



DSC 2003 Working Papers
(Draft Versions)

<http://www.ci.tuwien.ac.at/Conferences/DSC-2003/>

Opportunities and requirements for open source/zero cost statistical software in companies under permanent reorganization

Christian Ritter

Monnet Centre International Laboratory
Avenue Jean Monnet 1, B-1348 Louvain-la-Neuve
Belgium

Abstract

In companies under permanent reorganization, stable software environments are hard or even impossible to maintain. Parts of the company break off into joint ventures and licences need to be split, mergers occur implying mixed software cultures, and the market moves towards Microsoft-like installation and licencing: Centrally managed individual licenses instead of floating licenses for a defined number of concurrent users.

Since average usage time (usage time per employee) of statistical software is very small compared to office tools (word processors, spreadsheets, etc.), "global" availability of commercial statistical software is therefore not economical. This means that even in companies in which a common commercial statistical tool was available to everybody before, this will no longer be the case in the near future. Access to commercial statistical software will be limited to a pool of statisticians and para-statisticians. All others have to live with what their spreadsheet offers.

I believe that now is a unique opportunity for imposing standardized zero-cost/open-source software: This would allow global coverage, training, and communication even under changing environment. The R-language holds the potential to become such a solution. However, its lack of a GUI facilitating elementary tasks and its insufficient integration into office tools (Microsoft, Open Office, etc) make this currently impossible.

This abstract is essentially the content of the talk I gave at the DSC2003 session. In this paper, I shall describe my view of the status just after the conference.

1 Introduction

At present, there is a unique opportunity for a good open source/zero cost statistical software system to become a de facto standard environment in many companies and in particular those which are under permanent reorganization and in which statistical work is not centralized.

Why is this so? -There are several reasons:

1. Although rudimentary statistics are widely used, the market for statistical software has remained too limited to guarantee sufficient profits for mass market statistical software.

As a consequence, the several producers of statistical software (such as SAS, Insightful Corporation, and StatSoft) are more and more aiming at the market of data warehousing, data mining and regulated pharmaceutical research, areas in which there are pools of dedicated applications programmes and the willingness to pay high license fees.

2. Developers of mass market statistical software have to struggle hard to stay alive in a market which evolves towards microsoft-style licensing: One license per workstation. Since statistics are not used on a day-to-day basis (such as word processing, and spreadsheet manipulation) by the target public, this implies a high price-to-use ratio. In the era of cost-cutting, this often leads to the disappearance of mass market statistical software from the workplace.
3. Many common users have developed the habit of dealing with their statistical problems within their spreadsheets (Excel, Lotus, Calc, etc.). The problem with this development is that the currently available spreadsheets have quite limited statistical (exploratory and inferential) abilities and that developing such abilities seems of little importance to their vendors.

This environment favors a situation in which there are advanced and well developed tools for advanced users and rudimentary tools for common users, and nothing in-between.

Since open-source/zero cost software is zero cost, it fares well in cost-cutting but individual licensing times. Since it is open-source, "everybody" can contribute interfaces to commonly available tools, such as spreadsheets. In the area of statistics, the R-language is the most advanced open-source/zero cost tool. It therefore has the potential to become the unifying back-end to possibly a large number of interfaces which import its power into software tools (such as spreadsheets) available to common users.

Attempts for building interfaces to allow common users to use R will be most successful if they allow these users to stay within their familiar environment(s). The result will be most powerful, if the common user can gradually progress from elementary to advanced use without having to switch environments.

At DSC2003, several new and renewed interfaces between R and other computing environments were presented. In the remainder of this paper I shall give my first impressions on two of such interfaces, the (D)COM server for R coupled with an Excel Add-In (Baier, T. and E. Neuwirth), and SciViews, a full GUI in Windows for R and similar languages.

2 Embedding R into Excel using the R(D)COM interface presented by Baier T. and E. Neuwirth at DSC2003

A first version of a (D)COM server for R and an associated Excel Add-In had been presented by Baier and Neuwirth at DSC2001. This set of tools has now been enhanced and consists of a (D)COM server for R to allow COM clients (such as Excel) to use the functionality of R, a COM client for R to allow R to control other COM servers (such as office applications), a revised version of the Excel Add-In and an R library facilitating the use of the COM client. These tools are documented in Baier T. (DSC2003) and Baier T. and E. Neuwirth (DSC2003).

2.1 Using the R(D)COM server via the Excel Add-In

The R(D)COM/Excel interface allows essentially to transfer elementary data (arrays) between Excel and an instance of R and to execute scripts of R commands. This enables an Excel developer to build custom applications within Excel which use the capabilities of R. For example, one could develop a spreadsheet for a routine statistical analysis of a split plot structure, transfer the data to R (in array pieces) transfer the results back to Excel (in array pieces) and to assemble the returned data into an spreadsheet like results table. Technically, this would most likely be organized as a VBA macro linked to a “Calculate” button.

The execution of R code and the passing of elementary arguments can be wrapped into user defined worksheet functions and can therefore be included in the automatic recalculation loop of Excel. This in turn allows extending the limited number of statistical functions by any function in R which can be executed by a simple call with simple arguments. For example, calculations of P-values for tests which are not available in excel, generation of random variables, etc. can be arranged in this way. They then follow the spreadsheet paradigm: Add data to the table and the summaries will be updated automatically. More complex constructions are possible but some rules on dependencies have to be respected.

2.2 Using the R(D)COM client

The R(D)COM client allows to access other COM interfaces from R. Spreadsheets can be opened, used for data entry and display (including graphs), closed. Information can be drawn from spreadsheets, processed in R, and displayed in other environments, such as presentation tools or text documents.

So far, I can see clearly, how spreadsheet and office developers can benefit from these interfaces. Using the COM interface of R, they can enhance the abilities of office components by the functions of R and they can develop customized access to R and from R to other office components.

However, I do not yet see how the common user by him/herself can best exploit the R(D)COM/Excel interface. Except for very basic functions, writing the R-code to be executed from Excel requires knowing R. Debugging is even more cumbersome than via the common command interface. This can be partially alleviated by using the internal connection mode, but how to do this is not yet sufficiently obvious to common users.

More complex interaction with R (using RProc, RApply, REval where results of REval depend on results of RProc for example) requires careful handling of dependencies via dummy arguments which again is not for the uninitiated.

So, the R(D)COM interface is a very welcome tool for the office developer, but not yet a gentle entry to R for the common user. It will allow Excel developers to put R into the tank of Excel analyses while maintaining a familiar environment for the user. However, the common users will not be able to move to “more R” by themselves by means of this interface.

3 The SciViews GUI presented by Philippe Grosjean at DSC2003

SciViews takes a radically different approach, although some of the internals are quite similar to the R(D)COM. In SciViews a new independent GUI is presented which is made to look similar to common Office tools. The arrangements and behaviors of common elements of a spreadsheet or a web browser (side bar, status bar, formula bar, help segments) are reproduced to facilitate the initial orientation of common users.

The connection to R is accomplished via a generic plug-in protocol which also allows to link in other applications for which compatible plugs have been written. On the R plug level, communication is similar to the (D)COM interface, although SciViews requires a set of extensions (such as asynchrone operation) which are the R(D)COM developers judge less essential for their environment.

SciViews tries to make full use of Windows specific elements. For example, graphs initially created within the R graphics window are not displayed independently, but become child windows of the GUI.

On the side of the GUI implementation, SciViews implements several linked elements: A command window, a script editor, a function and command list linked to help documentation, an analysis note pad, and (under construction) an object browser.

The familiar office look of SciViews helps new users to enter the program but the structure leads them progressively to use the (enhanced) command tool. However, it is still at a development stage and will require a substantial amount of additional work to become fully mature. Moreover, by design, SciViews is a stand-alone tool and therefore not integrated into common office applications.

4 Present and Future Challenges

In trying to work with tools like SciViews and R(D)COM, I find many unresolved challenges such as:

1. Which role and power should common users have and in which environment should they start working when using R? In their familiar spreadsheet or in a comfortable specialized GUI?
2. How should the initial usage experience kept simple, reassuring and fully office integrated while providing layers of increased power and sophistication. This is important when common users need to solicit help from statistical experts.

It is best if this help can be given in an extension of the familiar environment instead of requiring a move to a different environment.

3. In the case of R(D)COM/Excel, the limitation of transferring (crude) arrays without column and row names stands in contradiction with the most common R object, the data frame. Yet, it is not obvious how to label such objects in Excel and how to safeguard the transfer.
4. Again, in the case of R(D)COM/Excel, the lack of the command prompt and return message makes it hard to begin working with R, however, such tool would make little sense for R-evaluations within the event loop.
5. For SciViews the main challenge is to reach maturity in a very ambitious setting and then to reach full office integration.
6. In general, so far none of the presented approaches explicitly supports analysis building. In analysis building, data extraction, exploration, modeling, and interpretation are archived in a reproducible, documented, portable, auditable, and archivable fashion. This is the way anyone should carry out data analysis and software should support this approach. Both interfaces, R(D)COM and SciViews implement parts of the required elements. R(D)COM/Excel allows for example to construct dynamic spreadsheets linking the data to the analysis. However, it is not convenient to include the interpretation in such a spreadsheet. SciViews, on the other hand, includes a note pad in which one can paste results of analyses and graphs. However, the data processing steps are not automatically archived with the results. An approach which better supports creating reproducible, auditable, and archivable analysis is the Sweave project (F. Leisch, DSC2003). However, so far, its structure is not suitable for the common user and there are several limitation which make its use for real world projects cumbersome (inclusion of annotated functions, parsing of sourced files). Another interesting element in this context is the notion of R-projects created similarly to R-packages. However, also this approach is still in a immature state. Much could be gained by a tool suite allowing expert and common users to create reproducible, documented, portable, auditable, and archivable analyses.

In summary, I see potential but not yet maturity. R(D)COM and SciViews show two different ways of making R available to a larger community of users but both are still somewhat limited. Other attempts exist and their respective merits and shortcomings will have to be studied.

The danger at present is that too many different and competing attempts are started and carried on out of curiosity and individualism. Some respectful coordination would be very helpful to avoid waste in time and effort. It is important to remember that now is a unique opportunity to unify the statistical end of commonly available software. It may not last.