



Journal of Statistical Software

June 2012, Volume 49, Issue 1.

<http://www.jstatsoft.org/>

Graphical User Interfaces for R

Pedro M. Valero-Mora
Universitat de València

Rubén D. Ledesma
Universidad Nacional de Mar del Plata

Abstract

Since R was first launched, it has managed to gain the support of an ever-increasing percentage of academic and professional statisticians. However, the spread of its use among novice and occasional users of statistics have not progressed at the same pace, which can be attributed partially to the lack of a graphical user interface (GUI). Nevertheless, this situation has changed in the last years and there is currently several projects that have added GUIs to R. This article discusses briefly the history of GUIs for data analysis and then introduces the papers submitted to an special issue of the *Journal of Statistical Software* on GUIs for R.

Keywords: GUI, statistical software, R.

1. Introduction

Nowadays, graphical user interfaces (GUIs) are the most common way of interacting with a computer or other electronic devices. Whereas they are arguably not the best way of performing every conceivable job, its dominance has reached the principal operative systems, application domains, and tasks. Measured it in purely statistical terms, the triumph of this style of interaction is unquestionable and it seems clear that it will remain very popular in the foreseeable future.

However, regarding the discipline of statistics, this supremacy does not seem so dramatically clear, as many of the people working in this field seem to favor that using a typed language via a command line interface (CLI) is a much more productive, accurate and reproducible way of performing their tasks than a GUI. Actually, we are positive that they are right, because writing up commands can be undoubtedly much more productive than using a GUI, as far as the learning cost is not included in the bill. Thus, although getting the knowledge to manage a command-based language is within reason if you are going to use it very often, the novice and occasional users of statistical software may not ever see sufficient returns to justify the

high entry effort needed for learning it.

Users of statistics often do not own a degree on this subject. These users are usually experts in their own fields that have mastered a number of well-established statistical techniques, which they apply rather routinely to datasets with well-known structure. Although they may require occasionally more sophisticated tools for their analysis, in general they act in well delimited scenarios with few unexpected surprises. These people are quite different of seasoned statisticians applying groundbreaking techniques to novel problems, hence demanding flexibility, computational power, last minute developments and so forth. Therefore, it should not come with a big shock to find out that the software of choice of one type of users is hardly the apple of the eye of the other.

The previously described interaction among tasks and expertise in statistics is very relevant for understanding the value of one type of interface over the other. The familiarity with software environments and computer tasks is other relevant factor. Thus, those acquainted with computer programming are very at ease with typing commands, although they may also appreciate the support of a sophisticated integrated development environment (IDE) that may be based partially in a GUI. Actually, they may skew so much to CLIs that they often prefer to solve mundane tasks using their programming skills rather than learning the usual office programs that the rest of world usually is more inclined to use.

As pointed by [De Leeuw \(2009\)](#), the GUI revolution reached the statistical software more or less at the arrival of the Macintosh computers. Thus, about 1985, after the long dominance of the big three statistical packages – BMDP, SPSS, and SAS – a program with a truly innovative GUI, **DataDesk** ([Velleman 1997](#)), was released. This program was shortly followed by **JMP** ([SAS Institute Inc. 2012](#)) in 1989 as well as others (*Systat 5* was quite popular, and **Statview** and **SuperANOVA** had also very intuitive GUIs). However, after the big packages released version for microcomputers there was a quick convergence to them and the rest of the statistical software became rather marginal.

From the point of GUIs, this change meant a leap backwards, as these big packages dragged their origin in non-graphical computers-sometimes, you could not help feeling an eerie impression watching those character-based plots within a Macintosh. Fortunately, the succeeding GUIs for these packages improved somewhat and they approached to the standards of usability of other computer applications. Nevertheless, they still consisted mainly of a layer that hid the program to the eyes of the users interested in carrying out only basic analysis without providing added value to it.

Non-commercial statistical software, on the other hand, did not ever have much of a GUI. So, the most influential representative, the S language ([Becker, Chambers, and Wilks 1988](#)), was, despite its superb features for graphics, a typed language. It was not until S-PLUS, the commercial version of the language, turned up that a GUI was developed for S. However, S-PLUS success was very limited because R ([Ihaka and Gentleman 1996](#); [R Development Core Team 2012](#)) quickly occupied its place as the environment of choice for using S, and, as R was utilized mainly on a CLI, this replacement meant again a step back in terms of GUIs for statistics.

In another branch of non-commercial software, **Lisp-Stat** ([Tierney 1990](#)), a statistical language partially inspired by S and by the interest on applications of interactive graphics to data analysis, appeared at the beginning of the 90s and enjoyed popularity for a few years. **Lisp-Stat** was remarkable because it had native support of GUI programming so it was rather usual

that researchers provided a more or less sophisticated GUI accompanying their programs. Also, the emphasis on visualizing data using interactive graphics led to explore GUI issues related with them. However, **Lisp-Stat** active development stalled following the advent of R and despite some recent claims showing a renovated interest in using **Lisp** for statistical programming (Ihaka and Temple Lang 2008), its return does not look likely.

Along the years, we see that statistical software has addressed the needs of the intensive users or of the non-intensive, or of both at the same time, with more or less success. With respect to R, its main aim has always been the support of the tasks of the most advanced users. However, along the years, there has been attempts to extend this support to other types of users. In particular, although it lacks native support for GUI functions, the capability of linking it to other languages soon permitted the release of the R Commander (Fox 2005), one GUI interface based on Tcl/Tk, which subsequently was followed by others, making the list of options grow up considerably. Given the potential impact of these GUIs we reckoned that an overview of them in a special issue of the Journal of Statistical Software (JSS) would be of interest.

As a result of our call, we received a number of papers, which, after the selection process, left us with the ten papers included in this special issue. These papers can be classified in four approximate categories, namely, papers discussing a specialized GUI for a relatively small or specialized package, general interfaces to R trying to cover all the basic functionality in statistical packages, toolkits or other tools supporting the programming of GUIs in R and, finally, papers dealing about how to build innovative GUIs for R that would provide better support to data analysis and exploration. We will provide a short description of the papers in the different categories in the following paragraphs and then we will finalize this document with some concluding remarks.

1.1. Specialized GUIs

This category includes papers written by Cheng and Peng (2012), Snellenburg, Laptinok, Seger, Mullen, and van Stokkum (2012), Wallace, Dahabreh, Trikalinos, Lau, Trow, and Schmid (2012), and Huang, Cook, and Wickham (2012). These papers coincide in describing specialized graphical user interfaces for specific task domains.

The article by Cheng and Peng (2012) describes a user interface programmed with Tcl/Tk for a package written in R for the analysis of reliability of products. The assumption of the authors of this paper is that users of their package will not be interested in using R at full steam but they will be content with the functions in this package. Therefore, a GUI interface will provide to these users what they require while saving them to learn the syntax. This is an approach that could be used in many packages in order to improve its friendliness to occasional users.

The paper by Snellenburg *et al.* (2012) describes a rather sophisticated user interface written in Java, which permits the data exploration, visual modeling and interactive inspection of analysis results. This interface, called **Glutaran**, is therefore not only a convenient way of specifying the input options in the **TIMP** package developed previously by the authors, but also facilitates the exploration of the results of the analysis. Actually, **Glutaran** is a GUI that tries to provide an added value to R and not only make redundant the task of typing commands.

The article by Wallace *et al.* (2012) is still another interface for a specialized domain of

statistical analysis. In this case, the language chosen for writing the interface is **Python** and the application is meta-analysis. The authors explain that providing user-friendly software is very important for facilitating that researchers pay attention to new methods. Thus, they believe that the reason for underusing certain methodologies often stems from the lack of convenience for occasional users of learning how to run it. Therefore, adding a GUI to a specialized program, may boost the popularization of advanced methods that would remain underused otherwise.

Finally, the article by [Huang *et al.* \(2012\)](#) describes a GUI for the **tourr** package. The tour is a method for visualizing multivariate data that has been available in the package **ggobi** for many years and that had been translated to a package, named **tourr**, written purely in R ([Wickham, Cook, Hofmann, and Buja 2011](#)). However, this package had only a command line interface to the tour, which in a package focused on data visualization was regarded as odd. Therefore, the authors developed a GUI for controlling the tour. Dynamic graphics is a domain where interactivity plays such a crucial role that importance of a GUI is especially evident.

1.2. General interfaces for R

The second group of papers contains three papers that describe general user interfaces for R in two cases, and an example of how to extend the probably most popular of such systems, the R Commander, in the other. The papers in this category attempt to provide easy access to the most used functions for data analysis. So, in general, all the packages in this category allow importing a dataset from different formats and run some basic and intermediate analysis from menus or dialog boxes. This type of GUIs seems to be especially appropriate in the context of teaching introductory courses to statistics, and in fact, as this is not an infrequent objective, a quick search in the internet provides several projects with similar aim that could also have been included in this special issue. Given the relatively large size of the projects in this category, somebody trying to decide which package fits better to his needs may have many things to consider. As an example, the capability of running the same GUI in multiple platforms may be critical for some users but relatively unimportant for others. So, of the papers portrayed in this special issue, we see that **RKward** by [Rödiger, Friedrichsmeier, Kapat, and Michalke \(2012\)](#), which originated on the GNU/Linux platform, is the best supporting this environment but the version for Windows is also acceptable according to the same developers, whereas installing it in OS X is not so good. On the other hand, **Deducer** by [Fellows \(2012\)](#) runs in the three main OSs similarly. Finally, **Rcmdr** used in [Fox and Carvalho \(2012\)](#) runs in the three OSs but, as it uses Tcl/Tk, the appearance of the system does not look native in all of them. Of course, none of these aforementioned limitations are actually written in stone and they may be overcome by future progresses.

All these packages also offer methods for extending the core system. This way, users of these general interfaces may plug-in new methods to the core GUI so that the GUIs can grow up according to their needs. Good ways of supporting these inputs to the GUIs are critical and should be planned very carefully by the developers.

1.3. Toolkits for GUI programming in R

The GUIs previously described usually take advantage of programming toolkits. These toolkits are available as part of the utilities of general programming languages such as **Java**, **Python**,

Tcl/Tk, and so forth. Learning these toolkits can be somewhat tedious so facilities for interfacing them with R are a source of big convenience.

The paper by [Verzani \(2012\)](#) is an extension of the **gWidgets** package [Verzani \(2007\)](#) for developing interactive web applications. The package is called **gWidgetsWWW** and it provides a framework for the R user to create interactive web pages, using more modern and mature technologies than other technologies previously available.

Providing this type of facilities for the end-user is an important endeavor. The paper in this special issue is only one specific example of the kind of work that needs to be done.

[Austenfeld and Beyschlag \(2012\)](#) is a paper that describes a system embedded in the Eclipse. This GUI has the advantage of having immediate access to a number of components and toolkits that are immediately available for it. This way, it is possible to extend the GUI and link it to other scientific tools, extending the applicability of the system beyond its original purposes.

1.4. Innovative GUIs in R

The fourth category of articles in this special issue has only one paper, by [Unwin \(2012\)](#), in it. This paper is exceptional because it does not provide actual code but focuses instead on providing a rationale on how a good GUI for R (or for statistics in general) should be like. We believe this is an important approach, as present GUI efforts might be guided too much for what is currently easy to implement and not so much by what would be most interesting or exciting to have. Therefore, truly innovative GUIs for R might not be in the agenda of developers if their view is narrowed by the options more readily available to them. User interface designers in the industry rely often on quick non-functional prototypes because they know that putting too much effort at the beginning of a project will make very difficult to change it in the future, let alone to throw it away and restart the design from the scratch if new ideas arise along the process. Thinking the design carefully in abstract has the advantage that more options, and possibly more innovative, can be examined quickly so that more advanced designs can be envisioned. Therefore, if others continued the path undertaken by Unwin, either providing further discussions about what is desirable to have, or trying to implement the fresh ideas and see how they work in practice, we might step up to a new level of excellence in GUIs for R.

2. Discussion: Can we do better?

A cursory reading of the papers in this special issue will show to a reader acquainted with R a picture that may look very familiar: several groups of researchers working rather independently in partially complementary but somewhat overlapping projects. Therefore, the researcher or practitioner interested in adopting one of the projects will have many options at his avail. This is of course positive, as choice is always advantageous, but also have some negative side effects that must be also taken into account. We will discuss the pros and cons of this situation below.

Different groups working on similar projects guarantees that many potentially useful strategies will be explored. For instance, we have seen in this issue that Java, Python, Tcl/Tk, **gWidgets** have been used as toolkits for the GUIs hereby discussed. Each of them has advantages and disadvantages, and the results obtained may be critically constrained by them in ways that

are not easily predicted without actually trying them. Therefore, the energy used in this way may pay off in terms of the experience gained about the alternatives available.

However, the excess of choice may also have negative consequences. GUIs are often an important percentage of the total effort required for building an application and maintaining them can be also costly. Therefore, developing a sophisticated GUI may require a significant amount of resources. Many of the projects here described set forth implicitly or explicitly the expectation that a community will be created about them and then members of this community will contribute to continue and expand the project. However, if too many of these projects are being run simultaneously and any of them stands out as a clear winner, the total resources available might dissipate without powering sufficiently one in particular, preventing the advent of really comprehensive GUIs for R. Indeed, a clustering of the projects might have as consequence a big advancement with respect to what currently exists in this area.

The growth of the importance of R has been huge in the last years but it has impacted mainly the connoisseurs and not so much the masses, i.e., to the large number of users that only have basic to moderate needs in statistics. A more sophisticated GUI could change dramatically the situation for them as they might take advantage of all the power of R without having to pay the cost of learning a CLI that are going to use only occasionally. This would be greatly interesting not only for them but also for developers of R packages, as the impact of their work would increase notoriously. And, of course, it would promote R to an even more influential role than already has in the history of statistical computing.

References

- Austenfeld M, Beyschlag W (2012). “A Graphical User Interface for R in a Rich Client Platform for Ecological Modeling.” *Journal of Statistical Software*, **49**(4), 1–19. URL <http://www.jstatsoft.org/v49/i04/>.
- Becker RA, Chambers JM, Wilks AR (1988). *The New S Language*. Wadsworth & Brooks/Cole.
- Cheng YS, Peng CY (2012). “Integrated Degradation Models in R Using **iDEMO**.” *Journal of Statistical Software*, **49**(2), 1–22. URL <http://www.jstatsoft.org/v49/i02/>.
- De Leeuw J (2009). “Statistical Software – Overview.” *Technical report*, UCLA Department of Statistics. URL http://www.stat.ucla.edu/~deleeuw/janspubs/2009/reports/deleeuw_R_09a.pdf.
- Fellows I (2012). “**Deducer**: A Data Analysis GUI for R.” *Journal of Statistical Software*, **49**(8), 1–15. URL <http://www.jstatsoft.org/v49/i08/>.
- Fox J (2005). “The R Commander: A Basic-Statistics Graphical User Interface to R.” *Journal of Statistical Software*, **14**(9), 1–42. URL <http://www.jstatsoft.org/v14/i09/>.
- Fox J, Carvalho MS (2012). “The **RcmdrPlugin.survival** Package: Extending the R Commander Interface to Survival Analysis.” *Journal of Statistical Software*, **49**(7), 1–32. URL <http://www.jstatsoft.org/v49/i07/>.

- Huang B, Cook D, Wickham H (2012). “**tourrGui**: A **gWidgets** GUI for the Tour to Explore High-Dimensional Data Using Low-Dimensional Projections.” *Journal of Statistical Software*, **49**(6), 1–12. URL <http://www.jstatsoft.org/v49/i06/>.
- Ihaka R, Gentleman R (1996). “R: A Language for Data Analysis and Graphics.” *Journal of Computational and Graphical Statistics*, **5**(3), 299–314.
- Ihaka R, Temple Lang D (2008). “Back to the Future: Lisp as a Base for a Statistical Computing System.” In P Brito (ed.), *COMPSTAT 2008*, pp. 21–33. Springer-Verlag.
- R Development Core Team (2012). *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/>.
- Rödiger S, Friedrichsmeier T, Kapat P, Michalke M (2012). “**RKward**: A Comprehensive Graphical User Interface and Integrated Development Environment for Statistical Analysis with R.” *Journal of Statistical Software*, **49**(9), 1–34. URL <http://www.jstatsoft.org/v49/i09/>.
- SAS Institute Inc (2012). *JMP 10: Statistical Discovery Software*. Cary, NC. URL <http://www.jmp.com/>.
- Snellenburg JJ, Laptinok SP, Seger R, Mullen KM, van Stokkum IHM (2012). “**Glottaran**: A Java-Based Graphical User Interface for the R Package **TIMP**.” *Journal of Statistical Software*, **49**(3), 1–22. URL <http://www.jstatsoft.org/v49/i03/>.
- Tierney L (1990). *LISP-STAT: An Object Oriented Environment for Statistical Computing and Dynamic Graphics*. John Wiley & Sons, New York.
- Unwin A (2012). “Oscars and Interfaces.” *Journal of Statistical Software*, **49**(11), 1–18. URL <http://www.jstatsoft.org/v49/i11/>.
- Velleman PF (1997). *DataDesk Version 6.0 – Statistics Guide*. Data Description Inc., Ithaka.
- Verzani J (2007). “An Introduction to **gWidgets**.” *R News*, **7**(3), 26–33. URL <http://CRAN.R-project.org/doc/Rnews/>.
- Verzani J (2012). “**gWidgetsWWW**: Creating Interactive Web Pages within R.” *Journal of Statistical Software*, **49**(10), 1–12. URL <http://www.jstatsoft.org/v49/i10/>.
- Wallace BC, Dahabreh IJ, Trikalinos TA, Lau J, Trow P, Schmid CH (2012). “Closing the Gap between Methodologists and End-Users: R as a Computational Back-End.” *Journal of Statistical Software*, **49**(5), 1–15. URL <http://www.jstatsoft.org/v49/i05/>.
- Wickham H, Cook D, Hofmann H, Buja A (2011). “**tourr**: An R Package for Exploring Multivariate Data with Projections.” *Journal of Statistical Software*, **40**(2). URL <http://www.jstatsoft.org/v40/i02/>.

Affiliation:

Pedro M. Valero-Mora
Universitat de València
Department of Methodology of Behavioral Sciences
Valencia, Spain 46010
E-mail: valerop@uv.es
URL: <http://www.uv.es/valerop/>

Rubén D. Ledesma
CONICET (Consejo Nacional de Investigaciones Científicas y Técnicas)
and
Universidad Nacional de Mar del Plata
Facultad de Psicología
Funes 3350
Mar del Plata (7600), Argentina
E-mail: rdledesma@conicet.gov