R++, User-Friendly Statistical Software

Christophe Genolini[1(✉)], Emmanuel Dubois[2], and David Furió[1]

[1] Zebrys, 5 place Jean Deschamps, 31100 Toulouse, France
cg@rplusplus.com
[2] IRIT-LIIHS, 188 route de Narbonne, 31062 Toulouse Cedex 4, France

Abstract. Statistical analysis is gradually entering all areas of society, be in academia or in the private sector. Statistical software is used by statisticians but also by non-experts (medical doctors, psychologists...). Unfortunately, this kind of software is integrated into obsolete interfaces that completely ignore the principles of HCI and are poorly adapted to non-expert users.

R++ project aims to develop a modern statistical analysis software program integrated into a user-friendly interface. In this paper, we present the methodology that led us to the design of R++. We also give two examples that this methodology allowed us to achieve.

Keywords: Statistical analysis · Video prototyping · R++

1 Introduction

Statistical analysis is gradually entering all areas of society [1]. In the academic world, it is becoming more and more difficult to publish an article without some tests or some model. In the private sector, insurers [2] use it to define their rates, bankers [3] to decide whether or not to grant a loan, the pharmaceutical industry [4,5] to validate its clinical trials, etc.

The users of statistical analysis are therefore very diverse. Some are experts like Data Scientists, but many are casual users like medical doctors, psychologists, educational scientists, pharmacists, etc. In order to publish international papers, they need high level statistics despite the fact that they are not statisticians.

From a software point of view, statistician software (SAS©, R, SPSS, Stata...) is complete in terms of methods available, but performs very poorly [7]. On the other hand, some languages (C, python) or software (Oracle) are very powerful but offer very few analytical tools and require serious programming skills. Furthermore, both types of software are integrated into obsolete interfaces that completely ignore the principles of HCI and are poorly adapted to non-expert users (Figure S1).

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-29390-1_46) contains supplementary material, which is available to authorized users.

Published by Springer Nature Switzerland AG 2019
D. Lamas et al. (Eds.): INTERACT 2019, LNCS 11749, pp. 605–609, 2019.
https://doi.org/10.1007/978-3-030-29390-1_46

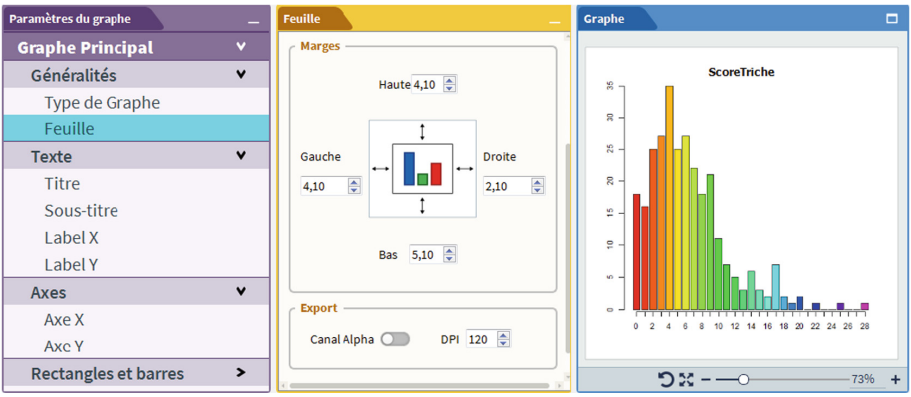

Fig. 1. The R++ project (graph modification interface).

The R++ project (Fig. 1) aims to develop modern statistical analysis software integrated into a user-friendly interface.

In this paper, we present the methodology that led us to the design of R++. Section 2 briefly recalls the concept of exploratory design and video prototyping. Sections 3 and 4 details elements that have been improved. Section 5 is the conclusion.

2 Methods

To conceive the user interface of R++, we used the exploratory design method and video prototyping: during 1 hour, statisticians and computer specialists brainstormed on: *What are the difficult, tedious, time-consuming or highly error-prone tasks with your current statistical software?* This round table meeting was done in an open mode: the participants intervened and a moderator took notes on the board. At the end of this first step, we selected two or three of the themes that were most frequently mentioned.

In the second step, we sought solutions for the previously selected themes: *Imagine the statistical analysis software of your dreams: how would it work?* For 10 min, participants were asked to write down three ideas, plus a "far-fetched" idea. The objective of this out-of-the-box idea was to avoid self-censorship: indeed, a participant may have a good idea but not say it for fear of being ridiculous. In doing so, it helped avoiding the said problem. When everyone had found their three plus one ideas, they were presented to the rest of the group and then debated, combined, and hopefully improved.

Finally, during the third hour, the participants created low-fidelity prototypes (Figure S2). With the help of paper, felt pens, post-its, and cut-outs, they created a scenario that they filmed using a telephone. These prototypes were then presented to the entire group. This provided the first feedback.

In total, we organized eight general sessions without specific instructions. We also organized thematic sessions where the agenda was announced in advance (e.g. "merging databases"). Only the statisticians directly concerned by the problem came to these sessions. The course of the session was then modified: (1) search for solutions (2) presentation of videos related to the theme discussed during the general sessions (3) creation of video prototypes.

During the "things to improve" phase, many ideas were suggested. They were then collectively grouped into frequently asked themes (see Table 1):

Table 1. Thematic grouping of the elements to be improved in statistical software.

Data-management (outlier detection, wrong type detection, merging databases,...)	33
Graphs (dynamic, interactive, automatic, exportable, DPI)	24
R Code (unified syntax, code folding, automatic generation, always visible)	17
Results layout (LaTeX or Word, table management)	16
Tables (interactives, with style)	16
Help (integrated, video, contextual, wiki)	14
Export (to Word, automatic reporting, document styling)	13
Timeline of the analysis	9
Interactive software (interactive graphs, interactive data)	7
Others	41

3 Focus on Data Management

Data management consists in preparing the data before analysis. Strictly speaking, this step is not really part of the statistical analysis but, unfortunately, it is essential. It consists of identifying inconsistencies in the data, such as a 350-year-old person, a binary variable encoded in three modalities: Female, Male and male[1], or an execution time of one hour for a task that should take a few seconds. This step does not require any special skills, and it is considered quite tedious by statisticians. This probably explains the large number of times it has been cited as an area for improvement.

Data management generally involves studying certain statistics (minimum, maximum, number of employees) and a graphical representation for each of the variables. A solution to simplify this process was to automate the calculation of the indices in question and the associated graphical representation. Thus, with a single mouse-click, it is possible to obtain a graphical representation of all the columns. The user can detect the problematic variables at a glance (Fig. 2 or S3). We also add a color in each column according to its type, e.g. the binary columns are in yellow. If a column that is supposed to be binary is not in yellow, then there is probably a problem in said column (Figure S4).

[1] Modalities are case sensitive by default, therefore, Male and male represent different modalities.

4 Focus on Graphs

The second highly problematic point concerns graphs. Traditional software produces graphs that are generally austere, fixed, and can only be manipulated by command lines. Statisticians are often forced to use additional software (Photoshop, Gimp) that they do not master well. In order to solve this, the following solutions have been chosen:

- Interactive graphs allow, when you click on a part of the graph (for example on an outlier) to select the corresponding data. This action also sorts the database so that the selected data is present at the top of the column. This makes it possible to access specific values very quickly.
- Some graphs are mandatory parts of an analysis. Their production has been automated: univariate analysis (Fig. 2 and S3) but also statistical test graphs (Fig. 3).
- An interface allows the user to easily modify the graph settings. A first window groups the parameters by theme. When a theme is selected, the details of the options are presented in a second window. The third shows the graph. For example, the publication of articles requires drawings with a resolution of 300 DPI and no alpha channel. Few statisticians know what these two terms mean. The interface simplifies the modification of these two parameters (Fig. 1 and S5).
- Finally, exporting a graph to other software can be done by a simple drag&drop (Figure S3).

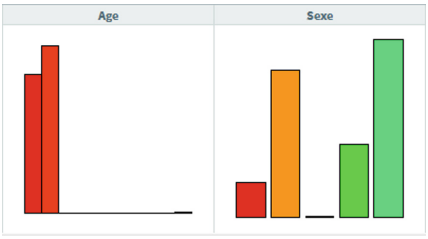

Fig. 2. Detection of problematic variables at a glance. There is an outlier in `Age` and `Sex` is incorrectly coded.

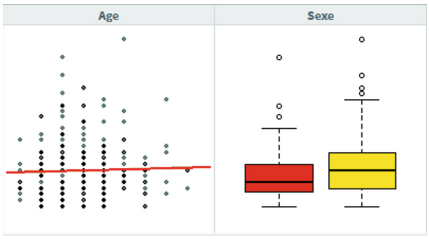

Fig. 3. Bivariate graphical representation. `Age` and `Sexe` relatively to another previously selected variable.

5 Conclusions

In this article, we have presented the methodology that led us to design R++, a statistical analysis software integrated into a user-friendly interface.

There are still many areas in which statistical analysis could benefit from HCI contributions. In Fig. 2, the appearance of the graphs limits the position of

the displayed data. User feedback could allow us to determine if it is a blocking element. If so, could solutions such as *perspective wall* [8] help? Or could *Focus+Context* [9] be adapted to the statistical context? Similarly, the graph interface requires you to leave the data page. Would it be possible to use a more direct configuration solution, such as Magic Lenses? [10].

Another interesting field, one of the pillars of science is the possibility of being able to reproduce an analysis. All statistical software therefore produces code. But statisticians are not programmers. Similarly, unlike some computer codes, which no longer need to be modified when they are fully functional, statisticians' codes are intended to be read again by others (to verify the analyses). It would therefore be interesting to work on a script editor that would make programming accessible to the novice and facilitate code review.

Finally, statistical analysis requires the simultaneous display of a large amount of information. Statisticians almost always have two screens, although this is not always enough. Moving part of the interface (data control, 3D graph display control,...) to an external device could optimize the display of information on the screen.

References

1. Muenchen, R.A.: The Popularity of Data Science Software. http://r4stats.com/articles/popularity/. Accessed 19 Apr 2019
2. Grize, Y.L.: Applications of statistics in the field of general insurance: an overview. Int. Stat. Rev. **83**(1), 135–159 (2015)
3. Hand, D.: Statistics in Banking. Published online in Encyclopedia of Statistical Sciences. https://doi.org/10.1002/9781118445112.stat00179. Accessed 19 Apr 2019
4. Lewi, P.J.: The role of statistics in the success of a pharmaceutical research laboratory: a historical case description. J. Chemom. **19**(5–7), 282–287 (2005)
5. Peterson, J.J., Snee, R.D., McAllister, P.R., Schofield, T.L., Carella, A.J.: Statistics in pharmaceutical development and manufacturing. J. Qual. Technol. **41**(2), 111–134 (2009). https://doi.org/10.1080/00224065.2009.11917764
6. R Core Team: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2019). https://www.R-project.org/
7. Chai, A.: Accélération des méthodes statistiques sur GPU. Master 2 internship report. https://rplusplus.com/wp-content/uploads/2018/03/Rapport_Anchen03.pdf. Accessed 19 Apr 2019
8. Mackinlay, J.D., Robertson, G.G., Card, S.K.: The perspective wall: detail and context smoothly integrated. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 1991), pp. 173–176 (1991). https://doi.org/10.1145/108844.108870
9. Cockburn, A., Karlson, A., Bederson, Benjamin, B.: A review of overview+detail, zooming, and focus+context interfaces. ACM Comput. Surv. **41**(1), 31 p. (2009). Article no. 2. https://doi.org/10.1145/1456650.1456652
10. Bier, E.A., Stone, M.C., Pier, K., Buxton, W., DeRose, T.D.: Toolglass and magic lenses: the see-through interface. In: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1993), pp. 73–80 (1993). https://doi.org/10.1145/166117.166126