

Statistical Software from a Blind Person's Perspective

R is the best, but we can make it better

by A. Jonathan R. Godfrey

Abstract Blind people have experienced access issues to many software applications since the advent of the Windows operating system; statistical software has proven to follow the rule and not be an exception. The ability to use R within minutes of download with next to no adaptation has opened doors for accessible production of statistical analyses for this author (himself blind) and blind students around the world. This article shows how little is required to make R the most accessible statistical software available today. There is any number of ramifications that this opportunity creates for blind students, especially in terms of their future research and employment prospects. There is potential for making R even better for blind users. The extensibility of R makes this possible through added functionality being made available in an add-on package called **BrailleR**. Functions in this package are intended to make graphical information available in text form.

A short introduction to general accessibility of computers for blind people

Access to computers for blind and vision impaired people is not easy to describe because the way each person does what they do is theirs alone. The level of visual acuity each blind person has may range from absolute zero useful vision through to an ability to read standard print, albeit using magnification of some description. The futility of making assumptions about how a blind person works is best demonstrated using a few examples. First, a person may be able to read 12pt font type on paper but have no access to the same size or larger print on a computer screen. Second, a person may be able to walk around their community in safety without use of a guide dog or white cane, but be entirely unable to read print in any form. For the purposes of this discussion, I use the term “blind” to indicate that a person has no ability to use print in any form, and “low vision” to mean that they can use print in some way; this second group is the harder to define and to cater for as their needs are much more diverse. Another reason for defining these terms is that the word “blind” does have a fairly consistent meaning across the world, but there is quite a lot of inconsistency about the terms used by those people who do not want to be labelled as blind, but do have some reduction in their vision compared to the majority of humans.

The vast majority of blind people use screen reading software. This software takes the text from keyboard entry or what is displayed on the screen and uses synthetic speech to read out as much as possible to the user. Screen reading software cannot handle graphics, except icons or images that are labelled with some substitute text; in web pages this text is known as an “alt tag”.

In many circumstances, screen readers are used by people who have quite reasonable (but might still be described as low) vision as this option is faster than struggling with print based solutions. Use of a screen reader is however still considerably slower than use of normal vision. Users can become quite adept at increasing the speed of the speech to improve their efficiency, but reading speeds seldom surpass 500 words per minute. Limiting the amount of material that is read aloud by the software is therefore a key element in minimizing the amount of irrelevant information that needs to be processed by the user and would otherwise make them less efficient.

Depending on their level of vision, screen reader users can often also use braille displays or image enlargement technology to improve their experiences. Braille

users can have reading speeds that approach the common reading speed of sighted people, but skimming unfamiliar text is difficult at best. The main limitation of braille is again that the access to graphics is basically nonexistent. Tactile image processing is being worked upon ([Gardner et al., 2004](#); [Watanabe et al., 2012](#)), but universal access to the current technology is quite limited. Some examples of successful use of tactile images do exist ([Ebert, 2005](#)), but little success has occurred to date with the graphs found in statistical analyses. Attempts have also been made to convey the information presented in graphs using sounds (rather than speech); success in this endeavour has been limited to plots of mathematical functions rather than the not so well-behaved depictions of data presented in statistical graphs.

For those people who are able to use enlargement software or equipment, access to graphics is possible. In this respect and for the purposes of this presentation, these people are working as sighted people rather than as blind people and are therefore given little further consideration.

Whether it is through use of screen reader software or braille displays, most blind computer users will find new software at least as challenging to engage with as do their sighted peers. The familiarity of menu structures and the like gives as much comfort to the blind user as it does to the sighted user. The issue with new software comes from the use of a multitude of different style dialogue boxes and controls used by the software to achieve the desired outcome. The access to these dialogues is made even more difficult when non-standard controllers are used by software developers. Unfortunately, not all commonly available software development systems are designed to incorporate accessibility into software at the design phase. This means that the screen reader must try to interpret what is given on-screen, rather than being given clear instructions by the software. Icons are a simple example. When a new software application is designed and the creator chooses the standard icon for a common task, it is likely the screen reader already knows what that icon is for. Choosing an alternative that has some minor visual changes on the same theme as the standard icon, poses no challenge for the sighted user; for the blind user, this new icon will not be linked with the intended meaning until the screen reader is told how to interpret the new graphic.

Most blind people do not use the mouse to control their computer. Keyboard use, including key combinations to emulate mouse clicks, means that the blind user moves around menus and dialogue boxes using extra techniques they must learn. Poor design, mainly derived from inconsistent and therefore confusing structures in presentation, creates a problem for blind users coming to grips with new software. Another consideration is the amount of unused functionality built into many applications, which gives the unfamiliar user more work in manipulating dialogue boxes. The blind user who hopes to become less inefficient, quickly learns hot keys which reduce the time taken to perform the most basic tasks. Many well-designed applications show the user what these hot keys are, but many sighted computer users do not know that keyboard alternatives to mouse clicking do exist and that they could also benefit from their use.

Why is R the best?

Aside from the reasons a sighted R user might cite for preferring R over its alternatives, a blind user would add the following features that make R the superior statistical software option.

1. As suggested above, the blind user needs to minimize the amount of text that must be processed to maximise efficiency. The brevity of output that results from what I call a WYRIWYG (what you request is what you get) system is extremely useful for the blind user.

2. The R help pages are all available in HTML. It is not uncommon for help pages to be viewed using a web browser but screen reader software and web browsers already work well together. The help pages in R are simply constructed and consistent. Navigation is easy and the user can build familiarity that does not exist with help systems based on an archive of PDF documents. (More about PDF documents later.) It is also worthwhile mentioning the change in the way R provides a search of the help files since version 2.14.0, with the search results from the `??blah` command leading to a web page as against the former display which was not accessible.
3. R uses text based input and output. Some statistical software uses graphics windows to display simple text output. Screen readers are not sophisticated enough to extract text elements out of a graphics window. Often these objects are created using Java which on occasion can work with screen readers, but shared experiences of blind users are variable (to be kind).
4. R offers the user options for modes of interaction. The command line input using the keyboard (rather than mouse clicks and dialogue boxes) means R works with screen readers and braille displays that echo the typed input. It should be noted here that the dialogue boxes in the graphical user interface (GUI) form of R running under Windows do not all work well with screen readers. It is fortunate therefore that all dialogue box functionality is easily replicated using the command line.
5. R users commonly use script files that can be edited or created using any text editor. This might seem obvious, but while many statistical software applications have this feature, those that are primarily driven by mouse clicks and dialogue boxes are often used by people who cannot support the blind student needing to use commands entered via the keyboard. Documentation to support use of the command syntax for many statistical software options is not always kept up to date.
6. Access to R is there for anyone regardless of operating system, choice of screen reader, braille display, or financial resources. First, note that blindness is more prevalent in less developed countries where the price of R makes it a very attractive option. Second, in an education setting, sighted students can often use university-funded commercial software. A blind student who cannot use the software on university computers could incur extra costs to use statistical software on their own computer which has the extra software and peripherals they need. In such settings, R provides a free back up plan for the blind student.
7. In situations where access is less than complete, such as PDF documents, the blind R user can fall back on the text documents that created the PDF. For example, *package vignettes* are additional documentation for packages provided as PDF documents with sources in Sweave. In this case the un-Sweaved files can be read in any text editor. This offers the blind user an opportunity to learn the necessary commands required to complete tasks.
8. The batch mode for running R jobs provides an alternative means of directing the input and output to files that can be viewed in the user's preferred applications. A common tool for gaining access to software that has difficulty interacting with screen readers is to direct output to a text file which can be viewed in a web browser. As the file is updated via the interaction with the statistical software, the text file is altered. Using the refresh functionality of the browser means the user can continue to interact with the statistical software even though it is not accessible. This is possible using R, but not necessary. It is a useful way of running medium sized tasks that might

have to be tweaked once or twice. Certainly, this author uses batch mode in situations where his colleagues use R interactively.

9. R offers the user access to the information that feeds into a graph in a way that is not paralleled in most statistical software. Whenever a user creates a graph, R processes the data into the form needed to create that summary graph. The object is often stored (implicitly) and discarded once the graph is formed. Some functions print this information out at the user's bidding (a histogram for example where the user has set `plot=FALSE` as an argument). Wrapping a print command around a graph command in R creates the graph and prints out the object used to create it. Occasionally this object is not desirable but there are situations where it provides access to the graph that is not otherwise possible.

Making base R usable by blind useRs

For users of the Windows operating system, the terminal window is the best way to use R interactively. This means the user must alter the shortcut placed on the desktop as part of the standard installation. This is all the blind user must do which makes the task of getting a new software application ready for use with a screen reader. Aside from doing nothing as software is ready to go, this is about as easy as one could ever expect. Many software solutions have special options for screen readers hidden away in the menus somewhere and the worst of these would require a sighted person to alter the settings on the blind user's behalf.

This author has used this approach without issue under Windows XP, but use of R in the terminal window of Windows Vista and Windows 7 has proved problematic for the screen reading software which can "lose focus". This means the cursor is no longer able to be controlled by the blind user and the session must be curtailed. This problem is not specific to a particular screen reader, and can be replicated in other terminal windows such as the command prompt of these operating systems. At the time of writing, this problem remains and the author can only recommend using base R in batch mode to compensate.

The option of using the R console under Windows in conjunction with a screen reader is feasible. The greatest benefit the terminal window offers is that the results from commands are voiced by the screen reader automatically. In the console mode, the screen reader must be switched into a mode that reviews the screen's contents. If the output is longer than can fit on screen, the blind user will not be able to access some of the output.

Those blind people working with Macintosh or Linux systems need to do very little. The Mac user must decide if they will use the terminal window or rely on the console version, but either is fine. Linux users will be working with a terminal-like environment so will be working with R straight out of the box. Both of these operating systems have screen reader software built in. The VoiceOver product for Apple products has clear sound and can be controlled easily. The ORCA screen reader for Linux is not quite as sophisticated and can be more difficult to use. The blind people using these operating systems have done so by choice and know how to work with these screen readers. Less experienced blind people tend to use the Windows operating system, often with a commercial screen reader, although freeware options exist.

In late July 2011, the author exposed a number of blind and low vision students to the opportunities R has to offer them in their upcoming studies as a back-stop should their chosen university's software preference prove inaccessible. These blind students were able to use R with a variety of screen readers (with and without braille displays) under the three major operating systems, including one Linux

user who used a braille display without the addition of the screen reader's speech output. The participants were given accessible documentation that guided them through use of basic R commands for creating simple data objects, creating simple graphs using the **datasets** package, and (optionally) to create simple regression or ANOVA models.

How could R be better?

Given the superiority of R for use by blind people over its competitors, there is a certain reluctance to propose issues. In most cases, however, I think the developers of R and its many users would appreciate gaining an understanding of its limitations for the blind users they may have to work with in the future.

First and most obviously is the inability of a blind person to use the excellent R Commander package **Rcmdr** (Fox, 2005) and similar front ends. The fault is not with the front end itself, but is due to the inability of the toolkit used to create the front end to communicate with the screen reader using each operating system's native API. Unfortunately, the Tcl/Tk GUI toolkit, accessed using the **tcltk** package in R, does not communicate with the screen reading software used by blind people to drive the speech output or braille displays. It is possible to teach the screen reader to engage properly with applications built using the Tcl/Tk GUI toolkit, but every change made in the front end needs to be taught to the screen reader. To the author's knowledge, no implementation of the Tcl/Tk toolkit has created accessible front ends for any software so R is not unusual in this respect. The blind user wishing to use packages like R Commander will therefore only benefit from the functions offered as command line tools.

In direct response to a referee's request, I have investigated the usefulness of RStudio (RStudio, 2013) to a blind user. My findings are that this front end is not useful to a blind user relying on screen reading software. As a consequence of the referee's request, I have created a web page¹ that details my experiences with R, especially when using such tools as R Commander and RStudio. I welcome opportunities to test developments in the wider R Project that may benefit blind users and update this resource accordingly.

¹ See <http://r-resources.massey.ac.nz/StatSoftware> for more details.

There is no question that a requirement for developers of front ends to think of the needs of blind users is unreasonable as we are such a small audience. It does need to be pointed out however that the selection of the toolkit itself is where fault lies, if any is to be apportioned. Personally, I do not feel inclined to find fault, but if a successor to R comes to fruition, then I would want the toolkit included to be one that engages with every operating system in such a way that additional screen reader support is not required. The wxWidgets library (Smart et al., 2011) is an example of a toolkit for developing a GUI that creates applications that do communicate with screen readers without special adjustments being made by the developers or those that have an interest in screen reader software. Such toolkits are unfortunately seldom used and perhaps the most relevant example worth noting is the front end developed for Maxima (Maxima.sourceforge.net, 2012) called wxMaxima².

² The accessible front end wxMaxima has only recently featured as a built-in component of Maxima. See <http://maxima.sourceforge.net/> for details and to download this software.

One of the benefits of using a GUI creation toolkit that communicates with the screen reading software is that accessibility features are generally built into those toolkits and little or no extra work is required from the developer. For example, most GUI applications have hot keys on menu items. Once a user (blind or sighted) learns the hot keys for commonly used menu items they tend to use them. Some of these hot keys are able to be used directly from the command line, while those found most commonly in R are key letters for menu items that can be selected once the menu is on screen. It should be noted that the provision of hot keys does not guarantee access or efficiency. The blind user still needs to remember the key

combinations required to avoid slow browsing or having to remember the location of menu items.

A feature that would be useful for blind users, and perhaps many others, is the desire to save the console (or terminal) window including all input and output for future reference. For the blind user this is an essential means of copying content to a document as they cannot use the mouse to highlight the elements desired. The work around is to use the `sink` command, but this stores output only. Operating in batch mode is the only real option unless additional packages are used that create this functionality, such as that found in the **TeachingDemos** package (Snow, 2010), or the Sweave functionality found in the **utils** package.

The use of HTML for the help pages in R has been mentioned as a major drawback for the blind user. I can report that low vision users would prefer to have greater control over the way these pages are displayed so they can make the best use of their residual vision. This may include the use of colour as well as different font families and styles.

The use of PDF as the format of choice for vignettes and other documentation creates some difficulty for many blind users. The PDF format is not as easy to work with for many reasons, with the total inaccessibility of equations and mathematical notation being a key example. Creation of HTML, or preferably XML in place of PDF for vignettes and documentation would improve the accessibility of this material. If XML is used in place of HTML, the blind reader can use the MathPlayer³ plug in for Internet Explorer which converts the mathematical expression into a plain English text string (Soiffer, 2005). This approach is satisfactory for the vast majority of formulae encountered in an introductory statistics course. Failing this, the blind user must wait until such endeavours as Moore (2009) to improve the accessibility of PDF documents created by \LaTeX / \TeX are successful. Access to equations in PDF documents is a further challenge which is only just receiving attention⁴.

There are several ways to create either HTML or XML documents from \LaTeX , notably using the $\text{\TeX}4\text{\Ht}$ ⁵ package (Gurari, 2004). As an alternative, the Plastex⁶ package is used to create the online documentation (presented in HTML not PDF) for SAS (SAS Institute Inc, 2011).

A second issue with reading PDF files is that many blind users will not be able to deal with the ligatures used in the basic \LaTeX fonts. These are the letter combinations that are printed as a single character, such as ‘fi’, ‘ff’, or ‘ft’. It is possible to disable ligatures in many instances; the competent user may add the necessary \LaTeX commands when they re-process the files distributed with add-on packages. Creation of these files in HTML is therefore also an option available to the blind user, but is probably not an option for the novice R/ \LaTeX user.

A final element of R that is not currently accessible to blind users is the data window. This spreadsheet screen for editing/browsing a data frame directly has proven to be of little value to the blind user under Windows. Fortunately, the blind user can learn how to use R commands to investigate their raw data, or can fall back on other accessible spreadsheet software to do this.

Introducing BrailleR

The **BrailleR** package is under development (Godfrey, 2012) and an initial offering appears on CRAN, as well as having its own homepage⁷. In broad terms the package aims to provide blind users a text interpretation of graph objects, using the (implicitly) stored list object created as part of the function called.

This is being implemented by creation of a method called the VI method as a first step. It is working for some simple graph types such as histograms. Use of a method is reliant on assignment of a distinct class attribute by the function being

³ MathPlayer is available from <http://www.dessci.com/mathplayer/> and is a free download.

⁴ Neil Soiffer of Design Science is working on this problem courtesy of a grant received from the Institute of Education Sciences (U.S. Department of Education) in 2012.

⁵ More details about $\text{\TeX}4\text{\Ht}$ are available at <http://tug.org/applications/tex4ht/mn.html>

⁶ Plastex, found at <http://plastex.sourceforge.net/> (Smith, 2009), is an open source project for converting \LaTeX into HTML and other formats.

⁷ The **BrailleR** project home page is at <http://r-resources.massey.ac.nz/BrailleR>.

used to create the graph. Assignment of class attributes to more objects would broaden the scope of the VI method.

In situations where the class attribute is not yet assigned, creation of a wrapper function that will do the same job is possible. This back-up plan is seen as an undesirable but pragmatic means of achieving the aims of the package. Any functionality worked on in this group of functions will be transferred to the method once the corresponding class attribute assignment is assured.

The functionality offered by the VI method is aimed at describing what the sighted graph user can see as fact, without forming an opinion. This means the user is then reliant on interpreting what the graph tells the sighted person. In the case of a histogram, the VI method will detail the total number of items, the midpoints of bars, and the counts of objects in each bin. The shape of the distribution is left to the user to determine. The method is therefore not meant as any form of expert system. Readers can compare the output from base R functionality with that offered via **BrailleR** for a histogram of a sample of random normal values using:

```
require("BrailleR")
x <- rnorm(1000)
hist(x, plot=FALSE)
VI(hist(x))
```

Displaying summary statistics for data frames is a common task that leads to a nicely formatted display for the sighted person, but a screen reader reads the details line by line. This means hearing all the minimum values, then the lower quartiles, etc. and is often just too difficult to contemplate as a blind user. Simplifying the process of creating summary statistics for data frames is an example of where the blind person's efficiency can be improved by **BrailleR**.

Another efficiency can be brought about by developing functions that fit specific models to a data set, e.g., a multiple regression. The blind user (and sighted ones too) might find it useful to have a function that fits the model and automatically saves the associated output in a text file, as well as placing the residual plots in their own files. It makes sense to automate the process of creating and saving graphs given that the blind user will not actually look at the graphs, and may wish to insert them into a document at a later time. Alternative approaches for considering residual analyses are a key task for implementation in **BrailleR**. This may need to be done using numerical approaches, rules of thumb, and alternative hypothesis testing such as direct consideration of a Levene's test for homogeneity in an analysis of variance example.

Plans for interpreting more difficult constructs such as scatter plots are being considered. The main issue for finding a suitable interpretation is that of simplicity. How does one explain a scatter plot simply and effectively, without the benefit of statistical ideas to someone that cannot see the graph? Further, how would one do this when encountering such a graph in an examination context without answering the question for the blind student? This is not easy. The author has benefitted from overly keen examination supervisors in the past — sometimes he has been limited by their ability too.

In order to meet the needs of a blind user wishing to review an entire R session (warts and all), the terminal or console window needs to be saved in a text file. Functionality for this was found in the **TeachingDemos** package, and the **BrailleR** package is the better for the ability to make use of this functionality directly. Thanks are due to Greg Snow for the code and help files which are now included in **BrailleR**. Users wishing to use a more sophisticated file type than plain text are directed to the use of the **R2HTML** package (Lecoutre, 2003), although the author prefers to use the Sweave functionality of the **utils** package.

Anyone who has experience working with blind students using R, or any other statistical software is invited to contact the author with details of their experiences and resulting questions. Your enquiries will help improve the **BrailleR** package.

Summary

Yes, R could be better, but at the current time it is the best software for blind statisticians. This article has documented the fact that R is accessible to blind users of all operating systems. It has suggested some improvements that will make R even better and therefore set the standard for all statistical software in terms of meeting the needs of blind people around the world.

Running R in the terminal window instead of using the GUI mode of operation means the blind (Windows) user has instant access to as much functionality as their sighted peers. For Windows users, this means a simple alteration to the default shortcut placed on the desktop. The only improvement over this would be if the standard console window was accessible for screen reading software. Given the vast development already undertaken to extend R using the useful but inaccessible Tcl/Tk toolkit, it is likely that the blind community will need to continue operating using the terminal mode.

Of course, it is hoped that any successor to R will incorporate the most recent accessibility standards at the outset and that it will not lose the advantages R currently offers the blind community.

Bibliography

- J. Ebert. Scientists with disabilities: Access all areas. *Nature*, 435(7042):552–554, 2005. [p2]
- J. Fox. The R Commander: A basic statistics graphical user interface to R. *Journal of Statistical Software*, 14(9):1–42, 2005. URL <http://www.jstatsoft.org/v14/i09>. [p5]
- J. Gardner, C. Herden, G. Herden, C. Dreyer, and G. Bulatova. Simultaneous Braille tactile graphics and ink with Tiger ink attachment and Duxbury. In *Proceedings of the 2004 CSUN International Conference on Technology and Persons with Disabilities*, Los Angeles, California, Mar. 2004. [p2]
- A. J. R. Godfrey. *BrailleR: Improved access for blind useRs*, 2012. R package version 0.4. [p6]
- E. M. Gurari. T_EX4ht: HTML production. *TUGboat*, 25(1):39–47, 2004. URL <http://www.tug.org/TUGboat/tb25-1/gurari.pdf>. [p6]
- E. Lecoutre. The R2HTML package. *R News*, 3(3):33–36, Dec. 2003. URL http://cran.r-project.org/doc/Rnews/Rnews_2003-3.pdf. [p7]
- Maxima.sourceforge.net. *Maxima, a Computer Algebra System*, 2012. URL <http://maxima.sourceforge.net/>. [p5]
- R. Moore. Ongoing efforts to generate “tagged PDF” using pdfTEX. *TUGboat*, 30(2):170–175, 2009. URL <http://www.tug.org/TUGboat/tb30-2/tb95moore.pdf>. [p6]
- RStudio. *RStudio: Integrated Development Environment for R, Version 0.97.336*. RStudio, Boston, MA, 2013. URL <http://www.rstudio.com/>. [p5]
- SAS Institute Inc. *The SAS System, Version 9.3*. SAS Institute Inc, Cary, NC, 2011. URL <http://www.sas.com/>. [p6]

- J. Smart, R. Roebeling, V. Zeitlin, R. Dunn, et al. *wxWidgets 2.8.12: A portable C++ and Python GUI toolkit*, Mar. 2011. URL <http://docs.wxwidgets.org/stable/>. Accessed on 2012-02-20. [p5]
- K. D. Smith. *Plastex, A Python Framework for Processing LaTeX Documents*, 2009. URL <http://plastex.sourceforge.net/>. [p6]
- G. Snow. *TeachingDemos: Demonstrations for teaching and learning*, 2010. URL <http://CRAN.R-project.org/package=TeachingDemos>. R package version 2.7. [p6]
- N. Soiffer. MathPlayer: Web-based math accessibility. In *Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 204–205, 2005. [p6]
- T. Watanabe, T. Yamaguchi, and M. Nakagawa. Development of tactile graph automated creation software. In K. Yamaguchi and M. Suzuki, editors, *Proceedings of Digitization and E-Inclusion in Mathematics and Science*, pages 143–146, Tokyo, Japan, 2012. [p2]

A. Jonathan R. Godfrey
Massey University
Palmerston North
New Zealand
a.j.godfrey@massey.ac.nz