Journal of Statistical Software

October 2010, Volume 36, Issue 13. http://www.jstatsoft.org/

Interactive Teaching Tools for Spatial Sampling

Adrian W. Bowman
University of Glasgow

Iain Gibson
University of Glasgow

E. Marian Scott
University of Glasgow

Ewan Crawford
University of Glasgow

Abstract

The statistical analysis of data which is measured over a spatial region is well established as a scientific tool which makes considerable contributions to a wide variety of application areas. Further development of these tools also remains a central part of the research scene in statistics. However, understanding of the concepts involved often benefits from an intuitive and experimental approach, as well as a formal description of models and methods. This paper describes software which is intended to assist in this understanding. The role of simulation is advocated, in order to explain the meaning of spatial correlation and to interpret the parameters involved in standard models. Realistic scenarios where decisions on the locations of sampling points in a spatial setting are required are also described. Students are provided with a variety of sampling strategies and invited to select the most appropriate one in two different settings. One involves water sampling in the lagoon of the Mururoa Atoll while the other involves sea bed sampling in a Scottish firth. Once a student has decided on a sampling strategy, simulated data are provided for further analysis. This extends the range of teaching activity from the analysis of data collected by others to involvement in data collection and the need to grapple with issues of design. It is argued that this approach has significant benefits in learning.

The software which implements these tools is built on existing R packages, using **rpanel** controls for the **geoR** geostatistical simulation and modelling tools. The operation and construction of the software are described in detail. The software is made available as additional functions `rp.geosim`, `rp.mururoa` and `rp.firth` in the **rpanel** package.

Keywords: interactive graphics, graphical user interface, learning, sampling, simulation, spatial statistics, R, teaching.

1. Introduction

The statistical analysis of spatial data has been a major growth area of statistical research for around three decades and it continues to occupy a place at the very heart of current research. The need to adapt statistical analysis to spatial structures has generated very important methodological innovations. On the other hand, the existence of appropriate spatial methods has had enormously significant impact on a wide variety of application areas, from image processing to environmental problems. Cressie (1993) gives an authoritative and comprehensive overview of spatial concepts and techniques while a wide variety of more recent texts offer different perspectives on the subject and address important application areas, particularly environmental science (Webster and Oliver 2001; Piegorsch and Bailer 2005; Barnett 2006; Diggle and Ribiero 2007).

The concepts associated with spatial variation, and in particular spatial correlation, need to be firmly understood if spatial analyses are to be properly implemented and interpreted. When these concepts are met for the first time, they are not always easy to grasp, particularly for those whose background lies in application areas where quantitative traditions are not strong. However, even those who are able to approach the topic from a strong background in the mathematical sciences often benefit from opportunities to develop a more intuitive understanding to complement the theory.

Appropriate use of software can make substantial contributions towards these aims. A number of excellent packages to fit spatial models are now available and the ability to use these on relevant data allows a shift of focus from the algorithmic to the conceptual. An important example implemented in the R system for statistical computing (R Development Core Team 2010) is the **geoR** package (Ribeiro and Diggle 2001) which provides a wide variety of tools for the analysis of geostatistical data. This makes use in turn of the **sp** classes and methods (Pebesma and Bivand 2005; Bivand, Pebesma, and Gómez-Rubio 2008) for spatial data and the **RandomFields** (Schlather 2009) package for the simulation of data from random fields.

While direct use of analysis packages enables students to engage with real applications, there remains scope for software whose role is to explain the meaning of spatial models and to allow students to explore standard concepts such as the meaning of variation in a spatial setting. The `ShowModels` function in the **RandomFields** package, described by Schlather (2001), is a good example of what can be achieved with standard mouse-click interaction on an R graphics window. However, the advent of graphical user interface (GUI) software in R greatly extends the scope for the construction and use of interactive tools. There are now many different systems to construct GUI controls, such as **iplots** (Urbanek and Theus 2003), **JGR** (Helbig, Urbanek, and Fellows 2010), **RGtk2** (Lawrence and Temple Lang 2010) and **gWidgets** (Verzani 2007). The particular niche occupied by the **rpanel** package (Bowman, Crawford, Alexander, and Bowman 2007), which in turn is built on the **tcltk** system (Dalgaard 2001), is expressed in the aim to make the addition of interactive graphical controls as simple as possible. This paper reports on the the use of **rpanel** to create software which allow students to engage graphically with spatial concepts. It is available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=rpanel`.

The role of simulation is discussed in Section 2, where software to provide interactive control of model parameters and displays is described. Sections 3 and 4 discuss two scenarios where data are to be sampled from a spatial region. Students are required to make a decision on how this sampling should be conducted, with the software providing a variety of options, and some simple analysis of the collected data. Some final discussion is given in Section 5. Issues of software design are discussed in an appendix.

2. Simulation of spatial data

Simulation provides a very natural tool to promote an intuitive understanding of random variation in a variety of settings, simply by viewing repeated realizations of data. For spatial data, a helpful starting point is the simulation of random surfaces which are generated by a Gaussian process. If, for simplicity, the mean value is set to 0, then the process can be defined over a spatial region by the covariance function $C(s, t)$, where s and t denote spatial locations. When the process is (weakly) stationary, C is a function only of $h = ||s - t||$, the distance between the two spatial locations. The material described in this paper uses the Matérn covariance function

$$C(h) = \frac{\sigma^2}{2^{\kappa-1}\Gamma(\kappa)} \left(\frac{h}{\phi}\right)^{\kappa} K_{\kappa}\left(\frac{h}{\phi}\right),$$

where $K_\kappa(x)$ denotes the modified Bessel function of the third kind of order κ. This offers a flexible family of covariance functions with a degree of control of shape expressed in the parameter κ. For example, the standard exponential model is obtained by setting $\kappa = 0.5$. Since many covariance functions have broadly similar shapes, it is the role of the variance (σ^2) and correlation range (ϕ) parameters which are of most interest. However, in the parameterization shown above, the range is also directly affected by κ. This can be remedied by adopting the parameterization of Handcock and Wallis (1994), defined by $\phi = \theta/(2\sqrt{\kappa})$, which gives the new range parameter θ a clear interpretation, unaffected by the setting of κ. This parameterization has been adopted here.

For visualization, the random surface generated by the Gaussian process can be evaluated at a fine grid of spatial locations. A simple approach is to construct the covariance matrix Σ whose (i, j)th element contains C evaluated at grid points i and j. A Cholesky, or equivalent, decomposition of Σ provides a matrix which, when multiplied by a vector of independent standard normal random variables, produces a vector of data whose covariance matrix is Σ. This approach can be very inefficient when large samples are involved but a variety of other computational approaches are available. The `grf` function from the **geoR** package in R uses the `GaussRF` function from the **RandomFields** package of Schlather (2009), while Rue and Held (2005) and Rue and Follestad (2002) describe how Gaussian Markov random fields can be used to provide very efficient approximations. The **spectralGP** package of Paciorek (2007) offers another route, also in R, although this requires grid sizes based on powers of 2.

Direct calls to these applications are straightforward for those familiar with R. However, the use of a graphical user interface allows the focus to be directed towards repeated, and possibly rapid, simulations without the distraction of intervening code. Figure 1 shows displays created by the **rpanel** function `rp.geosim()`. Repeated simulations can be generated by pressing a button, while the parameter values can be controlled by the sliders. The instant visual feedback from the data patterns generated by different values promotes good understanding of what is meant by the variance of the process, referred to here by the standard terminology of ‘partial sill’, and the correlation range parameter. It is particularly helpful to explore the relationship between the range parameter and the smoothness of the resulting random surface. The speed with which new simulations can be generated makes the connection between parameter values and patterns easier to establish, without the intervening distraction of explicitly re-executing code. A checkbox allows an alternative display in three-dimensional form, using the tools provided by the **rgl** package of Adler and Murdoch (2010) to create a 3D scatterplot and surface facility in the **rpanel** function `rp.plot3d`.

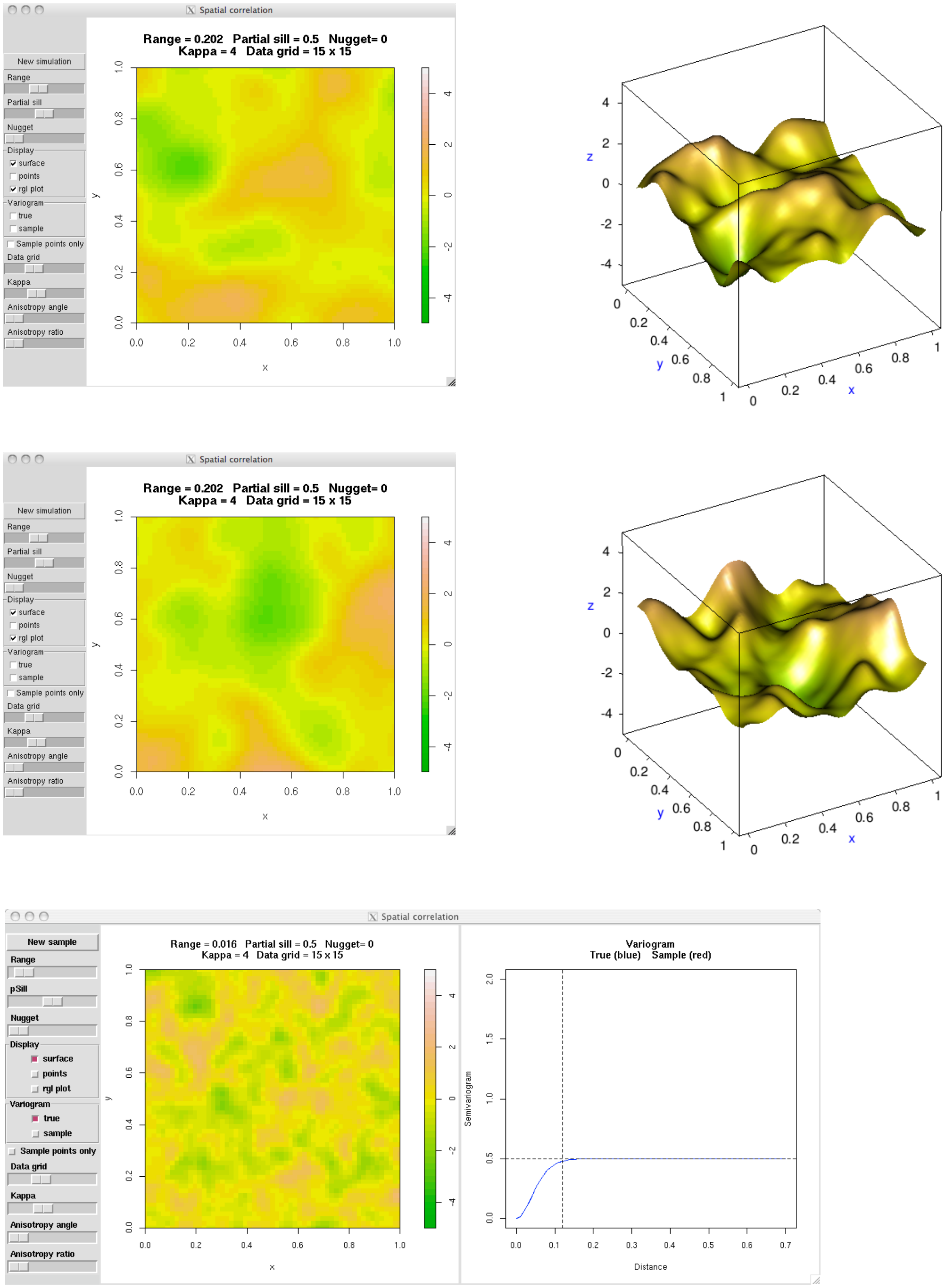
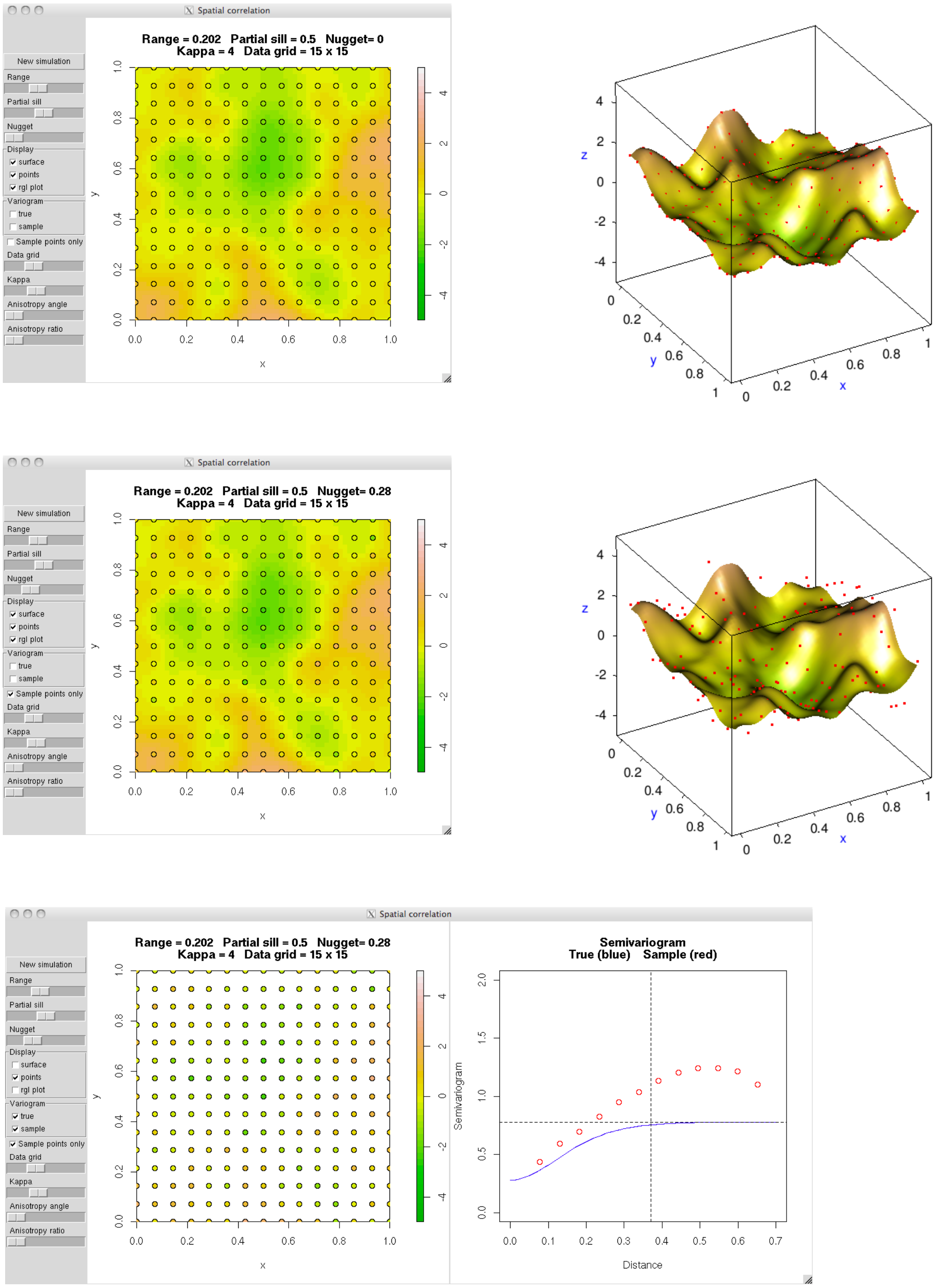


Figure 1: Simulations created by the **rpanel** function `rp.geosim()`. The top row shows a contour plot, and **rgl** plot, of a Gaussian process with the Matérn covariance function, using the parameter values shown in the plot title. The middle row shows the same plots for a second set of data simulated under the same conditions. The bottom panel shows a contour plot of data simulated with a smaller range parameter, together with a plot of the semivariogram.



Spatial correlation

New simulation

Range

Partial sill

Nugget

Display
☑ surface
☑ points
☑ rgl plot

Variogram
□ true
□ sample

□ Sample points only

Data grid

Kappa

Anisotropy angle

Anisotropy ratio

Range = 0.202 Partial sill = 0.5 Nugget= 0
Kappa = 4 Data grid = 15 x 15

Spatial correlation

New simulation

Range

Partial sill

Nugget

Display
☑ surface
☑ points
☑ rgl plot

Variogram
□ true
□ sample

☑ Sample points only

Data grid

Kappa

Anisotropy angle

Anisotropy ratio

Range = 0.202 Partial sill = 0.5 Nugget= 0.28
Kappa = 4 Data grid = 15 x 15

Spatial correlation

New simulation

Range

Partial sill

Nugget

Display
□ surface
☑ points
□ rgl plot

Variogram
☑ true
☑ sample

☑ Sample points only

Data grid

Kappa

Anisotropy angle

Anisotropy ratio

Range = 0.202 Partial sill = 0.5 Nugget= 0.28
Kappa = 4 Data grid = 15 x 15

Semivariogram
True (blue) Sample (red)

Figure 2: These plots show the simulated Gaussian process displayed in the middle row of Figure 1. In the top row, sampling points for the extraction of data from this process are indicated. In the middle row, further independent variation has been added at these sampling positions, to simulate a nugget effect. In the bottom panel, the underlying process has been removed from the plot, leaving only the data which would be available in a sampling exercise. The true and sample variograms have also been added.

The surfaces shown in Figure 1 represent the entire spatial process. However, it would be rare to observe a process in complete form and so it is then helpful to discuss the process of measurement, leading to observed data corresponding to a more limited set of spatial locations. A regular grid of sampling locations is shown in the top two panels of Figure 2. This raises the issue of measurement error, or more generally of additional small–scale variation, in the so-called 'nugget' effect. A slider is available to control the variance of this additional source of variation, represented in the discrepancies between the colors of the points and the underlying surface in the contour plot, and more obviously in the separation of points and surface in the 3D plot, both shown in the middle panels of Figure 2. Again, repeated simulations and altered nugget values communicate the meaning of this parameter effectively. It is also useful to be able to separate simulations which generate new measurement errors on a fixed surface from simulations which generate new values for both. The former type of simulation can be implemented by checking the '`Sample points only`' checkbox. This can helpfully illustrate a discussion of what it means to take repeated samples of spatial data, and whether the underlying process or simply the measurement errors will have changed between the repeat visits.

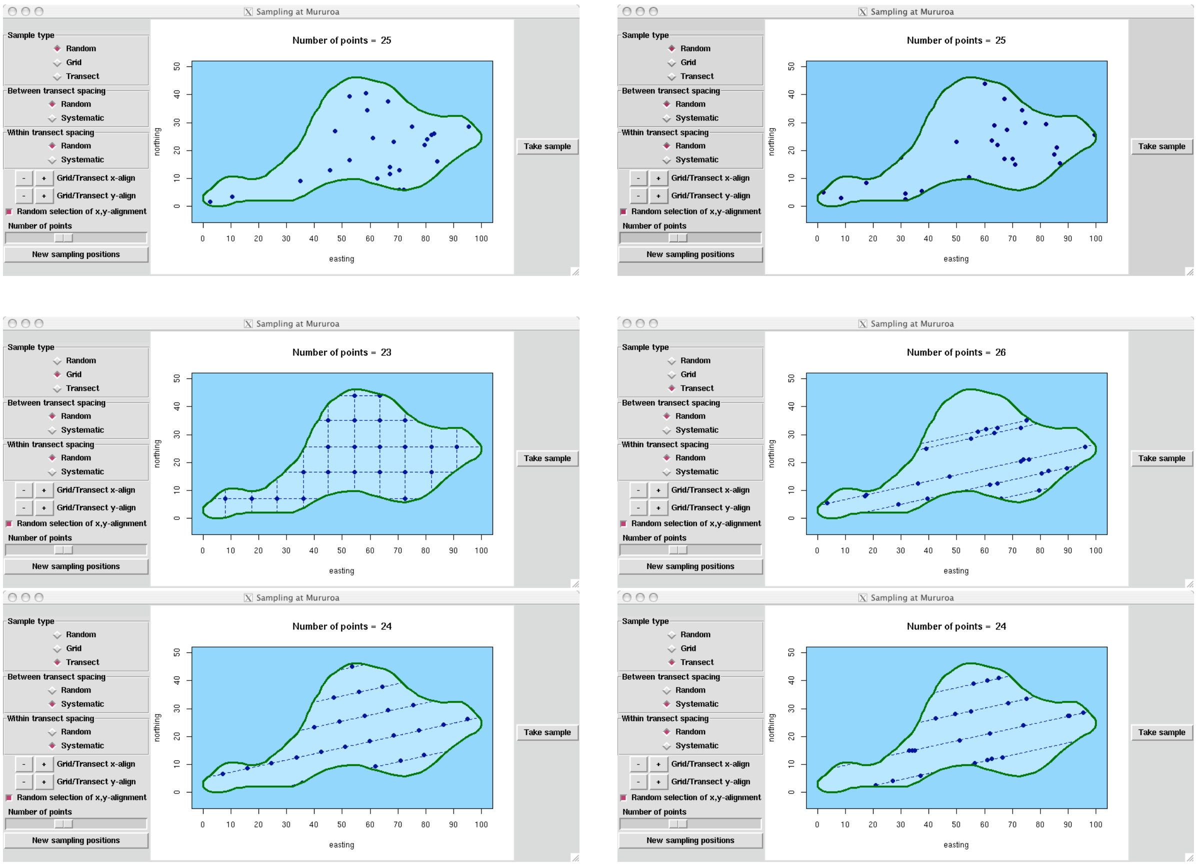
Finally, it is helpful to suppress the simulated surface and show only the measured values at the sampling locations. This, of course, is the usual starting point for a spatial analysis and leads, for example, to the construction of a sample semivariogram, as shown in the bottom panel of Figure 2. Repeated simulation at this stage has a further useful role to play in demonstrating the considerable variation which can be exhibited by sample semivariograms. However, more generally, the preceding discussion of the full spatial process as a description of the way the data were generated promotes a deeper understanding of the underlying model and the meaning of any subsequent fitted parameters.

The `rp.geosim` function gives control of additional aspects of the simulations, such as the sample size and the value of the κ parameter in the Matérn covariance function, as well as angle and ratio parameters for anisotropy. The effects of altering all of these parameters are very instructive.

3. Mururoa atoll: A spatial sampling scenario

While the analysis of data collected by others in real applications is a very instructive experience, there are some statistical issues which are brought to the fore most effectively when students are confronted with issues of design. The **rpanel** function `rp.mururoa` is constructed around a real sampling context based on the effects of nuclear experiments conducted between 1966 and 1996 in the South Pacific, at the atolls of Mururoa and Fangataufa, (IAEA International Advisory Committee 1998). As part of the assessment of subsequent radiological conditions, both terrestrial and aquatic samples were collected and assayed for activities due to strontium-90, caesium-137, plutonium and tritium. The sampling scenario in `rp.mururoa` is based on water sampling by boat for tritium in the Mururoa atoll.

The principle of random sampling is an important one in many application areas and so this is a natural starting point for Mururoa. However, repeated random selection of sampling positions, as illustrated in the top two panels of Figure 3, immediately draws attention to the difficulty that spatial gaps of substantial size may well occur. A more systematic approach, by placing a regular grid over the region of interest, solves this problem and ensures good spatial



Sampling at Mururoa

Sample type: ● Random ○ Grid ○ Transect
Between transect spacing: ● Random ○ Systematic
Within transect spacing: ● Random ○ Systematic
Number of points = 25

Sampling at Mururoa

Sample type: ● Random ○ Grid ○ Transect
Between transect spacing: ● Random ○ Systematic
Within transect spacing: ● Random ○ Systematic
Number of points = 25

Sampling at Mururoa

Sample type: ○ Random ● Grid ○ Transect
Between transect spacing: ● Random ○ Systematic
Within transect spacing: ● Random ○ Systematic
Number of points = 23

Sampling at Mururoa

Sample type: ○ Random ○ Grid ● Transect
Between transect spacing: ● Random ○ Systematic
Within transect spacing: ● Random ○ Systematic
Number of points = 26

Sampling at Mururoa

Sample type: ○ Random ○ Grid ● Transect
Between transect spacing: ○ Random ● Systematic
Within transect spacing: ○ Random ● Systematic
Number of points = 24

Sampling at Mururoa

Sample type: ○ Random ○ Grid ● Transect
Between transect spacing: ○ Random ● Systematic
Within transect spacing: ● Random ○ Systematic
Number of points = 24

Figure 3: The top two panels show sampling positions for Mururoa Atoll which have been selected randomly. The left hand middle panel illustrates the use of a regular grid, with a randomly selected starting point. The remaining three panels show line transects with different choices of random and systematic spacing between and within transects.

coverage. The '`Grid/Transect x-align`' and '`Grid/Transect y-align`' doublebuttons allow the grid position to be shifted and raises the issue of how this should be chosen. An element of random sampling can be adopted by selecting the horizontal and vertical location of the grid lines randomly, as illustrated in the left hand middle panel of Figure 3. A further approach is to use line transects, which are well adapted to movement of a boat along a fixed direction and with the spacing of sampling locations between and/or within transects selected randomly. A systematic rather than random strategy for both these spacings produces a regular grid which is no longer oriented in a north-south and east-west direction. Again, the starting position for the grid can be selected randomly. All of these options are illustrated in the remaining three panels of Figure 3. The direction of the transects has been selected to match the broad orientation of the water body within the atoll. This direction could, in principle, itself be selected randomly, although practical considerations in maneouvering the boat may militate against that. Finally, for all strategies, the number of sampling points also has to be specified.

The need to make a decision on sampling strategy requires students to think carefully about the consequences for later analysis and this is a very valuable activity. (The reader may wish to consider at this stage which strategy he or she would select.) When a decision has

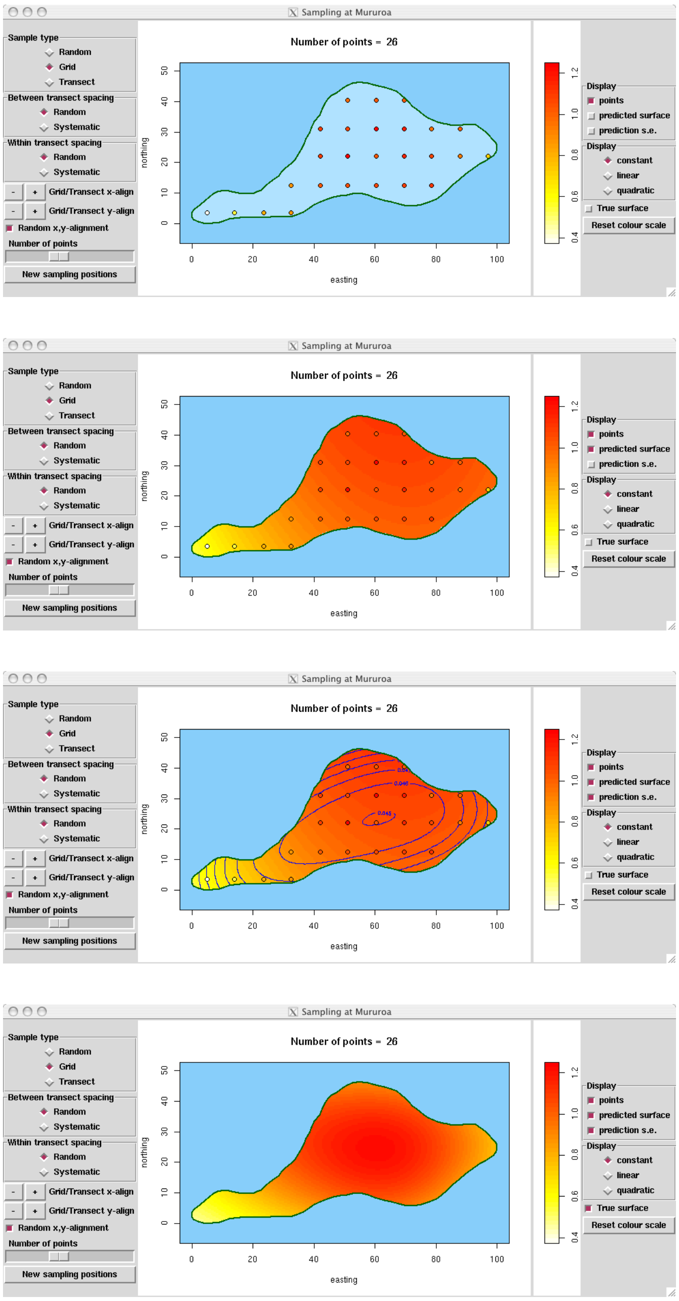


Figure 4: The top panel shows sampled points for Mururoa Atoll, with the observed values coded by color. The two middle panels display a predicted surface, with standard error information superimposed as contours, under an assumption of constant trend. The bottom panel shows the true surface (trend plus spatial process) from which the data were simulated.

been made, the '`Take sample`' button on the right hand size of the panel generates simulated data, on a scale of kBq/m^3 for subsequent analysis. (The structures used to simulate the data are discussed in Appendix A below.) The controls on the left hand side of the panel are also disabled, representing the fact that in practice a single decision must be made and its consequences lived with.

A new set of controls is now available on the right hand side of the panel. These call `geoR` functions such as `variog`, `variofit`, `krige.control` and `krige.conv` to perform some simple analysis based on kriging, illustrated in Figure 4. The measured values at the sampling positions, a predicted surface and its standard errors can all be viewed, and choices can be made on whether a constant, linear or quadratic trend function should be fitted. It is instructive to view the differences in predicted surfaces which these choices produce, with a quadratic trend prone to more extreme prediction at the edges of the spatial region. A checkbox is also available to display the structure from which the data were simulated, namely the trend function plus spatial process. Within the context of a teaching exercise, the ability to compare predictions with the underlying truth is very helpful.

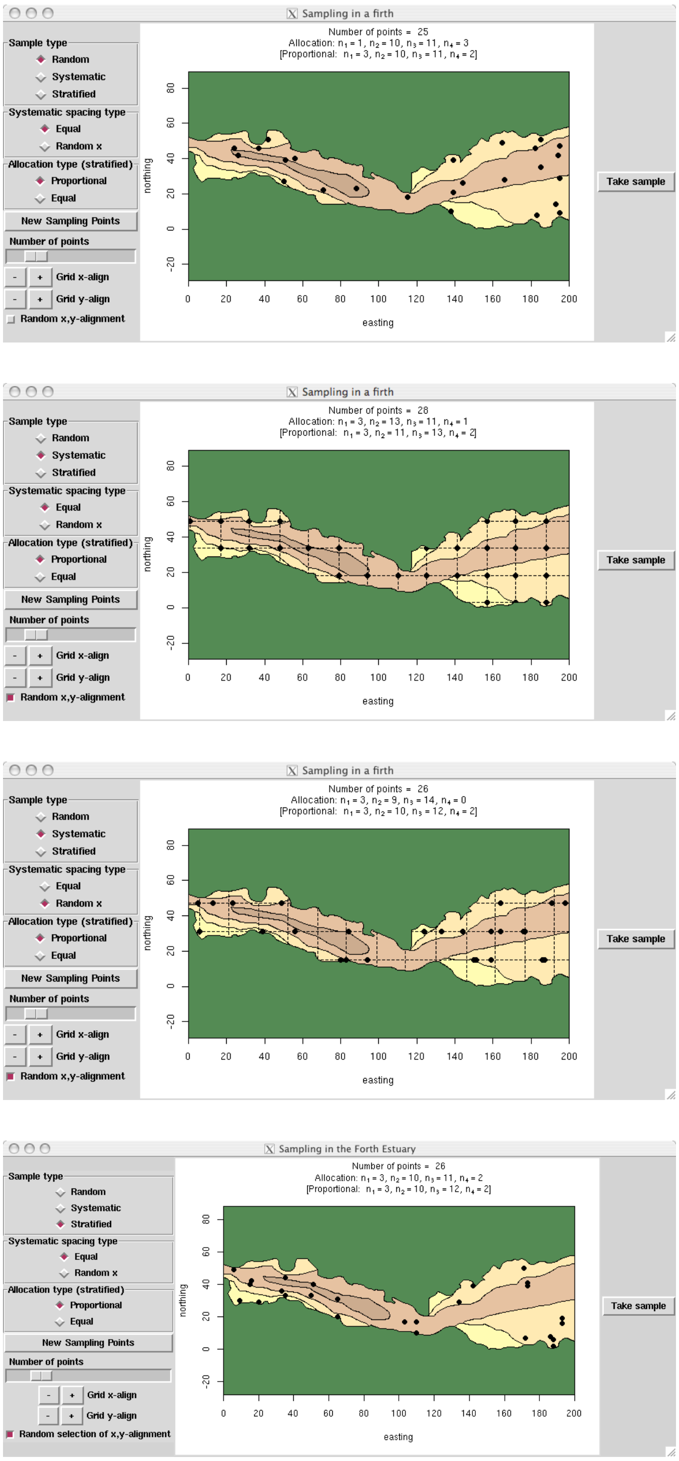
On some occasions, kriging may run into computational difficulties. The appendix shows how the simulated data may be exported to a file for examination outside the **rpanel** application. This allows investigation of whether large spatial gaps, which may occur with random sampling, are implicated in the computational problem.

4. The sea bed of a firth: Stratified spatial sampling

A second sampling exercise is available in the **rpanel** function `rp.firth`. This scenario is based on the mapping of radioactivity and the calculation of a radionuclide inventory within a water body. (A 'firth' is a Scottish term for a long, narrow indentation of the sea coast at the mouth of a river.) Interest lies in nuclides which, on release into a water body, attach (adsorb) to sediment in a manner which depends on the sediment particle size. Cobalt-60 and caesium-137 are examples of nuclides which exhibit this behaviour. In this sampling scenario, the map of sediment type is used to define regions of different particle size from which the sediment samples will be collected by grabs from a boat.

The additional issue to be faced in this scenario is the presence of strata, as the different types of material on the sea bed may affect the mean values of the measurements taken. One option is simply to select sampling positions randomly and hope that this will automatically cover the strata in a suitable manner. However, as illustrated earlier, this approach suffers from the possibility of large spatial gaps. A systematic approach using a regular grid, with random selection of the grid starting position, is again available. A further variation is to retain the grid as the basic structure but to allow the horizontal positions of the sampling points along the grid lines to be selected randomly. Finally there is an option to focus on the strata more directly, by carrying out stratified random sampling to ensure that the number of sampled points in each stratum matches the proportion of the spatial region which each stratum represents. Even where stratified sampling is not used, the numbers of sampling positions which fall into each stratum are printed at the top of the panel plot, so that this aspect of the sampling strategy can be monitored. A number of these different strategies are illustrated in Figure 5.

Once a decision has been made, and the '`Take sample`' button pressed, data are simulated and presented for analysis. The observed data and a simple spatial prediction are displayed in the top two panels of Figure 6, on a scale of Bq/kg. The additional issue of strata can be incorporated into the prediction process by fitting level shifts, using the `likfit` function in **geoR**. This assumes a model which has a trend function, a single random spatial process plus



Sampling in a firth

Number of points = 25
Allocation: $n_1 = 1$, $n_2 = 10$, $n_3 = 11$, $n_4 = 3$
[Proportional: $n_1 = 3$, $n_2 = 10$, $n_3 = 11$, $n_4 = 2$]

Sample type: ● Random ○ Systematic ○ Stratified
Systematic spacing type: ● Equal ○ Random x
Allocation type (stratified): ● Proportional ○ Equal
□ Random x,y-alignment

Sampling in a firth

Number of points = 28
Allocation: $n_1 = 3$, $n_2 = 13$, $n_3 = 11$, $n_4 = 1$
[Proportional: $n_1 = 3$, $n_2 = 11$, $n_3 = 13$, $n_4 = 2$]

Sample type: ○ Random ● Systematic ○ Stratified
Systematic spacing type: ● Equal ○ Random x
Allocation type (stratified): ● Proportional ○ Equal
☑ Random x,y-alignment

Sampling in a firth

Number of points = 26
Allocation: $n_1 = 3$, $n_2 = 9$, $n_3 = 14$, $n_4 = 0$
[Proportional: $n_1 = 3$, $n_2 = 10$, $n_3 = 12$, $n_4 = 2$]

Sample type: ○ Random ● Systematic ○ Stratified
Systematic spacing type: ○ Equal ● Random x
Allocation type (stratified): ● Proportional ○ Equal
☑ Random x,y-alignment

Sampling in the Forth Estuary

Number of points = 26
Allocation: $n_1 = 3$, $n_2 = 10$, $n_3 = 11$, $n_4 = 2$
[Proportional: $n_1 = 3$, $n_2 = 10$, $n_3 = 12$, $n_4 = 2$]

Sample type: ○ Random ○ Systematic ● Stratified
Systematic spacing type: ● Equal ○ Random x
Allocation type (stratified): ● Proportional ○ Equal
☑ Random selection of x,y-alignment

Figure 5: The top panel shows sampling positions for the firth which have been selected randomly. The second panel illustrates the use of a regular grid, with a randomly selected starting point. The third panel illustrates the additional step of assigning horizontal positions randomly within a grid structure. The bottom panel shows locations selected by stratified sampling.

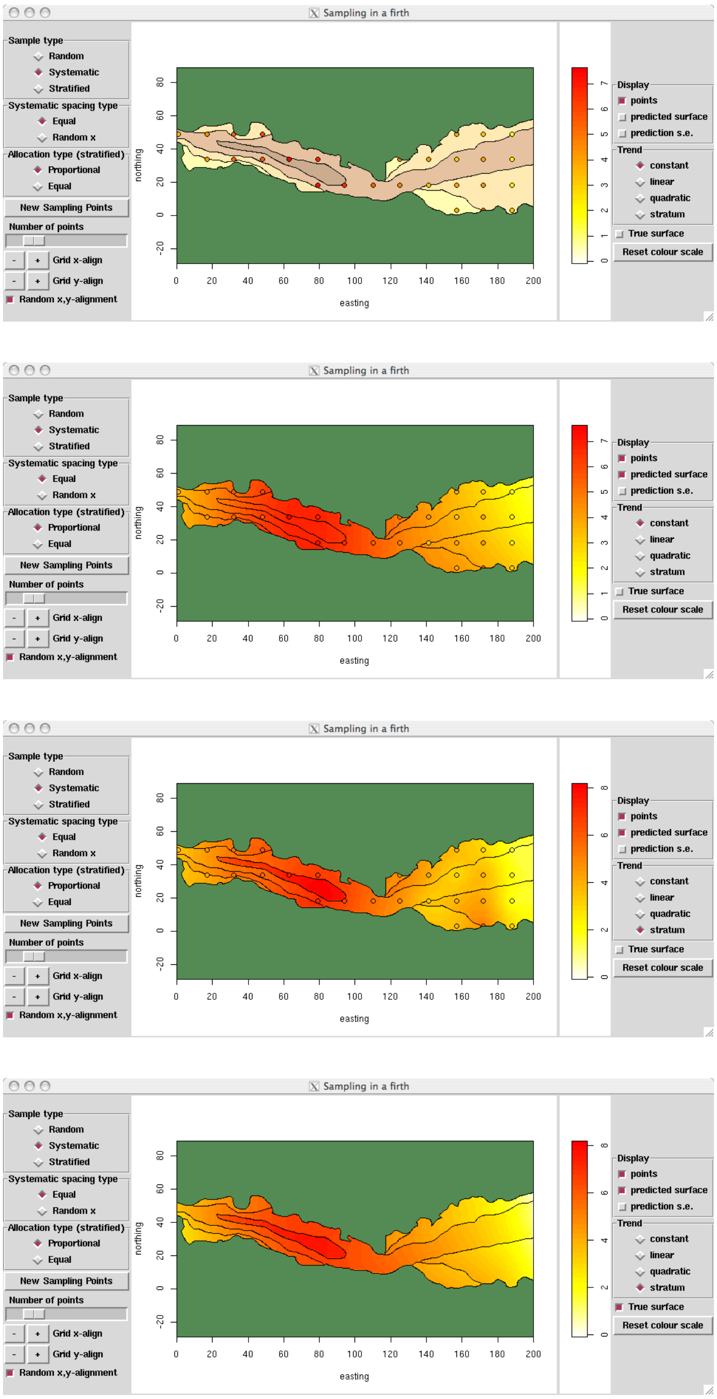


Figure 6: The top panel shows sampled points for the firth, with the observed values coded by color. The second panel displays a predicted surface under an assumption of constant trend while the third shows a prediction which incorporates stratum effects. The bottom panel shows the true surface (trend plus stratum effects plus spatial process) from which the data were simulated.

a mean shift for each stratum, as well as a nugget effect. This model structure may be beyond the level of sophistication which is appropriate for students who are meeting spatial data for the first time. However, even if the technicalities are beyond the grasp of the students, the underlying concepts can be demonstrated clearly in a graphical manner, through the changes in levels across the strata boundaries in the third panel of Figure 6. Indeed, on this occasion the data were simulated from a model which did indeed have strata effects, as shown in the bottom panel of Figure 6 where the true data structure is displayed.

5. Discussion

The aim of the tools described in this paper is to enable students and researchers to explore some of the issues associated with spatial data and models in an interactive manner, to promote intuitive understanding of the concepts involved. In particular, the sampling scenarios provide contexts in which issues of design have to be considered and this helpfully extends the range of activities to which students can be exposed.

Graphical controls for simulation can clearly be implemented to good effect in a much wider range of application areas. Even within the context of spatial data, it would be straightforward in principle to apply this to spatial point patterns, for example using the **splancs** (Rowlingson and Diggle 2008) or **spatstat** (Baddeley and Turner 2005) packages.

Teachers may wish to build on the sampling scenarios tools by incorporating other practical aspects of data collection. For example, information about likely forms of spatial correlation and nugget effects might be added to GUIde sampling strategies. Similarly, decisions on sampling strategies have to be taken within the context of budget limits so the costs of sampling could be given a simple quantification as a fixed overhead plus a component proportional to sample size. The allocation of a fixed budget, or the need to negotiate beyond this, provide further constraints which students need to consider.

Acknowledgments

The financial support of Ewan Crawford through the Higher Education Academy network for Mathematics, Statistics and OR, supported jointly by the university funding councils of the UK, is gratefully acknowledged. The advice of Richard Bowman on hiding the contents of functions in R is also greatly appreciated.

References

Adler D, Murdoch D (2010). ***rgl**: 3D Visualization Device System (OpenGL).* R package version 0.91, URL http://CRAN.R-project.org/package=rgl.

Baddeley A, Turner R (2005). "**spatstat**: An R Package for Analyzing Spatial Point Patterns." *Journal of Statistical Software*, **12**(6), 1–42. URL http://www.jstatsoft.org/v12/i06/.

Barnett V (2006). *Environmental Statistics: Methods and Applications.* John Wiley & Sons, London.

Bivand RS, Pebesma EJ, Gómez-Rubio V (2008). *Applied Spatial Data Analysis with R.* Springer-Verlag, New York.

Bowman A, Crawford E, Alexander G, Bowman RW (2007). "**rpanel**: Simple Interactive Controls for R Functions Using the **tcltk** Package." *Journal of Statistical Software*, **17**(9), 1–18. URL http://www.jstatsoft.org/v17/i09/.

Cressie NAC (1993). *Statistics for Spatial Data.* John Wiley & Sons, New York.

Dalgaard P (2001). "The R-Tcl/Tk Interface." In K Hornik, F Leisch (eds.), *Proceedings of the 2nd International Workshop on Distributed Statistical Computing, March 15–17, 2001, Technische Universität Wien, Vienna, Austria.*

Diggle PJ, Ribiero PJ (2007). *Model-Based Geostatistics.* Springer-Verlag, New York.

Handcock MS, Wallis JR (1994). "An Approach to Statistical Spatial-Temporal Modeling of Meterological Fields." *Journal of the American Statistical Association*, **89**, 368–378.

Helbig M, Urbanek S, Fellows I (2010). ***JGR** - Java GUI for R.* R package version 1.7-2, URL http://CRAN.R-project.org/package=JGR.

IAEA International Advisory Committee (1998). *The Radiological Situation at the Atolls of Mururoa and Fangataufa: Main Report.* International Atomic Energy Agency, Vienna.

Lawrence M, Temple Lang D (2010). "**RGtk2**: A Graphical User Interface Toolkit for R." *Journal of Statistical Software.* Forthcoming.

Paciorek CJ (2007). "Bayesian Smoothing with Gaussian Processes Using Fourier Basis Functions in the **spectralGP** Package." *Journal of Statistical Software*, **19**(2), 1–38. ISSN 1548-7660. URL http://www.jstatsoft.org/v19/i02/.

Pebesma EJ, Bivand RS (2005). "Classes and Methods for Spatial Data in R." *R News*, **5**(2), 9–13. URL http://CRAN.R-project.org/doc/Rnews/.

Piegorsch WW, Bailer AJ (2005). *Analyzing Environmental Data.* John Wiley & Sons, New York.

R Development Core Team (2010). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Ribeiro PJ, Diggle PJ (2001). "**geoR**: A Package for Geostatistical Analysis." *R News*, **1**(2), 14–18. URL http://CRAN.R-project.org/doc/Rnews/.

Rowlingson B, Diggle P (2008). ***splancs**: Spatial and Space-Time Point Pattern Analysis.* R package version 2.01-27, URL http://CRAN.R-project.org/package=splancs.

Rue H, Follestad T (2002). "**GMRFLib**: A C-Library for Fast and Exact Simulation of Gaussian Markov Random Fields." *Statistics Report 1*, Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway.

Rue H, Held L (2005). *Gaussian Markov Random Fields.* Chapman & Hall/CRC, Florida.

Schlather M (2001). "Simulation and Analysis of Random Fields." *R News*, **1**(2), 18–20. URL http://CRAN.R-project.org/doc/Rnews/.

Schlather M (2009). ***RandomFields**: Simulation and Analysis of Random Fields.* R package version 1.3.41, URL http://CRAN.R-project.org/package=RandomFields.

Tierney L (2005). ***tkrplot**: Simple Mechanism for Placing R Graphics in a Tk Widget.* R package version 0.0-19, URL http://CRAN.R-project.org/package=tkrplot.

Urbanek S, Theus M (2003). "**iPlots** – High Interaction Graphics for R." In K Hornik, F Leisch, A Zeileis (eds.), *Proceedings of the 3rd International Workshop on Distributed Statistical Computing, 2003, Technische Universität Wien, Vienna, Austria.* Technische Universität Wien, Vienna, Austria. URL http://www.ci.tuwien.ac.at/Conferences/DSC-2003/Proceedings/.

Verzani J (2007). "An Introduction to **gWidgets**." *R News*, **7**(3), 26–33. URL http://CRAN.R-project.org/doc/Rnews/.

Webster R, Oliver MA (2001). *Geostatistics for Environmental Scientists.* John Wiley & Sons, Chichester.

A. Software design

The spatial functions described above are all constructed with the graphical control tools provided by the **rpanel** package. The aims of **rpanel** are to make it as easy as possible for R users to add GUI controls, using single function calls for each, and in addition to provide a range of useful applications which make use of these tools. The basic mechanism of the package is described in Bowman *et al.* (2007). It is built on the **tcltk** package, with 'behind the scenes' management of its communication mechanisms, and requires the user to supply an 'action' function which will be executed when the state of a particular control widget is altered. The complexity of an **rpanel** application is therefore determined by the sophistication of the user's code. Integration of R plots within the control panel, built on the **tkrplot** package of Tierney (2005), is a particularly helpful feature which avoids the need for multiple windows. This feature is used repeatedly in the spatial applications described above, as the Figures illustrate.

A recent feature of **rpanel** is the extension of facilities for arranging controls within the panel. The spatial applications use these features extensively, for example allowing the 'Take sample' button to be replaced by a larger set of controls which activate analysis of the sampled data. Earlier modes of operation allowed the positioning of control items within a panel through simple top/bottom/left/right orientation, or by the specification of pixel positions. These modes make use of the 'pack' and 'place' modes within Tcl/Tk. As an alternative, more flexible approach, a grid system can now be specified for the panel as a whole, using the rp.grid function, which in turn calls the 'grid' mode within Tcl/Tk. For example, the basic layout of the Mururoa panel is constructed by identifying each grid component by name and row/column position in a grid layout.

```
rp.grid(panel, "controls1", row = 0, column = 0)
rp.grid(panel, "controls2", row = 0, column = 3)
rp.grid(panel, "plot1",     row = 0, column = 1)
rp.grid(panel, "plot1a",    row = 0, column = 2)
rp.grid(panel, "plot2",     row = 1, column = 1)
```

Individual controls can then be added by referring to a row/column position in a named grid element. For example, the following code adds a color chart and three controls to the grid on the right hand side of the Mururoa panel.

```
rp.tkrplot(panel, plot1a, mururoa.colour.chart,
  hscale = panel$hscale * 0.12, vscale = panel$hscale * 0.7,
  grid = "plot1a", row = 0 , column = 0)
rp.checkbox(panel, display.options,
  labels = c("points", "predicted surface", "prediction s.e."),
  initval = c(TRUE, FALSE, FALSE), action = mururoa.predict,
  title = "Display", grid = "controls2", row = 0, column = 0)
rp.radiogroup(panel, trend.setting, c("cte", "1st", "2nd"),
  labels = c("constant", "linear", "quadratic"), action = mururoa.predict,
  title = "Display", grid = "controls2", row = 1, column = 0)
rp.checkbox(panel, mururoa.true, mururoa.samp.redraw,
  title = "True surface", grid = "controls2", row = 2, column = 0)
rp.do(panel, mururoa.samp.redraw)
```

Trend	`trend.fn`	A function of a position vector of length two, to be evaluated over the grid defined by horizontal and vertical margins `seq(0, 100, by = 0.5)` and `seq(0, 50, by = 0.5)` for `rp.mururoa` and `0:200`, `0:60` for `rp.firth`.
Covariance parameters	`cov.pars`	A vector of length two giving the variance and range parameters of the spatial process, in the form required by **geoR**.
Nugget	`nugget`	The variance of the nugget effect.
Strata effects	`strat.effect`	The shifts in mean associated with each stratum. (This is required only for `rp.firth`.)

Table 1: The components used to define the structure from which data are simulated in the `rp.mururoa` and `rp.firth` functions.

This gives considerable flexibility in layout.

In the spatial scenarios embodied in `rp.mururoa` and `rp.firth`, sampling takes place over irregular regions. The simple approach which has been adopted to deal with this is to work with a fine grid of discrete spatial locations. This avoids the need for more complex strategies such as uniform sampling within an enclosing rectangle followed by exclusion of points falling outside the region of interest. It is also particularly helpful in the context of strata as each discrete location has an associated stratum identifier, which allows simple implementation of proportional sampling strategies.

With random sampling, the use of this fine grid allows a specified number of sampling locations to be sampled in a simple manner. With grid and transect sampling, particularly with random selection of the x and y alignment, this is much more difficult to achieve. With grid sampling, a simple solution is to set the grid spacing to $\sqrt{A/n}$, where A is the area of the region and n is the desired number of points. This does not guarantee exactly the specified number of points but the deviation is likely to be very small. With transect sampling, the same formula is used to determine the number of transects, which are then placed either systematically or randomly. The number of points on each transect is then set proportionately to the length of each transect within the spatial region of interest. The points are then placed either systematically or randomly. These strategies have the additional advantage of adjusting the numbers of grid lines or transects automatically, as the requested sample size increases.

The spatial sampling applications have embodied within them a particular data structure from which data are subsequently sampled. This structure is defined by the components shown in Table 1, together with their corresponding R expressions. These objects are loaded automatically by **rpanel**, along with other information used to draw region boundaries and other features. However, it is much more flexible to allow these parameters to be altered, so that teachers can construct the detailed nature of the simulated data, to highlight particular issues. This raises the interesting question of how a teacher can alter the required settings without these being accessible to students. The first requirement is a means of passing parameter values to `rp.mururoa` and `rp.firth` and this is available in a `parameter` argument which takes a list object with named components. The second requirement is a means of hiding this parameter setting from others. This can be achieved by starting a fresh R session and executing the following code which, in this case, simply sets the stratum effects of `rp.firth` to zero.

```
spatial.samp <- function(arg) rp.firth(parameters = arg)
hideargument <- function(f, arg) function() f(arg)
spatial.sampling <- hideargument(spatial.samp,
  list(strat.effect = rep(0, 4)))
rm(hideargument)
```

The workspace should then be saved in a file. Students can then load this workspace file and launch the customized software simply by making the function call `spatial.sampling()`. The required information is lifted from the appropriate environment without being directly accessible by students.

It is also possible to write the sampled data to a file for more extensive analysis by other means. This is achieved simply by specifying a filename in the `file` argument of `rp.mururoa` and `rp.firth`.

Affiliation:

Adrian W. Bowman
Department of Statistics
The University of Glasgow
Glasgow G12 8WW, United Kingdom
E-mail: adrian@stats.gla.ac.uk
URL: http://www.stats.gla.ac.uk/~adrian/

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 36, Issue 13 *Submitted:* 2008-12-15
October 2010 *Accepted:* 2010-02-03