothers, i. e., Mean, Median, Nadaraya–Watson, and Spline. These are available when the user opens the *Tools* menu, then selects *Smooth*, and opens the *Smoothers* submenu (see Figure 2). Options that are available appear in solid fonts while options that are not available are "greyed out."

However, in the linked environment, five additional smoothers are available using XploRe. To use those smoothers, the user first executes *Start XploRe* in the *Tools* menu before opening the *Smoothers* submenu. In that case, the five smoothers which are greyed out when XGobi runs independently are available (see Figure 2): the code to link in NADAWAT (Nadaraya–Watson estimator) and LOCPOL (local linear estimator) has already been implemented; LPREGEST, LPREGAUTO, and KNN will be completed soon.

When the user selects one of these smoothers in

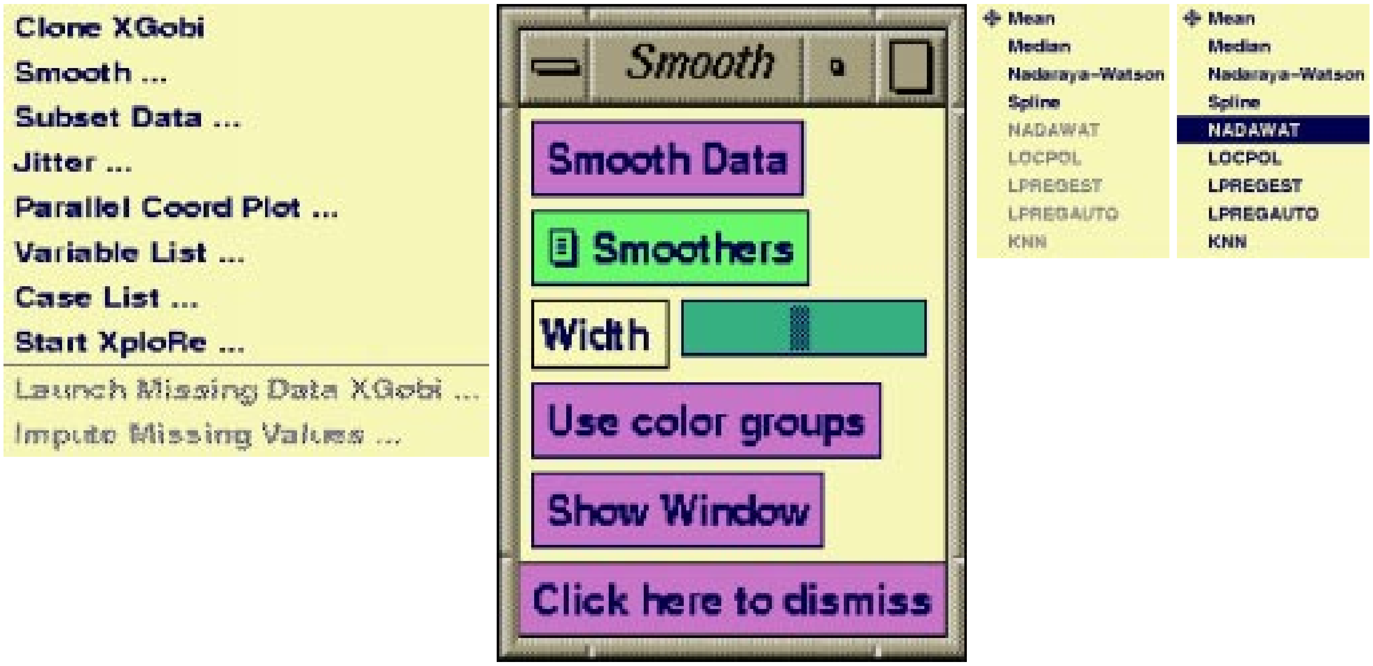

Figure 2: XGobi Windows and Menus: *Tools* Menu, *Smooth* Window, *Smoothers* Submenu when the link to XploRe is inactive, and *Smoothers* Submenu when the link to XploRe is active [from left to right].

XGobi, an XploRe command is generated that contains all the necessary data and parameters for the selected smoother. This command is passed to XploRe (using the RPC mechanism) where it is executed. The resulting smoothed line is passed back to XGobi where it is finally displayed. In a simple example with only two data points in XGobi, the XploRe command created for the NADAWAT smoother could look like

```
  XGobiSmooth=xgobinadawat(#(-16384,16384),
    #(-16384,16384), #(0,0), 1000)
```

where the first two parameters of the input to the XploRe function `nadawat` are the vectors of the x and y screen coordinates of the data points used in XGobi, the third parameter is the group identification (in this case there are no groups, i.e., differently colored datapoints), and the last parameter is the smoothing parameter (here, bandwidth). The result of this calculation is assigned to the XploRe variable `XGobiSmooth`. The user does not see any of this underlying communication and only experiences a minor time delay if a large data set is passed between the two programs.

On the XploRe side, a library (`xgobismooth`) has been written that contains the appropriate macros `xgobinadawat` and `xgobilocpol`. A special startup program `xgobistart.xpl` ensures that the library is loaded into XploRe when *Start XploRe* is executed.

The communication between XGobi and XploRe is much richer than the communication between XGobi and S (Becker, Chambers, and Wilks, 1988), described in Swayne, Buja, and Hubbell (1991a) and Swayne et al. (1991b). The "link" between XGobi and S described in those papers means two different things. In the simplest case, it describes a way of invoking XGobi from within S, on S data files, by executing an S function. After that invocation, no further communication can be done. The richer "linking" is much closer to the communication described in this paper: a communication between XGobi and S is established via pipes or sockets, both of which are UNIX interprocess communication methods. Commands are generated in XGobi (or XSmooth) and sent to S for execution; S executes those commands (on data files already present in the S format) and passes some results back for plotting.

This sounds in principle very much like the connection described here between XGobi and XploRe, but the earlier work was much more limited, both in functionality and in scope.

Problems of the XGobi/S link arose through the use of pipes and sockets in contrast to the use of RPCs (Remote Procedure Calls) in the XGobi/XploRe link. We found that IPC based on pipes and sockets is much more difficult to implement than IPC based on RPCs. In particular, the error handling using RPCs is very convenient. The result of each RPC either indicates the error type and allows immediate action from the software or it states that no error occurred. Using pipes and sockets, it

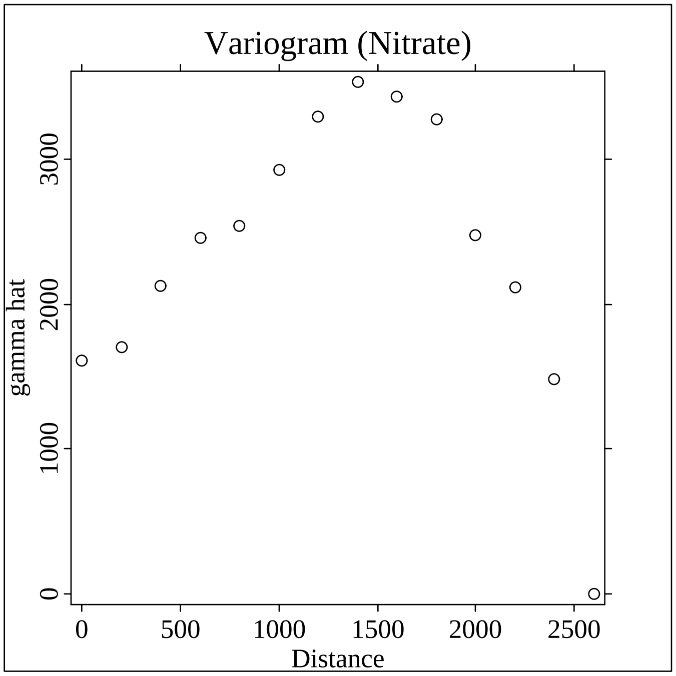

Figure 3: Variogram Estimate of Nitrate.

was very difficult to handle errors. If the command to S was imperfect in any way, such as getting the data name wrong, XGobi was helpless to handle the error message.

Another problem arose due to the fact that S in 1991 was not specifically designed to support interprocess communication (IPC) and to parse externally provided commands. Only version 4 of S, available as a beta release since 1996, has been designed to support IPC and might be a future candidate to be added to the link.

Another aspect made the implementation of the XGobi/XploRe link very easy: Since developers of both packages, XGobi and XploRe, contributed to this work, we could make modifications and additions directly at the C/C++ source code level and did not have to implement any patches outside these packages.

In the XGobi/XploRe link each program can be the source of the data, and each can invoke the other program. S was never invoked from XGobi, and XGobi never transmitted data to S; it sent only commands. The linking here has therefore much more potential: XGobi and XploRe can share brushing information. Also, XploRe can interactively provide additional information to XGobi, such as smoothed lines, that can not be calculated in XGobi itself (see Section 4 for further possibilities).

In the terminology of Goodchild, Haining, and Wise (1992), the S/XGobi link is an example of loose coupling where ASCII files and files in the S format are exchanged. The XGobi/XploRe link is an example of close coupling and allows each program to directly access the functionality of the other.

2.3 Spatial Data Analysis in XploRe

Based on the overview given in the introduction, it should be obvious that XploRe is not a package specifically designed for spatial data analysis but can be used for all types of statistical analyses, and that it is extensible. We have made use of XploRe's high–level programming language to write our own functions for spatial data analysis. The new XploRe library *"spatial"* is a container for these functions. As an example, Figure 3 shows a variogram estimate of Nitrate for the precision agricultural data set. It has been calculated in XploRe, using the following code:

```
; z [xcoord, ykoord, variable, missing?]
;
;   0 = missing
;   1 = available
;

proc (values, count) = variogram (z, lagnumber, lagvalue)
  values = matrix ((lagnumber + 1), 1)
  values = values * 0
  count = matrix ((lagnumber + 1), 1)
  count = count * 0
  critdist = ((2 * lagnumber + 1) * lagvalue / 2)
  z = paf (z, z[,4] == 1)
    ; delete missing values
  i = 1
```

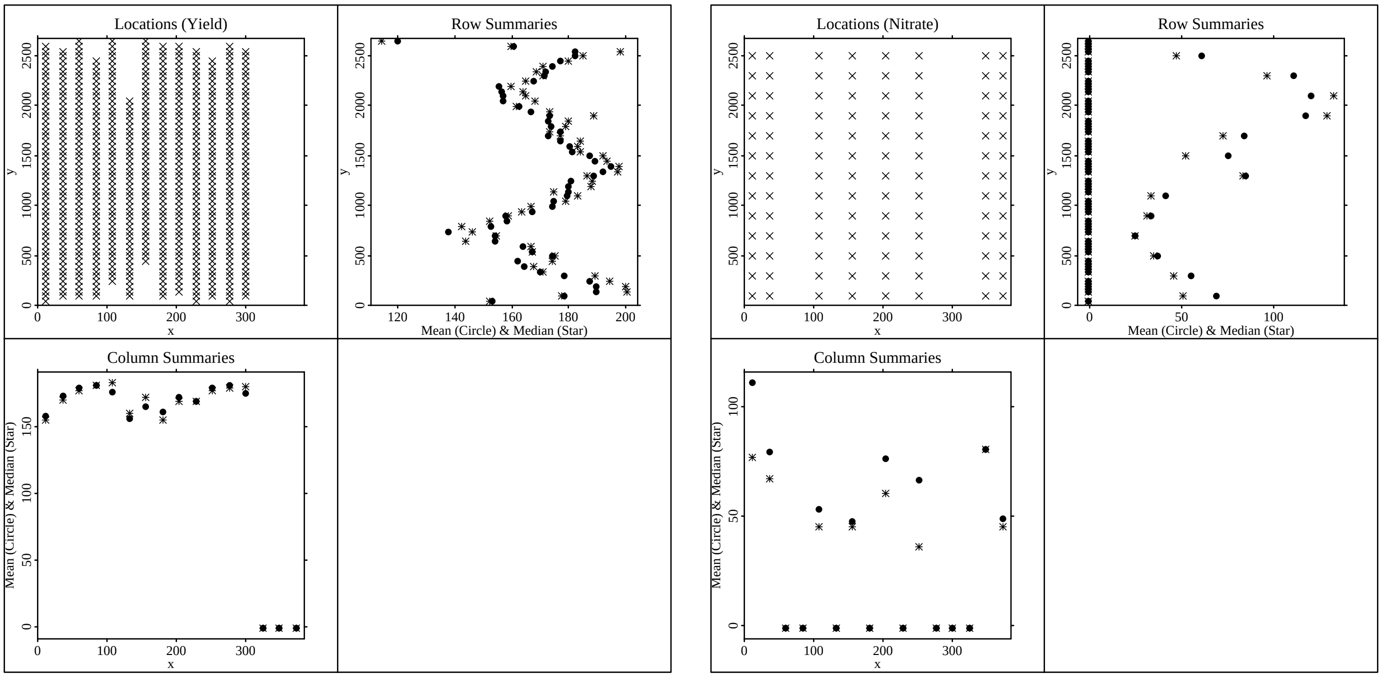

Figure 4: Mean and Median Summaries of Nonstationarity for Yield (left) and Nitrate (right).

```
  while (i <= rows(z))
    i
    j = 1
    while (j <= rows(z))
      if (i != j)
        dist = sqrt ((z[i, 1] - z[j, 1])^2 +
                 (z[i, 2] - z[j, 2])^2)
        if (dist > critdist)
          lag = -1
        else
          lag = floor ((2 * dist / lagvalue + 1) / 2)
        endif
          ; determine lag of current observation
        if (lag >= 0)
          ii = lag + 1
          values [ii, 1] = values [ii, 1] +
                             (z[i, 3] - z[j, 3])^2
          count [ii, 1] = count [ii, 1] + 1
        endif
          ; accumulate values and count according to lag
      endif
      j = j + 1
    endo
    i = i + 1
  endo
  i = 1
  while (i <= lagnumber + 1)
    if (count [i, 1] > 0)
      values [i, 1] = values [i, 1] / (2 * count [i, 1])
    endif
    i = i + 1
  endo
endp
```

The following sequence of XploRe commands has been used to calculate the variogram estimate of Nitrate and to draw the plot visible in Figure 3.

```
library("spatial/spatial")
  ; add the spatial library

adams = startadams ()
  ; initialize the data set

varadamsnit = variogram (adams[,1:2] ~ adams[,5:6] , 13, 200)
  ; calculate the variogram of Nitrate

d = createdisplay (1,1)
show (d, 1, 1, (0:13) * 200 ~ varadamsnit.values)
setgopt (d, 1, 1, "title", "Variogram (Nitrate)",
           "xlabel", "Distance", "ylabel", "gamma hat")
  ; create the display and draw the data

print (d, "variogram:nitrate.ps")
  ; produce the postscript output
```

Here, `adams` is a matrix that contains the precision agricultural data where columns 1 and 2 are the $x-$ and $y-$coordinates of the sampling locations, columns 3 and 4 are related to Yield, and columns 5 and 6 are related to Nitrate.

Figure 4 (also created in XploRe) shows row/column summaries for Yield and Nitrate. For each row and for each column, the median ($*$) and the mean ($\bullet$) of the measured variable are displayed with respect to their spatial location. If an entire row/column contains no measurements, the value 0 has been assigned to the mean/median plot. The four large observations of Nitrate, depicted in a non–spatial sense already in Figure 1, have an extreme influence on the means in columns 12 and 252, but their influence is less significant on rows.

This data set clearly deserves further attention. Our purpose in this paper, however, is limited to the demonstration of the use of the linked software environment for conducting spatial data analysis, so we leave the complete analysis for the future.

3 Still To Do

When ArcView, XploRe, and XGobi are linked, we can maintain geographical information and attribute data at one single location, in the ArcView data base, and transfer selected data to XGobi (and, soon, to XploRe) whenever needed. Considering a full triangular communication structure intended for the three packages, direct communication between ArcView/XGobi and XGobi/XploRe is possible. The final link between ArcView and XploRe is still missing, and we will discuss shortly the requirements for its implementation. The following examples demonstrate why this final link is important to support spatial data analysis.

In Symanzik, Majure, and Cook (1996, 1997a), we used a satellite image of the Lake Icaria watershed, located in southwest Iowa (near Corning) where 3 ranges of wavelengths of electromagnetic radiation have been measured. The measurements of the three variables *Icariac1*, *Icariac2*, and *Icariac3* plus additional information were originally available in ArcView. In addition, the variable *Clust* has been added to the data set, containing the result of a hierarchical cluster analysis. However, five steps were required before further analysis could be conducted using *Clust*. The data first had to be written to an ASCII file, it had to be loaded into S, the variable *Clust* had to be calculated, the result had to be written into another ASCII file, and this had to be reloaded back into ArcView. Then, finally the link between ArcView and XGobi was activated. Within the ArcView/XGobi/XploRe environment, these steps should be done by pressing only a few buttons within ArcView to activate the required functionality within XploRe and XGobi.

In Majure, Cressie, Cook, and Symanzik (1996), forest health data had been passed back and forth between ArcView (where the data originally came from), S (to fit a large–scale regression model and to conduct small–scale variogram analyses on the residuals), and XGobi (to visually assess regression diagnostics and outliers among the residuals). Using the ArcView/XGobi/XploRe environment, the same kind of analysis should be possible in an easier way than transferring data through external ASCII files as it was done here.

The requirements for the link between ArcView and XploRe are well understood, and we do not anticipate any technical difficulty in its implementation. Basically, all required components exist. The RPC mechanism exists to link ArcView and XploRe and has been successfully used for the other directions of the link. We also have experience in modifying the Graphical User Interface (GUI) of ArcView and writing "Avenue functions" in ArcView to support RPCs.

What is missing then? We still have to implement a dialogue box in ArcView (using Avenue) that allows the user to issue XploRe commands based on ArcView variable names. Invisible to the user, the RPC mechanism passes commands (and data) from ArcView to XploRe where commands are executed. Results are returned to ArcView. The dialogue box we have in mind should be very similar to that of ArcView *Query Builder* so that ArcView users who are unfamiliar with XploRe nevertheless have easy access to its spatial data analysis functionality.

4 Summary and Directions

Even though the dynamic link among ArcView, XploRe, and XGobi will allow us to dynamically link different types of plots among these programs, interactively exchange data, and directly issue XploRe commands from within XGobi (and hopefully from ArcView soon), there remain several other features we would like to see implemented in the near future.

Linked identification would be a natural next step: If we identify one point in one view, additional information such as name, index number, and values are displayed in this view. In the linked case, the various identification styles of all three programs could be activated at the same time: XGobi shows only the case label, for example, while ArcView shows additional textual information, e. g., the date when this sample has been taken and the name of the person that took these measurements, and XploRe shows all values of the numerical attributes of the selected point.

With an underlying statistical package such as XploRe we will be able to do "statistical brushing"; that is, graphical attributes such as color, glyph, and size are set by some function, and points are appropriately brushed or highlighted in the linked graphical displays. Possible calculations might be as easy as statistically brushing the maximum value or the inner 50% of one of the variables, and can be more complicated such as brushing all points where the squared sum of all variables is larger than a given value.

Two features could be added to the XGobi/XploRe part of the link that extend existing features in XGobi. Currently, XGobi supports a handful of variable transformations, e. g., natural logarithm and square root. XploRe could be easily used to provide additional transformations such as general power transformations, trigonometric functions, or transformations that affect only parts of the data, e. g., to assign the constant c to all points that are larger than c.

Similarly, imputations of missing values could be done in XploRe. Recent additions to XGobi allow missing values, and sets of user–supplied imputed values can be examined, but XGobi does not provide any means of imputing values. It should be easy to implement various imputation methods in XploRe, and then invoke these functions from XGobi, just as XploRe's smoothers can be used from XGobi now.

To further extend XploRe's capability for spatial data analysis, it seems to be possible to call the C routines from Venables/Ripley (1994) from within XploRe. This will immediately provide us with additional functions (that are well–tested and well–documented) for spatial data analysis within the ArcView/XGobi/XploRe environment.

Acknowledgements

This work was initiated while Symanzik was visiting Humboldt–Universität zu Berlin in January 1997. This research was supported by the Deutsche Forschungsgemeinschaft, Sonderforschungsbereich 373 "Quantifikation und Simulation ökonomischer Prozesse", Humboldt–Universität zu Berlin. We would like to thank Carol A. Gotway for providing us with the precision agricultural data set used as an example through this paper. Thanks are also due to Andreas Buja for his helpful comments on this paper.

References

Becker, R., Chambers, J., and Wilks, A. (1988). *The New S Language — A Programming Environment for Data Analysis and Graphics.* Wadsworth and Brooks/Cole, Pacific Grove, CA.

Buja, A., Cook, D., and Swayne, D. F. (1996). Interactive High–Dimensional Data Visualization. *Journal of Computational and Graphical Statistics*, 5(1):78–99.

Cook, D., Majure, J. J., Symanzik, J., and Cressie, N. (1996). Dynamic Graphics in a GIS: Exploring and Analyzing Multivariate Spatial Data Using Linked Software. *Computational Statistics*, 11(4):467–480.

Cook, D., Symanzik, J., Majure, J. J., and Cressie, N. (1997). Dynamic Graphics in a GIS: More Examples Using Linked Software. *Computers and Geosciences*, Forthcoming: Contains live video footage, available at `http://www.public.iastate.edu/~dicook/compgeo.html`.

Corbin, J. R. (1991). *The Art of Distributed Applications: Programming Techniques for Remote Procedure Calls.* Springer, New York, Berlin, Heidelberg.

Goodchild, M. F., Haining, R. P., and Wise, S. (1992). Integrating GIS and Spatial Data Analysis: Problems and Possibilities. *International Journal of Geographical Information Systems*, 6(5):407–423.

Gotway, C. A. and Hartford, A. H. (1996). Geostatistical Methods for Incorporating Auxiliary Information in the Prediction of Spatial Variables. *Journal of Agricultural, Biological, and Environmental Statistics*, 1(1):17–39.

Haining, R., Ma, J., and Wise, S. (1996). Design of a Software System for Interactive Spatial Statistical Analysis Linked to a GIS. *Computational Statistics*, 11(4):449–466.

Härdle, W., Klinke, S., and Turlach, B. A. (1995). *XploRe: An Interactive Statistical Computing Environment.* Springer, New York, Berlin, Heidelberg.

Klinke, S. and Kötter, T. (1996). XploRe 4 — A Statistical Computing Environment. In *SoftStat '95 Advances in Statistical Software 5*, pages 113–122, Stuttgart. Lucius & Lucius.

Koschat, M. A. and Swayne, D. F. (1996). Interactive Graphical Methods in the Analysis of Customer Panel Data (with discussion). *Journal of Business and Economic Statistics*, 14(1):113–132.

Kötter, T. (1997). Interactive Interfaces of Statistical Software for the Internet. In *SoftStat '96 Advances in Statistical Software 6*, pages 153–158, Stuttgart. Lucius & Lucius.

Majure, J. J., Cressie, N., Cook, D., and Symanzik, J. (1996). GIS, Spatial Statistical Graphics, and Forest Health. In *Proceedings of the Third International Conference/Workshop on Integrating GIS and Environmental Modeling, Santa Fe, NM, January 21–26, 1996*, Santa Barbara, CA. National Center for

Geographic Information and Analysis. CD and `http://www.ncgia.ucsb.edu/conf/SANTA_FE_CD-ROM/main.html`.

Stevens, W. R. (1990). *UNIX Network Programming.* Prentice–Hall, Englewood Cliffs, NJ.

Swayne, D., Buja, A., and Hubbell, N. (1991a). XGobi Meets S: Integrating Software for Data Analysis. *Computing Science and Statistics*, 23:430–434.

Swayne, D. F., Cook, D., and Buja, A. (1991b). XGobi: Interactive Dynamic Graphics in the X Window System with a Link to S. In *ASA Proceedings of the Section on Statistical Graphics*, pages 1–8, Alexandria, VA. American Statistical Association.

Symanzik, J., Majure, J. J., and Cook, D. (1996). Dynamic Graphics in a GIS: A Bidirectional Link between ArcView 2.0 and XGobi. *Computing Science and Statistics*, 27:299–303.

Symanzik, J., Majure, J. J., and Cook, D. (1997a). Dynamic Graphics in a GIS: A Bidirectional Link between ArcView 2.1 and XGobi — An Update. In *Second World Conference of the International Association for Statistical Computing (IASC), Pasadena, California, USA, February 19–22, 1997.* Forthcoming.

Symanzik, J., Majure, J. J., Cook, D., and Megretskaia, I. (1997b). Linking ArcView 3.0 and XGobi: Insight Behind the Front End. Technical Report 97–10, Department of Statistics, Iowa State University, Ames, Iowa.

Venables, W. N. and Ripley, B. D. (1994). *Modern Applied Statistics with S–Plus.* Springer, New York, Berlin, Heidelberg.