Muste – editorial computing environment within R

Reijo Sund, *National Institute for Health and Welfare, Finland*, `reijo.sund@thl.fi`
Kimmo Vehkalahti, *University of Helsinki, Finland*, `kimmo.vehkalahti@helsinki.fi`
Seppo Mustonen, *University of Helsinki, Finland*, `seppo.mustonen@survo.fi`

Abstract. Statistical computing requires appropriate tools. We describe the integrated editorial environment for data analysis and related tasks developed since the 1960s under the name of Survo and its recent open source R package implementation known as Muste. As examples of the editorial approach in Muste, we consider a statistical application on influence curves for a correlation coefficient as well as on simulation and contour curves of bivariate Gaussian data.

Keywords. Muste, Survo, R project, statistical software, text editor, user interface, data analysis, bivariate normal distribution, history of statistical computing

1 Introduction

Statistical computing is not possible without appropriate tools. In practice, any statistician working with concrete data requires software packages that help to analyse data with computers. Quite recently, the `R` project [21] has become an increasingly popular tool that allows enough flexibility even for very specialized tasks requiring novel approaches to data analysis [4]. R is an open source software package that has roots in a research project of *Robert Gentleman* and *Ross Ihaka* in the 1990s [6]. R originated as a `Scheme` variant, but has developed into a dialect of `S` language that was introduced in the Bell Labs by *John Chambers, Rick Becker* and others in 1976 [2].

Another example of a flexible tool is `Survo`, an integrated environment for statistical computing and related areas, developed since the 1960s by *Seppo Mustonen* [1, 9, 13, 15, 17, 20]. The central feature of Survo is the *editorial interface* that was introduced in 1979: *"It seems quite natural to extend editing operations towards normal statistical operations and this will be a new form of interactive statistical computing which covers the final documentation as well"* [12]. Since then, Survo has allowed statistical computing and data analysis to be an inseparable part of writing and reporting the results in a research process. Somewhat similar ideas have been described more specifically under different headings such as literate programming [7], reproducible research and compendium [5]. Certain properties of using statistical programs from a text editor are also present in the `Emacs Speaks Statistics` approach [22], `TeXmacs` platform [24] and `Mathematica` notebooks [25]. However, the main distinction is that Survo has always provided a general, integrated and self-documenting environment for statistical computing and related areas in one stand-alone software program.

The aim of this paper is to describe how Survo and R have been combined in the `Muste` project that was initiated in 2009 by *Reijo Sund*. So far we have successfully implemented the key parts of the Survo system as an open source R package Muste.

We begin with a brief review of the long history of Survo and the short history of Muste. As examples of the editorial approach in Muste, we consider a few small statistical applications.

2 Survo – statistical programming language since the 1960s

The roots of statistical software program Survo date back to 1960, when young Seppo Mustonen was recruited in the Electronics Department of the Finnish Cable Works (that later emerged as the Nokia Corporation) to develop a library of statistical operations for the Elliott 803 computer [8]. In 1962, Mustonen attended the congress of International Federation for Information Processing at Münich, Germany. There, during a coffee break on a hot summer day in the late August, Mustonen and his colleague *Martti Tienari* told professor *Olli Lokki* about the ideas concerning the future of data processing developed in the Finnish Cable Works. Professor Lokki was very enthusiastic about the more or less specific dreams about how the things related to statistical computing should be arranged. During those discussions the aim to develop a whole statistical programming language was presented for the first time. Mustonen extended these ideas with Tienari and *Timo Alanko* during the years 1963–1964 and prepared a first proposal for a statistical programming system SURVO 64. Survo refers to the word "survey", but it also has a connotation with the Finnish verb "survoa" ("to compress") [15].

The first fully implemented system was SURVO 66 on Elliott 803 computer. The system allowed the user to define the analyses to be performed using a simple command language [1]. The SURVO 66 was implemented also for Elliott 503. An implementation for Honeywell H1640 series was developed during the 1970s at the University of Tampere, Finland, and it was known by the name of SURVO/71. The next generation of Survo was the SURVO 76 system working on the Wang 2200 minicomputer [9]. The PC version SURVO 84 [13] and especially its successor, the first C language implementation SURVO 84C [14] expanded to resemble a statistical operating system or integrated environment instead of merely a program for statistical analyses [15]. Along with SURVO 98 the program code was converted for 32 bit systems, and in 2001, a Windows version SURVO MM was introduced. At the beginning of May 2012, SURVO MM was on its version 3.36.

Editorial approach and general computing environment

Since the very first ideas of Survo in the 1960s, the challenges of integrating the various activities and the interactivity between the user and the computer have had an important role in the development of Survo. While the computers of the 1960s were only able to satisfy the wishes to a slight degree, the computers of the next decade offered much better options.

The SURVO 76 system on a Wang minicomputer was one of the first, truly interactive statistical packages. It provided an interactive menu-driven environment in addition to a command language, and its use was like a conversation with the computer [9, 10]. The most remarkable single advance took place in 1979, when Mustonen invented the so called *editorial approach* for data analyses. This approach means that the interactivity in data analysis is achieved by working within a text editor that allows a free mixture of natural language, data, activatable commands and results [11]. The idea originated from the Mustonen's need to have a "paper and pen" type of interface to work with computers. Somewhat surprisingly, the Survo editor was originally developed to help Mustonen's son *Olli Mustonen* to write and print music sheets. Writing of music sheets certainly was not the main purpose of an interactive statistical program, but these experiments convinced Mustonen to develop the idea further. The editorial approach turned out to provide such a flexibility and freedom to the user that it completely replaced the "old-fashioned", menu-driven user interaction in Survo at the beginning of the 1980s. Since then, the idea and features of the editorial approach have been continuously extended in many ways.

Muste is an R implementation of Survo

The main motivation for the Muste project was to transform the unique or otherwise useful properties of Survo system into the form of an open source multiplatform software. As the R project had become very popular, offering an appealing environment for implementing open source statistical software for multiple platforms, it was a natural choice to test the possibilities of producing a Survo-look-alike functionality in that context.

It quickly turned out that the text widgets available via the `tcltk` package [3], a base part of R, would allow the implementation of the Survo editor in Muste. The initial tests in 2009 were promising and Seppo Mustonen kindly provided all the necessary parts of the Survo source code written in C for the Muste project. Due to this and the fact that virtually all platform specific parts of Survo had been isolated and thus made easily replacable, the only functions that had to be re-programmed from scratch, were the input and output functions. The flexibility of R made it possible to creatively utilize a mixture of C, Tcl/Tk and R to create a multiplatform package as reported elsewhere with more technical details [23].

The development of Muste has taken place in different platforms including Linux, Mac OS X and Windows. Most of the functionality of the Survo system has already been included in Muste. Muste is available for anyone interested from the R-Forge development platform: `http://muste.r-forge.r-project.org/`.

3 Examples on editorial approach in Muste

After installing the Muste package in R, it can be loaded by the function `library(muste)`. When the editor is launched with `muste()`, a new editor window, similar to the one presented in Figure 1 appears. Muste editor works as a text editor in which the user can freely write text on an *edit field*. The text may contain commands, operations, specifications and other parts that Muste will interpret as the user *activates* them, by the `Esc` key or the mouse double-click.

The header line displays the current date and time, the working directory, the size of the edit field, and certain status information. At the beginning of each line there is a line number followed by a *control column* that typically contains asterisks (`*`), but may include any characters. Different colors are obtained by using "shadows" for the text. Technically, the shadows are another layer of text reflected in different colors in the edit field. In general, the shadows may have both visual and structural purposes in various operations. Throughout this example, they are merely used to emphasize the distinction between the commands, data, output and other

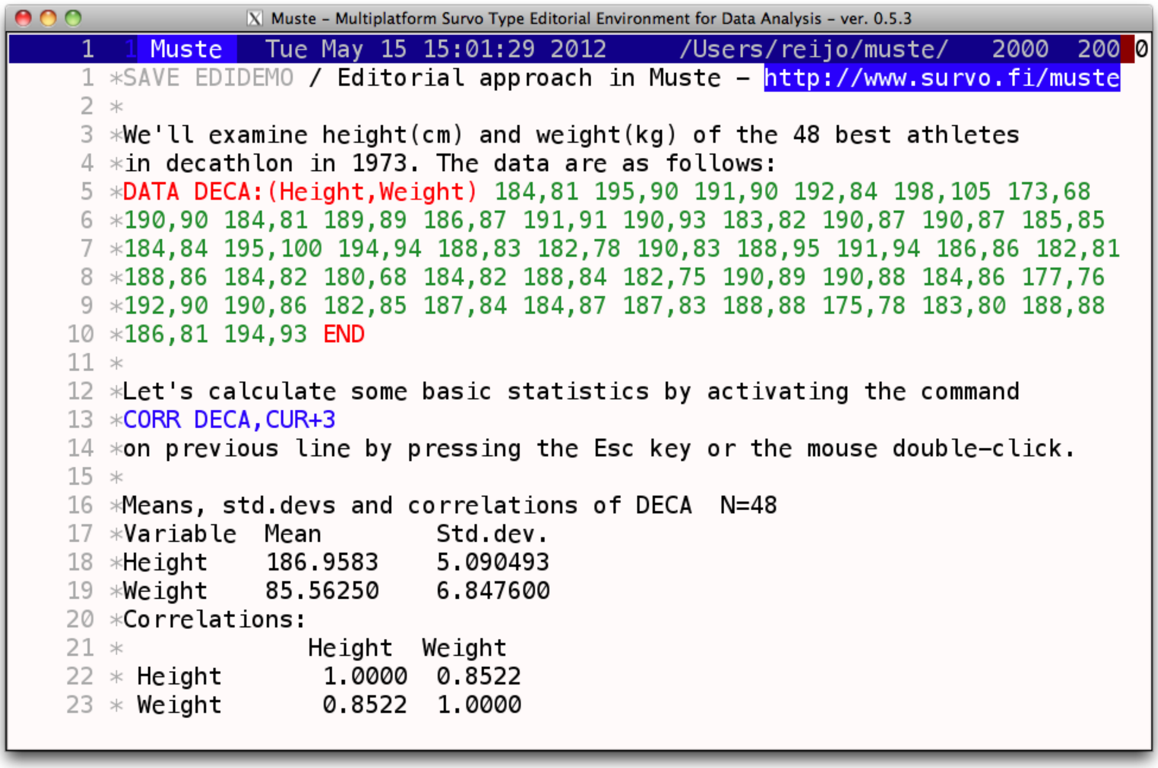


Figure 1: Editorial approach in Muste (1/4)

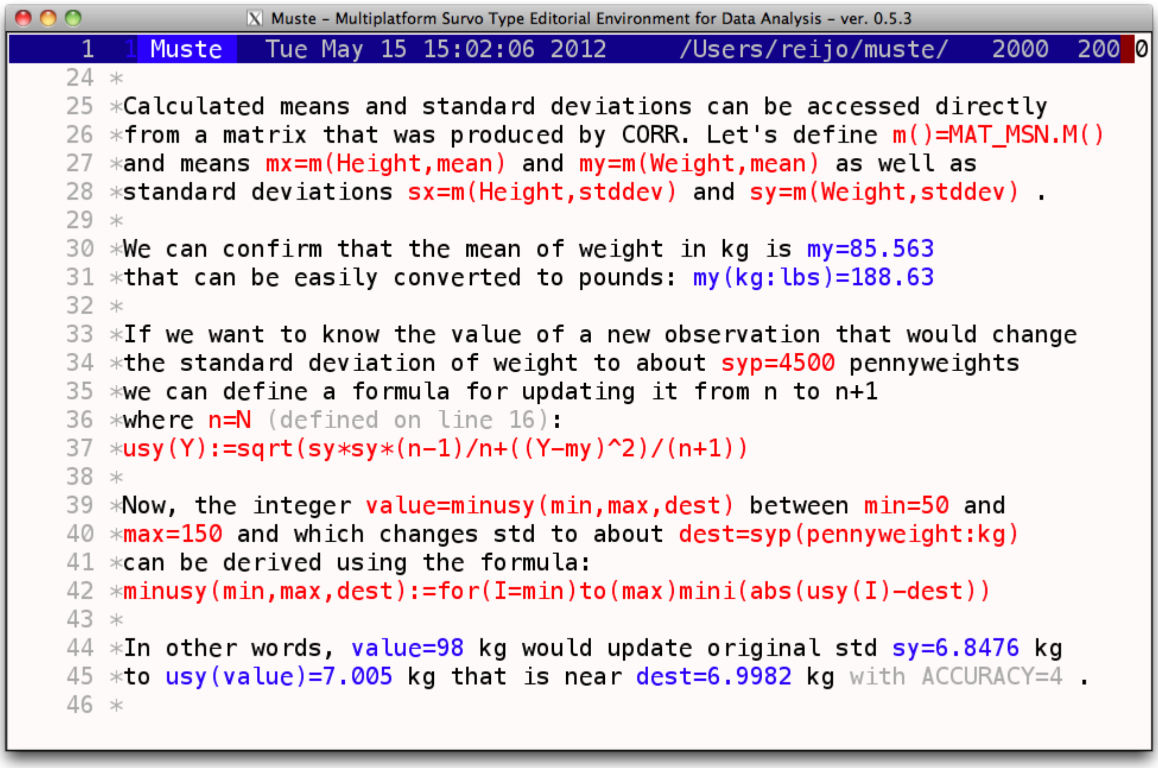
parts of the text in the Muste editor. Usually, there is no need to highlight this distinction explicitly. Instead, the user may choose whatever (s)he wants to highlight with the shadows.

Since both the data and the commands can be displayed and handled together in the edit field, the user is in close contact with his/her work, with the possibility to control each detail of the computational and editorial activities. Therefore, the editorial approach is more than just running "programs" from a command line or using a script editor to write code to be executed, as is typical in R. The editorial approach gives more versatile possibilities for mixing text, commands, data and results, because any elements can be freely combined in the edit field, with their exact meaning to be interpreted dynamically, depending on the context at the moment of the activation.

In Figure 1, both data and commands can be seen, in addition to free-form text. As soon as the user activates the `CORR` command on line 13, the data referenced in the command line will be "found" on the the lines 5–10. These data will then be used to produce the results that immediately appear on lines 16–23.

Editorial arithmetics and conversions

Muste editor contains a powerful tool for making many types of calculations with a flexible interface. The example that continues in Figure 2, shows how to access the results of the `CORR` command in editorial arithmetics. Further, on line 31, a conversion from one measurement unit



```
Muste – Multiplatform Survo Type Editorial Environment for Data Analysis – ver. 0.5.3
   1   1 Muste    Tue May 15 15:02:06 2012       /Users/reijo/muste/    2000  200 0
  24 *
  25 *Calculated means and standard deviations can be accessed directly
  26 *from a matrix that was produced by CORR. Let's define m()=MAT_MSN.M()
  27 *and means mx=m(Height,mean) and my=m(Weight,mean) as well as
  28 *standard deviations sx=m(Height,stddev) and sy=m(Weight,stddev) .
  29 *
  30 *We can confirm that the mean of weight in kg is my=85.563
  31 *that can be easily converted to pounds: my(kg:lbs)=188.63
  32 *
  33 *If we want to know the value of a new observation that would change
  34 *the standard deviation of weight to about syp=4500 pennyweights
  35 *we can define a formula for updating it from n to n+1
  36 *where n=N (defined on line 16):
  37 *usy(Y):=sqrt(sy*sy*(n-1)/n+((Y-my)^2)/(n+1))
  38 *
  39 *Now, the integer value=minusy(min,max,dest) between min=50 and
  40 *max=150 and which changes std to about dest=syp(pennyweight:kg)
  41 *can be derived using the formula:
  42 *minusy(min,max,dest):=for(I=min)to(max)mini(abs(usy(I)-dest))
  43 *
  44 *In other words, value=98 kg would update original std sy=6.8476 kg
  45 *to usy(value)=7.005 kg that is near dest=6.9982 kg with ACCURACY=4 .
  46 *
```

Figure 2: Editorial approach in Muste (2/4)

to another is carried out. The results are written to the edit field by activating the calculation after the equal sign. In addition to a large selection of various conversion options available in Muste, the user can easily add new ones by defining a few simple rules. Certain special transformations, such as numbers with different bases, roman numerals, written words or factors of an integer are included by default.

As a slightly more complicated example, we consider adding one new observation to the data and study its effect on basic statistics. Beginning from the mean of a variable x with n observations, we have trivially

$$\overline{x}(n+1) = \frac{1}{n+1}\left(\sum_{i=1}^{n} x_i + x\right). \tag{1}$$

Similarly, Mustonen [18] has shown that the updated variance of x can be presented as

$$s_x^2(n+1) \quad = \quad \frac{n-1}{n} s_x^2(n) + \frac{1}{n+1}[x - \overline{x}(n)]^2, \tag{2}$$

where $\overline{x}(n)$ is the mean and $s_x^2(n)$ the variance of x with n observations.

In order to use the equation 2 in editorial arithmetics of Muste, it is defined on line 37 of the example in Figure 2. On line 42, we define a function that looks for a value for the weight fulfilling the conditions, using a grid search over integer values on a reasonable interval. The results are achieved by activating the calculations on lines 44–45. The point of these small “toy” examples is to demonstrate the possibilities of mixing natural text, numbers, definitions of equations and actual calculations in the edit field of Muste.

Influence curves for the correlation coefficient

Our example continues in Figure 3, where we proceed with the correlation coefficient of two variables, x and y. According to Mustonen [18], an update formula for the correlation coefficient can be written as

$$r(n+1) = \frac{r(n) + uv}{\sqrt{(1+u^2)(1+v^2)}}, \tag{3}$$

where

$$u = \sqrt{\frac{n}{n^2-1}}\frac{x - \overline{x}(n)}{s_x(n)}, \qquad v = \sqrt{\frac{n}{n^2-1}}\frac{y - \overline{y}(n)}{s_y(n)}, \tag{4}$$

and $r(n)$ is the correlation coefficient with n observations.

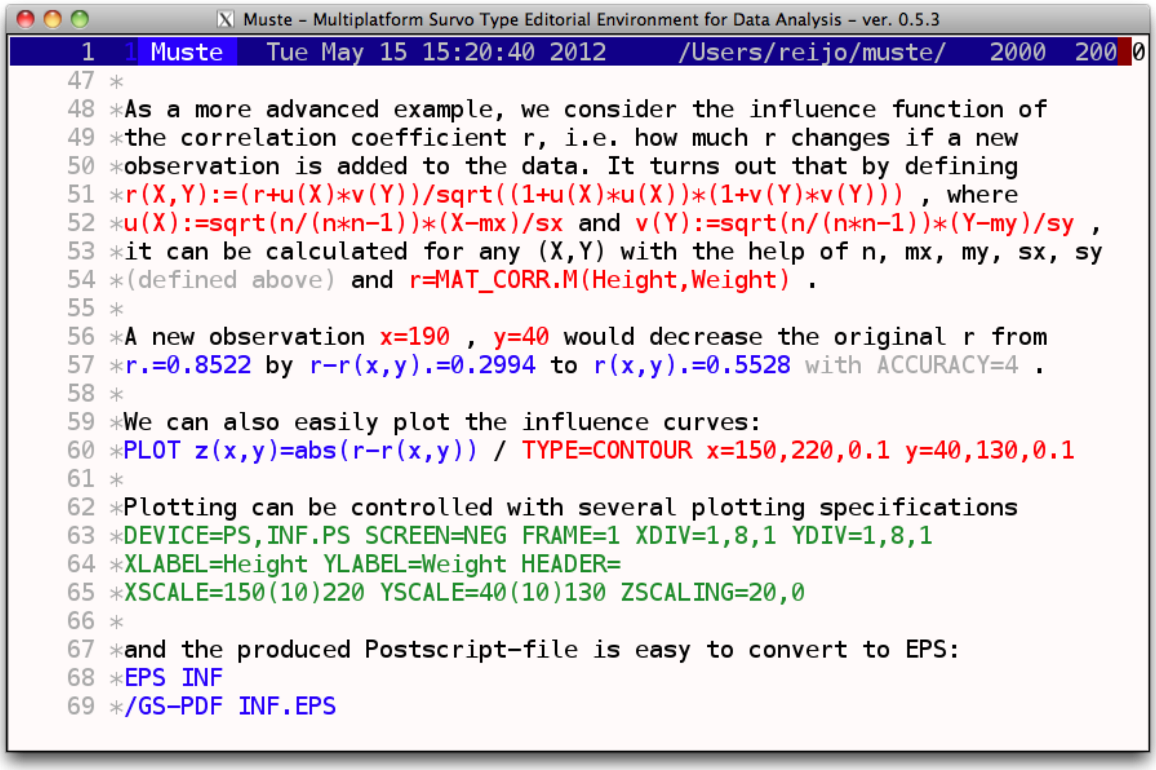


Figure 3: Editorial approach in Muste (3/4)

The update formulas 3 and 4 are defined on lines 51–52 of the example in Figure 3. They are then applied in editorial arithmetics on lines 56–57. In addition, they are used below the calculations to plot a graph.

The work scheme on lines 60–65 is intended for plotting scatter diagrams with appropriate contour curves which visualize the robustness of the correlation coefficient. Actually these influence curves will appear as contours of a raster image of the influence function $|r(n+1) - r(n)|$ given as $z(x, y)$ on lines 51–52 and 60, indicating the amount of the increase or decrease of the correlation coefficient when a new observation (x, y) is obtained. The final graph presented in Figure 4 is produced by activating the `PLOT` command on line 60.

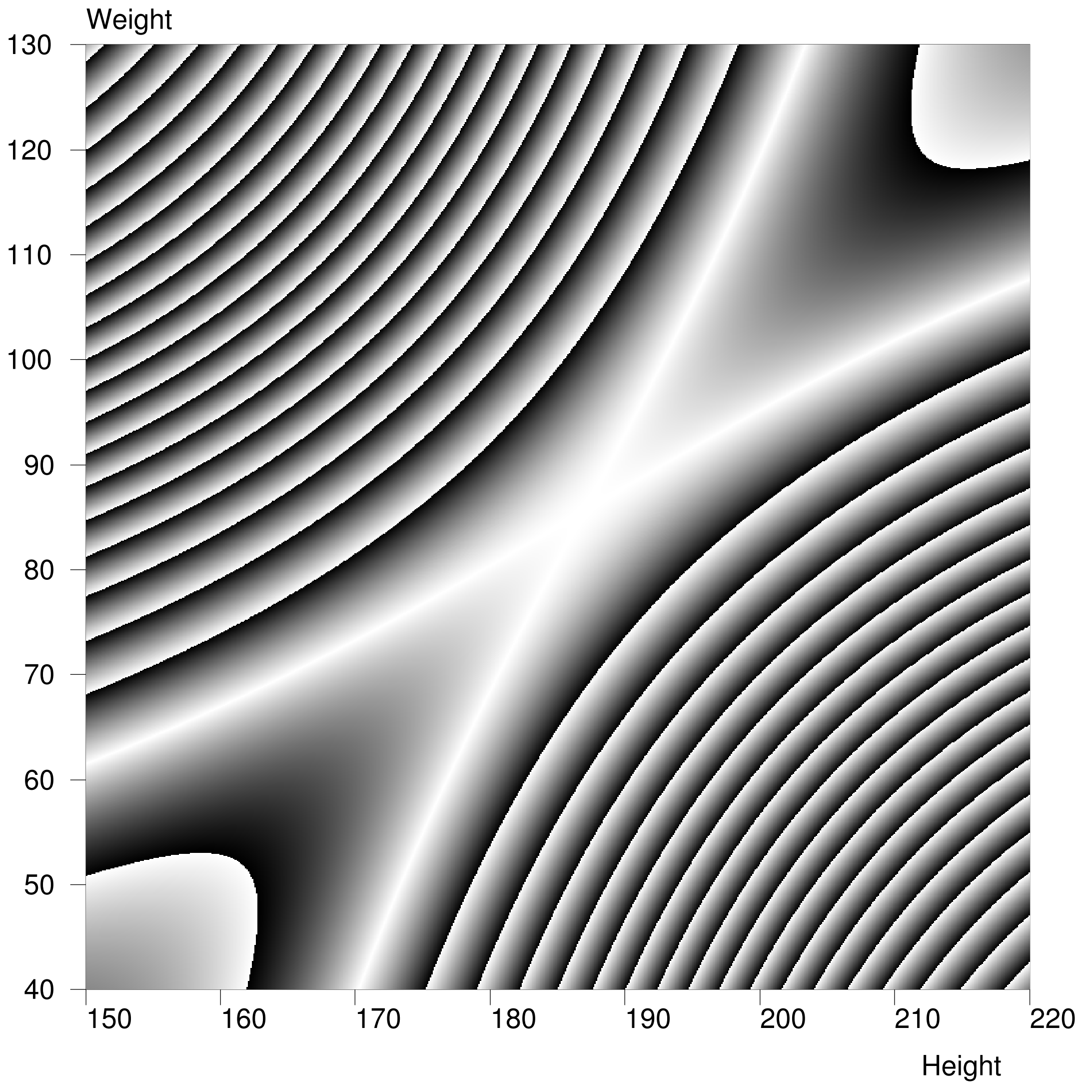


Figure 4: Influence curves for the correlation coefficient

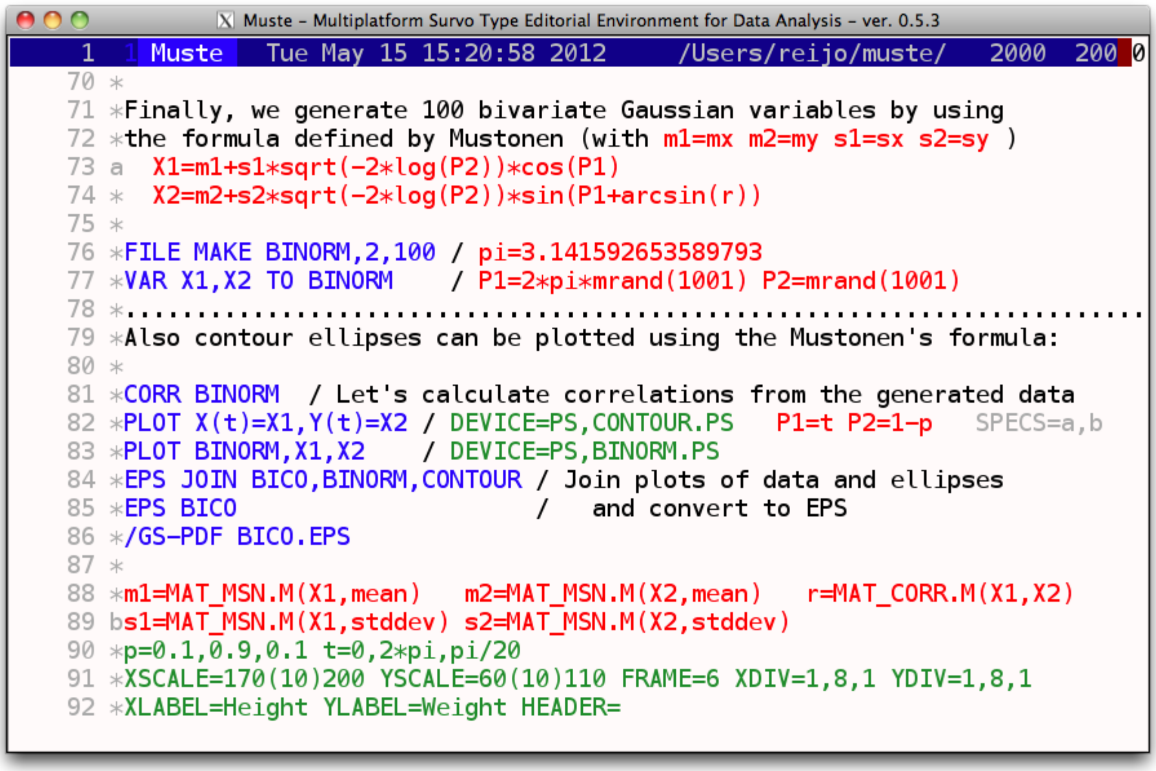
When making the raster image, the values of the function `z` are mapped continuously to various shades of gray in such a way that 0 corresponds to "black" and 1 corresponds to "white". If the function values exceed 1 the shading is selected "modulo" 1. In this case, the original function values are multiplied by 20 (by `ZSCALING=20,0`) which gives a complete cycle of shadings when the function value changes by $1/20 = 0.05$. Thus, the final graph depicts contours of r with increments of 0.05. The specification `DEVICE=PS,INF.PS` on line 63 implies the graph to be produced as a PostScript picture and saved in file `INF.PS`.

Contour ellipses for bivariate Gaussian data

We conclude the example by simulating data from a bivariate Gaussian distribution and by drawing contour ellipses for the simulated data. Muste includes a specialized command `MNSIMUL` for simulating data from multinormal distribution, but here we construct the variables manually, using formulas for bivariate Gaussian distribution presented by Seppo Mustonen [16, 19], who derived them as generalizations to the well known Box–Muller formulas.

Let U_1 and U_2 be independent variables from the uniform distribution on $(0,1)$. Then a two-dimensional normal variable (X_1, X_2) with expected values (μ_1, μ_2), standard deviations (σ_1, σ_2) and correlation ρ is generated by

$$\begin{aligned} X_1 &= \mu_1 + \sigma_1 \sqrt{-2\log(U_2)}\cos(2\pi U_1) \\ X_2 &= \mu_2 + \sigma_2 \sqrt{-2\log(U_2)}\sin(2\pi U_1 + \arcsin(\rho)). \end{aligned} \tag{5}$$



```
Muste - Multiplatform Survo Type Editorial Environment for Data Analysis - ver. 0.5.3
   1  1 Muste   Tue May 15 15:20:58 2012       /Users/reijo/muste/    2000   2000
  70 *
  71 *Finally, we generate 100 bivariate Gaussian variables by using
  72 *the formula defined by Mustonen (with m1=mx m2=my s1=sx s2=sy )
  73 a  X1=m1+s1*sqrt(-2*log(P2))*cos(P1)
  74 *  X2=m2+s2*sqrt(-2*log(P2))*sin(P1+arcsin(r))
  75 *
  76 *FILE MAKE BINORM,2,100 / pi=3.141592653589793
  77 *VAR X1,X2 TO BINORM    / P1=2*pi*mrand(1001) P2=mrand(1001)
  78 *.....................................................................
  79 *Also contour ellipses can be plotted using the Mustonen's formula:
  80 *
  81 *CORR BINORM  / Let's calculate correlations from the generated data
  82 *PLOT X(t)=X1,Y(t)=X2 / DEVICE=PS,CONTOUR.PS   P1=t P2=1-p   SPECS=a,b
  83 *PLOT BINORM,X1,X2    / DEVICE=PS,BINORM.PS
  84 *EPS JOIN BICO,BINORM,CONTOUR / Join plots of data and ellipses
  85 *EPS BICO                    /   and convert to EPS
  86 */GS-PDF BICO.EPS
  87 *
  88 *m1=MAT_MSN.M(X1,mean)   m2=MAT_MSN.M(X2,mean)   r=MAT_CORR.M(X1,X2)
  89 bs1=MAT_MSN.M(X1,stddev) s2=MAT_MSN.M(X2,stddev)
  90 *p=0.1,0.9,0.1 t=0,2*pi,pi/20
  91 *XSCALE=170(10)200 YSCALE=60(10)110 FRAME=6 XDIV=1,8,1 YDIV=1,8,1
  92 *XLABEL=Height YLABEL=Weight HEADER=
```

Figure 5: Editorial approach in Muste (4/4)

On lines 76–77 of the example that continues in Figure 5, a new empty data file with two variables and hundred observations is generated with `FILE MAKE`. The random variables are generated by the `VAR` command that utilizes the definitions of the Mustonen formulas on lines 73–74. The means, standard deviations and the correlation coefficient come from the `DECA` data introduced in the beginning of the example, in Figure 1.

We could plot the data and include the contour ellipses of the bivariate normal distribution, simply using a specification `CONTOUR` with the `PLOT` command on line 83. Here we, however, plot the contour ellipses manually. As proved by Mustonen [16, 19], the contour ellipses of the bivariate normal distribution can be presented in a parametric form, where $0 \leq t \leq 2\pi$ and p is the confidence level, with formulas

$$\begin{aligned} X_1 &= \mu_1 + \sigma_1\sqrt{-2\log(1-p)}\cos(t) \\ X_2 &= \mu_2 + \sigma_2\sqrt{-2\log(1-p)}\sin(t+\arcsin(\rho)). \end{aligned} \tag{6}$$

In our work scheme, we actually take use of the same equations (on lines 73–74) which were earlier used in the simulation of the data. The actual `PLOT` command for the contour ellipses is on line 82. The `EPS` command on line 84 combines the plots of the data and the contour curves as one PostScript file. The final graph is displayed in Figure 6.

In order to use different parameters with the equations, we have drawn a *border line* on line 78. It restricts the scope of all specifications above and below it. However, on line 82 we have locally extended the default scope with the `SPECS` specification, allowing all the lines from 73 to 89 (marked with symbols `a` and `b` in the control column) to be included when the Muste editor scans the specifications. Technical details like this demonstrate how the design of a Muste work scheme can resemble, at least to some extent, formal programming.

4 Conclusions and discussion

The Muste project aims to create an R implementation of Survo. Currently, most parts of Survo are implemented in Muste. In this paper, only a few rather simple examples of the editorial approach in Muste were briefly described. They should, however, illustrate the possibilities of a pragmatic “literate programming style” that is a naturally self-documenting result of working with Muste and its “paper and pen” type editorial interface.

Having the essential functionality of Survo, Muste could be used as an “independent” statistical software, although technically implemented as an R package. In this respect, Muste is a software within a software, although it is as integral a part of R as any other R package.

Muste can also be used as a flexible script editor or even more generally, a GUI for R. In addition, Muste provides advanced tools such as Survo data file support and matrix interpreter. With these tools several types of data preprocessing tasks may be easier to perform with Muste than with the traditional R. As Muste data sets are easily transformed to and from R data frames, the data analysis can be performed by alternating between the Muste functions and any R packages providing the methods needed for particular tasks. Compared with the usual interfaces, such as the default interface of R, a great advantage of using Muste is the self-documenting feature of its editorial approach, inherited from the original Survo editor from the beginning of 1980s.

A detailed exposition of the above-mentioned ways of working is excluded from this paper and will be described elsewhere.

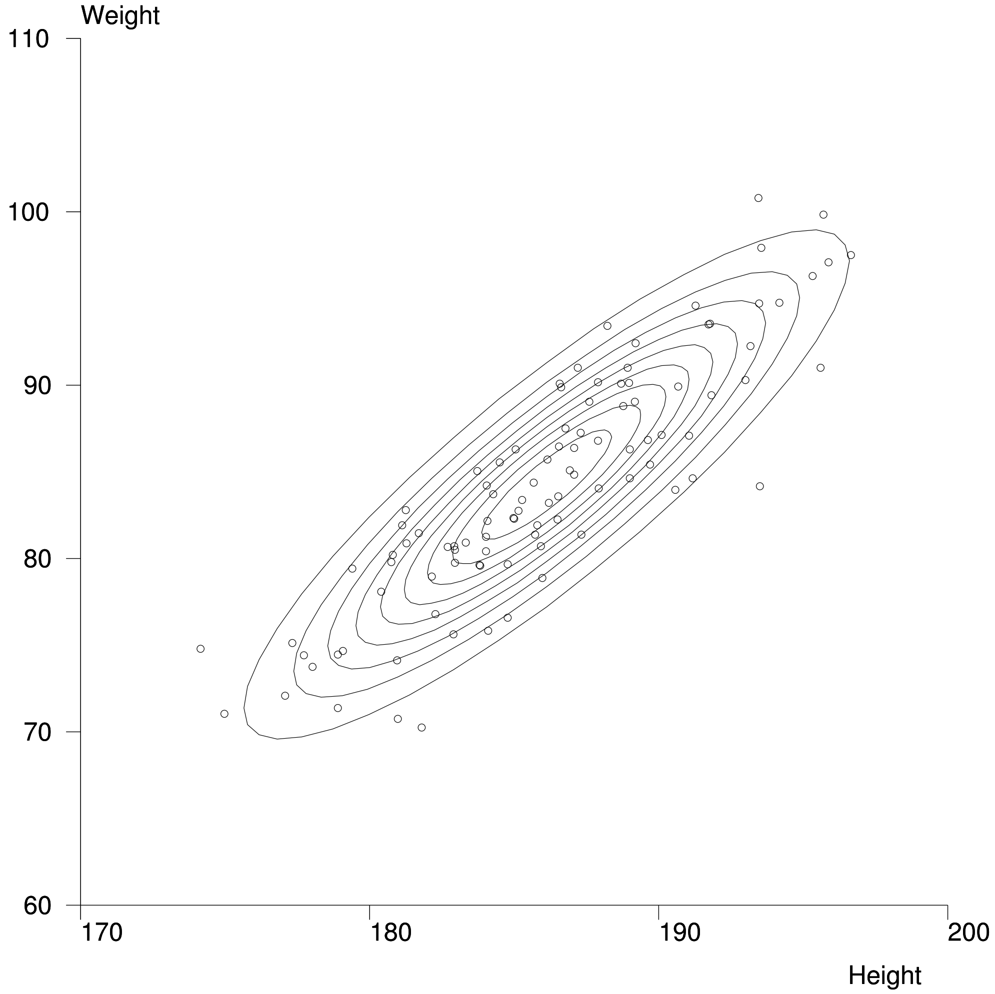


Figure 6: Contour ellipses for simulated data using Mustonen formulas 6

Authors' contributions

Reijo Sund has had the main responsibility for the development of Muste, especially for the R specific parts, and he designed and drafted the manuscript. *Kimmo Vehkalahti* has participated in the development of Survo and Muste as well as in the critical revision of the manuscript. *Seppo Mustonen* is the developer of Survo and his contributions to the Muste project have been extremely significant as most of the functionality in Muste is directly based on his original source codes of Survo. Also the statistical applications presented in the manuscript are based on Mustonen's earlier work. He has participated in the critical revision of the design and contents of the manuscript.

Acknowledgements

Reijo Sund has received a personal research grant from the Finnish Cultural Foundation to realize the Muste project. We would also like to thank two anonymous referees for useful comments.

Bibliography

[1] Alanko, T., Mustonen, S. and Tienari, M. (1968) *A Statistical Programming Language SURVO 66.* BIT, **8**, 69–85.

[2] Chambers, J. (2008) *Software for Data Analysis. Programming with R.* Springer Series in Statistics and Computing. Springer, New York.

[3] Dalgaard, P. (2001) *A primer on the R–Tcl/Tk package.* R News, **1**, 27–31.

[4] Fox, J. (2009) *Aspects of the Social Organization and Trajectory of the R Project.* The R Journal, **1**, 5–13.

[5] Gentleman, R. and Temple Lang, D. (2007) *Statistical Analyses and Reproducible Research.* Journal of Computational and Graphical Statistics, **16**, 1–23.

[6] Ihaka, R. and Gentleman, R. (1996) *R: A Language for Data Analysis and Graphics.* Journal of Computational and Graphical Statistics, **5**, 299–314.

[7] Knuth, D. E. (1984) *Literate Programming.* The Computer Journal, **27**, 97–111.

[8] Mustonen, S. (1963) *SMS, a System of Matrix Subroutines for use with the 803 autocode.* National Elliott computer application program LM21. Elliott Computing Division, Borehamwood.
`http://www.survo.fi/publications/Mustonen1963.pdf`

[9] Mustonen, S. (1977) *SURVO 76, a statistical data processing system.* Research Report No. 6. Department of Statistics, University of Helsinki.
`http://www.survo.fi/publications/ResearchîReportî6îMustonenî1977.pdf`

[10] Mustonen, S. (1980a) *Interactive analysis in SURVO 76.* In: M.M. Barritt and D. Wishart, Editors, *Proceedings of the 4th Symposium on Computational Statistics, COMPSTAT.* pp. 253–259. Physica-Verlag, Wien.

[11] Mustonen, S. (1980b) *SURVO 76 EDITOR, a new tool for interactive statistical computing, text and data management.* Research Report No. 19. Department of Statistics, University of Helsinki.
`http://www.survo.fi/publications/ResearchîReportî19îMustonenî1980.pdf`

[12] Mustonen, S. (1981) *On Interactive Statistical Data Processing.* Scandinavian Journal of Statistics, **8**, 129–136.

[13] Mustonen, S. (1984) *SURVO 84 – Interactive system for statistical computing, graphics and text processing.* Research Report No. 51. Department of Statistics, University of Helsinki.
`http://www.survo.fi/publications/ResearchîReportî51îMustonenî1984.pdf`

[14] Mustonen, S. (1989) *Programming SURVO 84 in C.* SURVO 84C Contributions, **3**. University of Helsinki, Department of Statistics.
`http://www.helsinki.fi/survo/c/index.html`

[15] Mustonen, S. (1992) *Survo – An Integrated Environment for Statistical Computing and Related Areas.* Survo Systems, Helsinki.
`http://www.survo.fi/books/1992/SurvoîBookî1992îwithîcomments.pdf`

[16] Mustonen, S. (1995) *Tilastolliset monimuuttujamenetelmät.* (Statistical multivariate methods), Lecture notes in Finnish. 209 pp. Survo Systems, Helsinki.

[17] Mustonen, S. (2001) *New Windows version of Survo.*
`http://www.survo.fi/mm/english.html`

[18] Mustonen, S. (2005) *Influence curves for the correlation coefficient.*
`http://www.survo.fi/papers/corrcurves.pdf`

[19] Mustonen, S. (2012) *Two formulas related to two-dimensional normal distribution.*
`http://www.survo.fi/papers/twoîformulas.pdf`

[20] Mustonen, S. and Vehkalahti, K. (1997) *Survo as an environment for statistical research and teaching.* Bulletin of International Statistical Institute, Istanbul. Proceedings, **2**, pp. 69–72.
`http://www.helsinki.fi/survo/isi97.html`

[21] R Development Core Team (2012) *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria.
`http://www.R-project.org/`

[22] Rossini, A. J., Heiberger, R. M., Sparapani, R. A., Maechler, M. and Horniki, K. (2004) *Emacs speaks statistics: A multiplatform, multipackage development environment for statistical analysis.* Journal of Computational and Graphical Statistics, **13**, 247–261.

[23] Sund, R. (2011) *Muste – the R implementation of Survo.* Yearbook of the Finnish Statistical Society 2010, pp. 133–146.
`http://www.survo.fi/muste/publications/sund2011îmusteîyearbook.pdf`

[24] van der Hoeven, J. (2006) *GNU TeXmacs: A scientific editing platform.*
`http://http://www.texmacs.org/Samples/texmacs.pdf`

[25] Wolfram Research (2012) *Wolfram Mathematica 8 Documentation Center.*
`http://reference.wolfram.com`